Using Counts as Heuristics for the Analysis of Static Models[1]

Keith Thomas Phalp
kphalp@bournemouth.ac.uk,
Empirical Software Engineering Research Group,
Department of Computing,
Bournemouth University,
Poole House, Talbot Campus, Fern Barrow,
Poole, Dorset, BH12 5BB, UK

Steve Counsell
steve@dcs.bbk.ac.uk,
Department of Computer Science,
Birkbeck College, University of London,
Malet Street, London, WC1E 7HX, UK

Liguang Chen
lchen@bournemouth.ac.uk,
Systems Engineering Research Centre,
School of Design, Engineering and Computing,
Bournemouth University,
Poole House, Talbot Campus, Fern Barrow,
Poole, Dorset, BH12 5BB, UK

Martin Shepperd
mshepper@bournemouth.ac.uk,
Empirical Software Engineering Research Group,
Department of Computing,
Bournemouth University,
Poole House, Talbot Campus, Fern Barrow,
Poole, Dorset, BH12 5BB, UK

ESERG: TR98-06

Abstract

The upstream activities of software development are often viewed as both the most important, in terms of cost, and the yet the least understood, and most problematic,

---

[1] An early version of this paper (Phalp and Counsell 1997) was included in the ICSE'97 Workshop on Process Modelling and Empirical Studies of Software Engineering, Boston, 18 May 1997. The authors have since extended the original (position paper), with the addition of a validation framework, and a further empirical study, to form the paper that now appears.

particularly 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, 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. Supporting this analysis is the focus of this paper.

Business processes are typically described with static (diagrammatic) models. This paper proposes the use of measures (counts) to aid analysis and comparison of these static process descriptions. The proposition is illustrated by showing how measures can be applied to a commonly used process-modelling notation, Role Activity Diagrams (RADs). Heuristics for RADs are described and measures suggested which support those heuristics. An example process is used to show how a coupling measure can be used to highlight features in RADs useful to the process modeller.

To fully illustrate the proposition the paper describes and applies a framework for the theoretical validation of the coupling measure. An empirical evaluation follows. This is illustrated by two case studies; the first based on the bidding process of a large telecommunications systems supplier, and the second a study of ten prototyping processes across a number of organisations.

These studies 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, study of the prototyping roles, which exhibited the greatest variation in coupling, found that coupling was highly correlated with the size of the development team. This suggests that prototyping in large projects had a different process to that for small projects, using more mechanisms for communication. Hence, the empirical studies support the view that counts (measures) may be useful in the analysis of static process models.

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

1 Introduction: Why Counts on Process Models

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 will inevitably support those business processes, and hence, it is necessary to understand the business needs, and the context for the proposed system.

Furthermore, the output from this 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). Hence, a number of projects have attempted to model both business process and legacy system, and construct a mapping between them (SEBPC). This mapping is then used in order to predict how changes to the business process affect the system, and hence, aid in its evolution.

Process modelling has been used in software engineering for many years, in order to help to understand, manage and control the development process (Potts 1984). However, the description of customer processes 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. Hence, 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 owners of the process, many of whom are not prepared to invest their time in understanding more complex representations. Hence, analysis of processes often consists solely of inspection of diagrams. Typically, this analysis is guided by the application of heuristics, the experience of the modeller and their knowledge of the particular business domain (Ould 1995). Analysis can be time consuming, and the conclusions drawn often rely too much on the skill of the modeller.

This paper proposes that simple measures of process diagrams can be used to complement analysis of static models. Such measures can be used to highlight features of the model and, hence, the process under scrutiny. To illustrate this idea, the paper uses the notation of Role Activity Diagrams (RADs) (Ould and Roberts 1986), a behavioural approach to process modelling (Curtis, Kellner et al. 1992). The paper suggests how chosen heuristics for RADs, based on those of Ould (Ould 1995), can be supported by associated measures.

The motivation for this paper is to illustrate the utility of using a quantitative approach to aid the analysis of static business process models, not to promote RADs, nor the specific measures of RADs suggested. However, to illustrate the approach fully, the paper includes both a theoretical and an empirical validation of the chosen measures. Section Two describes the notation (RADs), and the suggested measures. Section Three presents and applies a framework for theoretical validation of the proposed measures[2]. Results from two empirical studies are described in Sections Four and Five respectively. These results suggest that the example measures may be useful for the analysis of business processes. Finally, Section Six includes conclusions and suggestions for future work.

2 Generating Counts from Models

Any given representation scheme depicts some perspective of the process which it reflects. For example, it might show the activities and information flow within the business process or alternatively the roles, actions and interactions therein. Consequently, there will be different heuristics for the analysis of such models. Any plausible measures must be based on these heuristics, and be reflected in the particular notation being used. For example, one would have different heuristics and measures for data flow diagrams (Yourdon 1989) to those for RADs.

In addition, the proposed uses of the model will also affect the usefulness of the measure. Henceforth, this paper chooses both an example notation and usage; Role Activity Diagrams being examined in order to help understand existing processes, and to suggest where processes might be restructured. However, the general approach described is applicable to other notations and usage.

To analyse RADs, a selection of the heuristics described by Ould is used. From these heuristics a number of counts have been developed and collected for various process models across a number of application domains (Phalp 1998), (Chen 1997). To illustrate the collection and use of the metrics, a benchmark process, again taken from Ould is described. The paper then introduces industrial examples, known to contain process inefficiencies, which the authors participated in analysing and reengineering. These industrial examples are used as the basis for an empirical evaluation of the metrics (sections Four and Five).

A description of the RAD notation is given Section 2.1. This is followed by a description of the heuristics adopted (Section 2.2), and discussion of the derivation of counts (Section 2.3). Section 2.4 notes the need for validation of the measures, and points towards the theoretical and empirical evaluation that follows.

2.1 An Overview of RADs

Role Activity Diagrams 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).

---

[2] This framework could equally be applied to other static diagrammatic business process modelling notations, and measures based on heuristics for those notations.

Role Activity Diagrams have had extensive use and exposure within process modelling and re-engineering community. Miers (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’.

Figure 1, illustrates a RAD with three roles, Divisional Director, Project Manager and Designer. A role (depicted as a rounded rectangle) groups together activities which may be carried out by a person, group or machine (an actor or an agent). Activities (shaded squares), allow the role to move from its current state to the next. Roles act in parallel, and communicate and synchronise through interactions (shown as unshaded squares joined by a horizontal line). 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. 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) depicted by a point-up triangle. Choice (known as case-refinement) is depicted by a point-down triangle.

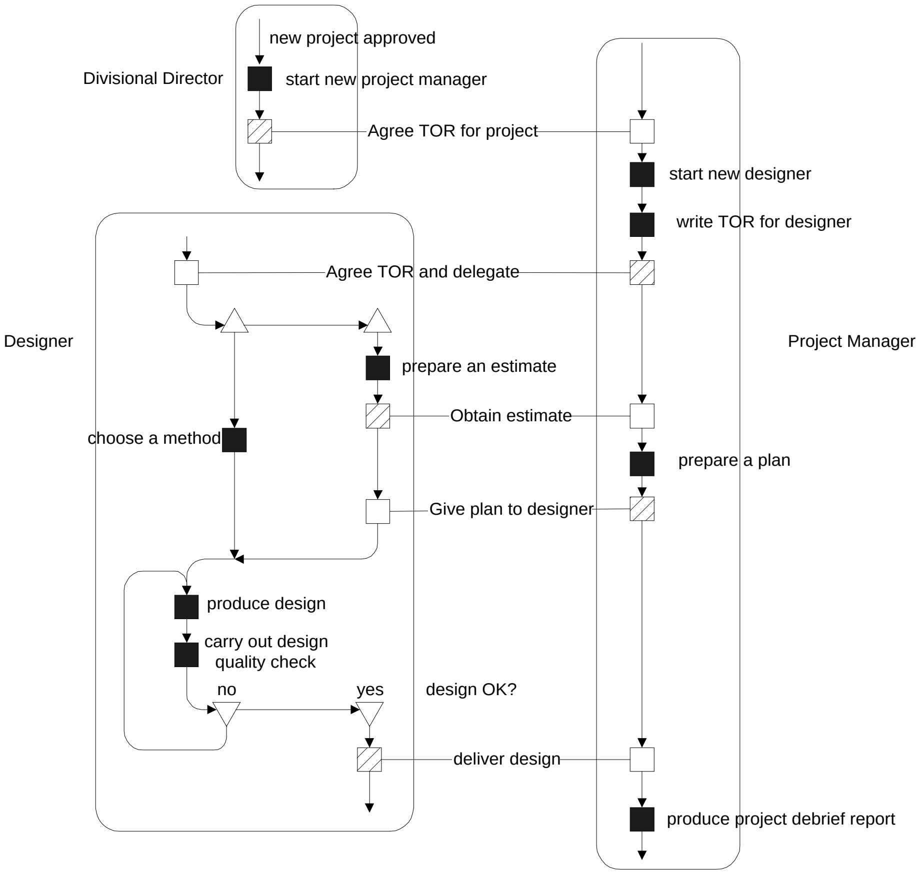

Figure 1: Example Role Activity Diagram

It is noted that roles are like *type*s or *classes* in that they describe a particular kind of behaviour, but are *not* instances of that behaviour. A role may be assigned to a number of different people and 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 roles, and a number of cashier roles. Similarly, a single role may be acted out by a number of different people at different times.

2.2 Heuristics for RADs

It is conceded that some analysis of a process depends on an understanding of the business process. However, equally, when business processes are depicted as RADs, the experienced modeller may look for standard features which suggest where there may need to be subsequent restructuring of the roles and, hence, the process. To explain further, consider a familiar concept for software engineers: *coupling*[3].

Ould believes that within business processes, 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 sensible for business processes. Ould states that:

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

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 states 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, but some separate groupings (roles) are required to maintain cohesiveness. Hence, though one may wish to minimise coupling, some level of coupling (owing to interaction among roles) is necessary. The discussion of what constitutes a 'sensible' level of coupling is further discussed in relation to the empirical work described in sections Four and Five.

2.3 A Simple Measure of Coupling within RADs

The activity *'carry out design quality check'*, performed by the Designer role in Figure 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 the RAD coupling

---

[3] The authors note that there are many alternative definitions of coupling and cohesion within software engineering (Henderson-Sellers, Constantine et al. 1996), (Hitz and Montazeri 1996). However, the paper relies upon Ould's definitions of the terms within RADs.

and cohesion measure. An interaction between role *X* and role *Y* is, therefore, counted as a separate interaction for each role, i.e., represents two interactions, since an interaction square is counted in each role. The 'Role Coupling Factor' (CpF) is calculated by forming the following quotient:

CpF = (interactions in role *X*) / (actions + interactions in *X*).

If a role has only actions, i.e., it engages in no interactions the coupling factor will be zero. In reality, 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 - see Sections Four and Five).

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. In this case, the role is viewed as a separate system with the coupling factor undefined. A simple example now follows.

2.3.1 Example

As an illustration of how coupling factors can be obtained from a RAD, consider the Divisional Director role in Figure 1. It has one interaction, and one action, hence, the coupling factor is 1/2. Consider next the Project Manager role in the same RAD. There are nine activities in total, five of which are interactions, the remainder being actions, hence, the coupling factor 5/9. Consider lastly the Designer role in the same RAD. There are eight activities in total, four interactions and four actions, hence, the coupling factor (after reduction) is 1/2.

In this example, the coupling factors are similar for each role. The RAD is taken from Ould, and is considered to be close to the ideal RAD for the problem under scrutiny. That is, the coupling is at a minimum. An aim in the design of RADs would, therefore, be typically to minimise the degree of coupling between roles. Reducing coupling allows roles to become more autonomous and, hence, because they no longer have to synchronise with other roles, gives them the opportunity to complete their tasks more quickly.

Of course, to find meaningful levels for role coupling it would be necessary to carry out extensive empirical calibration. Even then, it is unlikely that any specific levels for 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, the work described in Section Five (which examines the same type of process across different organisations) appears to suggest that different role-types typically exhibit different levels of coupling. Such issues are further considered in validation of the measure.

2.4 Theoretical Validation and Empirical Evaluation

Increasing importance is being attached to **validating** a metric to ensure that it measures what it purports to measure (Fenton and Pfleeger 1996), (Kitchenham, Pfleeger et al. 1995). In particular, satisfying the *representation condition* of measurement is a first step in this process. The representation condition enforces the rule that a measurement mapping (from entities to numbers and empirical relations into numerical relations) must preserve those relations. For example, if a role, say *A*, has a greater number of interactions than a role *B*, then the coupling factors of those roles will reflect this.

As well as theoretical validation, emphasis should be placed on **empirical validation** to support the claims made by the proposed metrics (Briand, Bunse et al. 1997), (Basili, Briand et al. 1996). For this paper, empirical validation is taken to mean a quantitative or qualitative evaluation of how successfully the metric captures the characteristics it was intended to. An empirical validation of the coupling metric is presented in Sections Four and Five.

In the next section, a framework is provided to allow the theoretical validation of measures such as the coupling factor introduced. The coupling measure will then be validated by reference to this framework, and the implications of this validation discussed.

3 A Framework for Theoretical Validation

In addition, to the Role Coupling Factor (CpF) described in section 2.3, a *system*-coupling factor is defined to be:

SysCpF = (total interactions in system) / (total actions + total interactions in system).

The term *system* refers to a RAD that contains at least one role. For Figure 1, the system coupling is therefore 10/19. The system-coupling factor has an identical range of values to that of the role-coupling factor, i.e., between zero, if there are only actions and no interactions, and one, if there are only interactions and no actions.

To validate coupling factors, the following five properties are adopted. These properties are based on those proposed by (Briand, Daly et al. 1996) for coupling within Object-Oriented systems[4]:

1. **Non-negativity.** The coupling of a role or system cannot be negative.
2. **Range of values.** System and role coupling factors must be in the range zero to one.
3. **Monotonicity**. Adding only communications (interactions) to a role increases the coupling level of that role and of the system.

---

[4] The similarities between role-based process modelling approaches, specifically RADs, and the Object-Oriented paradigm, are described in a further paper (Phalp and Counsell 1998).

4. **Merging of connected roles.** Merging two roles (which previously communicated with each other) to form one composite role decreases the level of coupling in the newly formed role and in the system.
5. **Merging of unconnected roles.** Merging two roles, which are not connected to each other, decreases the level of coupling in the newly formed role, but the system coupling remains constant.

The following Sections (3.1 to 3.5) show how these properties may be applied to both the role and system coupling metrics. This is followed by a discussion of some anomalous properties of the metrics (Section 3.6), which leads to the need for empirical evaluation.

3.1 Satisfaction of Condition One (Non-negativity)

Since a count of either actions or interactions can only be a natural number, then both numerator and denominator must be non-negative, and hence so must the coupling factor.

3.2 Satisfaction of Condition Two (Range of Values)

The level of coupling must range from zero to one. There are three possible cases.

1. When there is a single role in the system both the role and the system have a coupling factor of zero.

2. When there is no communication among any roles, all roles have coupling factors of zero, as does the system.

3. In all other cases, the role and system coupling factors will range between zero and one.

Although, in practice, it is unlikely that any role would not communicate with at least one other role, these cases illustrate that conditions One and Two are rigorous in their application. The role and system coupling factors cannot yield a negative value and the minimum value of each is zero. Hence, conditions One and Two are satisfied.

3.3 Satisfaction of Condition Three (Monotonicity)

Turning to condition Three, the coupling factor for a role increases with the addition of extra communication. From Figure 1, adding only interactions to any of the roles will increase the coupling factors of that role; the system coupling will also increase.

Suppose a communication between Designer and Project Manager roles is added called *check conformance of design with standards.* This would cause the Designer role to have a coupling factor of 5/9, (previously 1/2), and the Project Manager role a coupling factor of 3/5 (previously 5/9). The overall system coupling factor would increase to 12/21 (previously 10/19). Condition Three is therefore satisfied, since the system coupling has increased.

Of course if actions had also been added, the role or system coupling factors may have been reduced. This anomalous aspect of coupling is discussed separately in Section Four.

3.4 Satisfaction of Condition Four (Merging Connected Roles)

Considering condition Four, merging two or more roles so that communication between those roles is no longer necessary decreases the role and system coupling factors. This is intuitive since the number of interactions will reduce, whereas the number of actions will not. To show this, simply reverse the introduction of the communication introduced for condition Three. Note that both the role and system coupling factors would now decrease, thereby satisfying condition Four.

3.5 Satisfaction of Condition Five (Merging Unconnected Roles)

Suppose the RAD of Figure 1 is augmented to incorporate an additional role called *Standards Controller*, responsible for the adherence to certain standards. Assume the standards controller has only a single action: *write-off standards completed.* The coupling factor for the role is zero, and the system coupling factor becomes 10/20 (1/2), where previously it was 10/19.

Assume now that the role of standards controller is to be amalgamated with that of the Designer. The standards controller role disappears, and the action of the Standards Controller becomes an action of the Designer.

The coupling factor for the Designer then becomes 5/10 (1/2), previously 5/9. That is, the role is less coupled; it has more actions, and the same amount of interactions. The coupling factor for the system remains at 1/2, since the same number of actions and interactions are simply redistributed among the roles. Condition Five is therefore satisfied.

3.6 Using Coupling Factors: Some Anomalies

A coupling factor is merely an indication of the coupling of a role or a system in comparison to the amount of internal activity (actions). For example, system coupling does not give a feeling for the overall complexity; rather it gives an indication of how well coupled the roles are in a system. (Note that system coupling is not the same as the average role coupling. This difference is explained fully in Section Five). Adding to the system, say with an additional role, does not necessarily increase the system coupling factor, yet intuitively one feels that the system should have higher coupling.

For example, consider a simple system, where two roles (say *A* and *B*) have a single interaction, and one action each (SysCpF = 2/4 = 1/2).

A third role (*C*) is added, which has a new interaction with *B*. It would seem sensible that the system-coupling factor should increase. Indeed, if only the new interaction is added (as in the Monotonicity condition) then this will be so (SysCpF would become 4/6 = 2/3). However, suppose that role *C* also contains two actions. Then the system-coupling Factor remains the same (4/8). Worse, if role *C* contained more than two actions then the system-coupling factor would be reduced.

Therefore, the system-coupling factor can be counter-intuitive. This indicates that whilst a measure might be valid according to specified criteria, such a measure does not necessarily convey anything meaningful about the artefact under examination. Indeed, reliance on a single measure is not recommended, and a number of authors have cautioned against such an approach (Pfleeger, Fitzgerald et al. 1992). In order to have a meaningful understanding of the process being modelled it is frequently better to examine a number of factors. Hence, any coupling measure would typically be used as one of the factors in the analysis of a RAD.

However, the usefulness of measures can only be gauged properly by their empirical validation. The following Sections (Four and Five) describe two studies, both of which used RADs to model business processes, and used coupling metrics in order to aid the analysis of those models.

4 An Empirical Evaluation: A Bidding Process

In a previous investigation, the bidding process for a large supplier of telecommunications systems was examined and depicted as a series of RADs (Phalp and Counsell 1997). One process fragment analysed comprised several large RADs. Of ten roles depicted, eight had a coupling factor of 1, and the remaining two each had a coupling factor exceeding 4/5. Of thirty roles in the total process, approximately 90% had a coupling factor greater than 4/5. This quantitative analysis suggested an overly coupled process, with a large degree of communication and interaction. Qualitative investigation of the process, by interview and workshops confirmed this view. The investigation revealed a highly bureaucratic process where few actions could be carried out by roles independently, and where the essential actors in the process spent at least 50% of their time in gaining approval for documents. Hence, a redesigned process was recommended which allowed roles far more autonomy in the bidding process, and significantly reduced the coupling and cycle time.

Although such conclusions could have been reached by qualitative analysis by an expert, the use of simple measures allows this analysis to be more efficient, and to quickly highlight areas of concern. Furthermore, it is the authors' experience that the presentation of measures increased the strength of argument for process change.

Though this single study shows that coupling factor might be indicative of some problems, it gives little guidance about what might be sensible levels of coupling. One way to continue would be to examine further processes within the organisation, and try to calibrate the measures internally. However, this approach needs caution, since the nature of processes within an organisation, even at a single site, may be very different.

In contrast, the following study examines the same type of process, in this case prototyping, across a number of organisations.

5 An Empirical Evaluation: Software Prototyping

This study was carried out to gain an insight into the management of prototyping across a number of organisations of various sizes (Chen 1997). Ten processes were investigated and described using Role Activity Diagrams. For each organisation, members of staff acting out the roles were interviewed. A number of visits were made to each site, to validate models and to conduct further interviews. Further process evidence was gathered from both documentary sources and observation. Process analysis consisted of inspection, aided by the collection of simple counts and measures. The application of the coupling factor is now described.

Ref.	Business area	Site size	Main app. domain or project type	Typical configuration of project participants	Main purpose of prototyping	% of prototyping effort by project
1	international banking	large	information system	8 participants: 3 to 5 development team (inc. manager(s) and others) + 2 to 3 customers and end-users.	user requirements	above 60%
2	airway service	medium	information system	30 participants: 3 to 5 development team (inc. manager(s) and others) + 20 to 30 customers and end-users.	user and / or system requirements	above 60%
3	software house	small	information system	15 participants: 1 developer + 10 to 15 customers and end-users	user and system requirements	above 60%
4	hotel service	small	staff scheduling	5 participants: 1 developer + 4 customers and end-users.	user requirements	above 60%
5	university computer centre	small	network monitoring	4 participants: 1 developer + 3 customers and end-users.	user and system requirements	above 60%
6	air traffic control	medium	air traffic control	unknown	risk analysis + system requirements	5-10%
7	electronic engineering	medium	circuit testing	15 participants: 1 prototyper + 10 -15 others (managers, engineers and proving staff) + 1 or 2 customer representatives.	interface design	5-10%
8	telecommunications	large	intelligent networking	80 participants: 12 prototypers + about 65 others (managers, designers, engineers, proving and marketing staff) + 1 or 2 customer representatives.	system design and / or implementation	10-30%
9	telecommunications	large	intelligent networking	30 participants: 9 prototypers + 20 others (managers, proving and commercial staff) + 1 or 2 customer representatives	risk analysis and / or system design	10-15%
10	publishing	large	information system	10 participants: 1 project manager, 1 design manager, 3 developers; 2 customers + 3 users	system design	N/A

Table 1: Characteristics of Processes

Table 1 summarises the characteristics of the ten processes studied. There were three small projects, three medium projects and four large projects. The projects were of varying complexity and spanned application domains. Table 2 shows the coupling measures collected for each process in turn.

Proc.	Org.	Roles	Act	Int	CpF	Mean CpF	| CpF-mean |	| CpF-SysCpF |
1	B	Business Control Board	1	3	0.75	0.82	0.07	0.01
1	B	Project managing	5	13	0.72	0.82	0.10	0.02
1	B	Prototyping	6	11	0.65	0.82	0.17	0.09
1	B	DBA	0	1	1.00	0.82	0.18	0.26
1	B	Customer	0	5	1.00	0.82	0.18	0.26
1	B	End user	1	4	0.80	0.82	0.02	0.06
Process 1 totals (and system coupling)			13	37	**0.74**			
2	A	Project managing	2	8	0.80	0.86	0.06	0.02
2	A	Prototyping (developers)	4	10	0.71	0.86	0.14	0.11
2	A	User group (internal)	1	10	0.91	0.86	0.05	0.08
2	A	End user	0	5	1.00	0.86	0.14	0.18
Process 2 totals (and system coupling)			7	33	**0.83**			
3	D	Project managing	3	6	0.67	0.80	0.14	0.07
3	D	Prototyping	5	6	0.55	0.80	0.26	0.19
3	D	Customer	0	7	1.00	0.80	0.20	0.27
3	D	End user	0	3	1.00	0.80	0.20	0.27
Process 3 totals (and system coupling)			8	22	**0.73**			
4		Project managing	6	7	0.54	0.76	0.22	0.07
4		Prototyping	6	6	0.50	0.76	0.26	0.11
4		Customer	0	5	1.00	0.76	0.24	0.39
4		End user	0	1	1.00	0.76	0.24	0.39
Process 4 totals (and system coupling)			12	19	**0.61**			
5		Managing	6	5	0.45	0.73	0.27	0.12
5		Prototyping	6	5	0.45	0.73	0.27	0.12
5		Customer	0	4	1.00	0.73	0.27	0.43
5		End user	0	2	1.00	0.73	0.27	0.43
Process 5 totals (and system coupling)			12	16	**0.57**			
6		Project managing	2	7	0.78	0.78	0.00	0.00
6		Engineering	4	8	0.67	0.78	0.11	0.11
6		Prototyping	1	2	0.67	0.78	0.11	0.11
6		Customer / user	0	7	1.00	0.78	0.22	0.23
Process 6 totals (and system coupling)			7	24	**0.77**			
7		Business Board	1	2	0.67	0.79	0.13	0.10
7		Project managing	2	6	0.75	0.79	0.04	0.02
7		Prototyping	4	5	0.56	0.79	0.24	0.21
7		Marketing	0	6	1.00	0.79	0.21	0.23
7		Customer	0	4	1.00	0.79	0.21	0.23
Process 7 totals (and system coupling)			7	23	**0.77**			

8	C(dept B)	Project managing	0	3	1.00	0.85	0.15	0.19
8	C(dept B)	System Design	3	9	0.75	0.85	0.10	0.06
8	C(dept B)	Prototyping	1	3	0.75	0.85	0.10	0.06
8	C(dept B)	Component Engineering	2	5	0.71	0.85	0.13	0.10
8	C(dept B)	Proving	2	5	0.71	0.85	0.13	0.10
8	C(dept B)	Marketing / product managing	0	8	1.00	0.85	0.15	0.19
8	C(dept B)	External customer	0	2	1.00	0.85	0.15	0.19
Process 8 totals (and system coupling)			8	35	**0.81**			
9	C(dept A)	Design Managing	4	9	0.69	0.74	0.04	0.03
9	C(dept A)	Prototyping	3	5	0.63	0.74	0.11	0.10
9	C(dept A)	Proving	1	3	0.75	0.74	0.01	0.03
9	C(dept A)	Commercial group	2	7	0.78	0.74	0.04	0.05
9	C(dept A)	Customer	1	5	0.83	0.74	0.10	0.11
Process 9 totals (and system coupling)			11	29	**0.73**			
10	E	Business Control	2	5	0.71	0.89	0.17	0.14
10	E	Project managing	2	10	0.83	0.89	0.05	0.02
10	E	Process Designing	2	10	0.83	0.89	0.05	0.02
10	E	System Developing	2	10	0.83	0.89	0.05	0.02
10	E	Training	0	1	1.00	0.89	0.11	0.14
10	E	Customer	0	8	1.00	0.89	0.11	0.14
10	E	End user	0	4	1.00	0.89	0.11	0.14
Process 10 totals (and system coupling)			8	48	**0.86**			

Table 2: Coupling among Processes

From Table 2, it would be tempting to use the system coupling metric (SysCpF) - shown in bold - as a simple indicator of the coupling within each process. For example, note that Process 9 exhibits the highest level of system coupling. However, to gain an insight into prototyping across organisations, the data needs to be analysed by role type, as will now be described.

Note that SysCpF is not the same as the average of the coupling for each role (mean CpF). The system coupling is calculated from the total of the interaction and action squares in the system. In contrast, the mean of the role coupling simply takes each role-coupling figure and gives the average over the roles. Hence, a role with few actions or interactions has as much impact on the average as one with many. For this reason, system coupling (SysCpF) gives a better (weighted) view of the overall coupling in a system than the mean of the role coupling figures (mean CpF).

5.2 Examination of Coupling by Role Type

Although data was obtained from a variety of organisations, similar roles can be discerned in the processes examined. For example, nine processes have a prototyping role, and eight processes have both a project-managing role and a customer role. It is within these roles that general patterns can be found, specifically with respect to coupling

Table 3 shows coupling ordered by role. (Outlier role coupling figures - the meaning of which will be introduced in this section - are shown in bold)

Proc.	Org.	Roles	Act	Int	CpF	Mean CpF	| CpF-mean |
7		Business Board	1	2	0.67	0.71	0.04
10	E	Business Control	2	5	0.71	0.71	0.00
1	B	Business Control Board	1	3	0.75	0.71	0.04
9	C(dept A)	Commercial group	2	7	0.78	0.78	0.00
8	C(dept B)	Component Engineering	2	5	0.71	0.71	0.00
1	B	Customer	0	5	1.00	0.98	0.02
3	D	Customer	0	7	1.00	0.98	0.02
4		Customer	0	5	1.00	0.98	0.02
5		Customer	0	4	1.00	0.98	0.02
7		Customer	0	4	1.00	0.98	0.02
9	C(dept A)	Customer	1	5	**0.83**	0.98	**0.15**
10	E	Customer	0	8	1.00	0.98	0.02
6		Customer / user	0	7	1.00	0.98	0.02
1	B	DBA	0	1	1.00	1.00	0.00
9	C(dept A)	Design Managing	4	9	0.69	0.69	0.00
1	B	End user	1	4	**0.80**	0.97	**0.17**
2	A	End user	0	5	1.00	0.97	0.03
3	D	End user	0	3	1.00	0.97	0.03
4		End user	0	1	1.00	0.97	0.03
5		End user	0	2	1.00	0.97	0.03
10	E	End user	0	4	1.00	0.97	0.03
8	C(dept B)	External customer	0	2	1.00	0.97	0.03
6		Engineering	4	8	0.67	0.67	0.00
5		Managing	6	5	0.45	0.45	0.00
7		Marketing	0	6	1.00	1.00	0.00
8	C(dept B)	Marketing / product managing	0	8	1.00	1.00	0.00
10	E	Process Designing	2	10	0.83	0.83	0.00
1	B	Project managing	5	13	0.72	0.76	0.04
2	A	Project managing	2	8	0.80	0.76	0.04
3	D	Project managing	3	6	0.67	0.76	0.09
4		Project managing	6	7	**0.54**	0.76	**0.22**
6		Project managing	2	7	0.78	0.76	0.02
7		Project managing	2	6	0.75	0.76	0.01
8	C(dept B)	Project managing	0	3	**1.00**	0.76	**0.24**
10	E	Project managing	2	10	0.83	0.76	0.07

1	B	Prototyping	6	11	0.65	0.61	0.04
3	D	Prototyping	5	6	0.55	0.61	0.06
4		Prototyping	6	6	**0.50**	0.61	**0.11**
5		Prototyping	6	5	**0.45**	0.61	**0.15**
6		Prototyping	1	2	0.67	0.61	0.06
7		Prototyping	4	5	0.56	0.61	0.05
8	C(dept B)	Prototyping	1	3	**0.75**	0.61	**0.14**
9	C(dept A)	Prototyping	3	5	0.63	0.61	0.02
2	A	Prototyping (developers)	4	10	**0.71**	0.61	**0.11**
8	C(dept B)	Proving	2	5	0.71	0.73	0.02
9	C(dept A)	Proving	1	3	0.75	0.73	0.02
8	C(dept B)	System Design	3	9	0.75	0.75	0.00
10	E	System Developing	2	10	0.83	0.83	0.00
10	E	Training	0	1	1.00	1.00	0.00
2	A	User group (internal)	1	10	0.91	0.91	0.00

Table 3: Coupling analysed by role type

Consider first, the customer role. In seven of the eight processes, the customer has a coupling factor of 1, with a mean of 0.98. This contradicts the initial view stated that coupling should be minimised. However, one might always expect customers to be very highly coupled because from the perspective of the systems engineer the customer is a passive role. That is, the internal activities (actions) of the customer are not usually modelled. The exceptional case (process 9) occurs because the developer and customer had a long-standing relationship in which the customer was active in specifying system requirements.

Consider next, the end-user role. A similar pattern occurs to that in the customer role; all but one of the processes having a coupling factor of 1. Again, one might expect this, given the development perspective.

Some roles, however, appear to exhibit greater variation. 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. This does not necessarily point to a problem in process Four. It could be that process Four exhibits low coupling because it has been better structured. Examining the grouping by process reveals that process Four has only four roles and exhibits the second lowest overall coupling. Project managing and prototyping roles having CpF of 0.54 and 0.50 respectively, with customer and end-user roles having a coupling factor of 1. Both processes Four and Five (which exhibit the lowest overall coupling figures) were small, well-organised teams, working on small, well-defined projects. Hence, communication was minimised.

These findings suggest an oversimplification in analysis and hence it is 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. Chart 1 shows the factors plotted together (showing a similar pattern), and chart 2 shows a scatter plot of Prototyping Factor (X) against System Factor (Y).

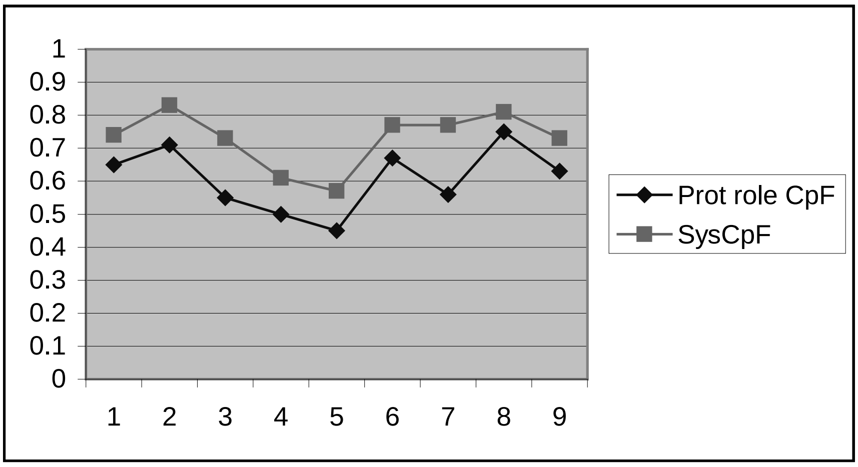

Chart1: System Coupling and Prototyping Role Coupling Factors

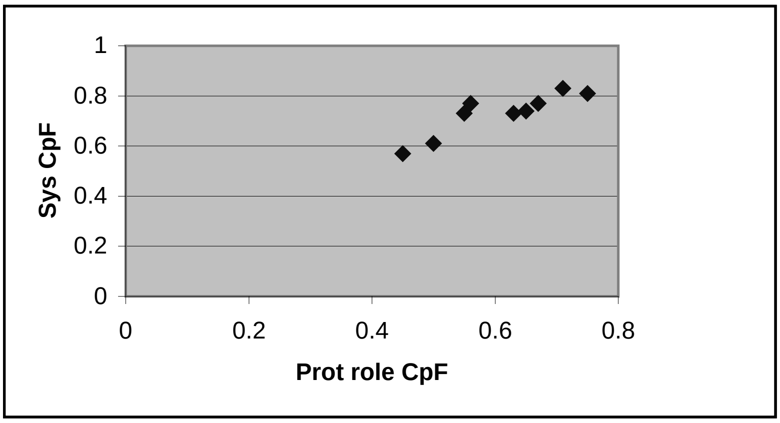

Chart1: System Coupling plotted against Prototyping Role Coupling

Table 4 shows the system coupling and prototyping role coupling factors for the each of the nine processes that include a prototyping role. The rankings for each of these factors and the correlation value are also shown. Note that system coupling and prototyping role coupling are significantly correlated (at the 1-% level). This further suggests that the prototyping roles account for much of the variation in system coupling among the processes examined.

Process	Prot role CpF	SysCpF	Sys CpF Rank	Prot CpF Rank	Corr.
1	0.65	0.74	5	4	0.89
2	0.71	0.83	1	2	
3	0.55	0.73	6	7	
4	0.5	0.61	8	8	
5	0.45	0.57	9	9	
6	0.67	0.77	3	3	
7	0.56	0.77	3	6	
8	0.75	0.81	2	1	
9	0.63	0.73	6	5	

Table 4: Relationship between System Coupling and Coupling in Prototyping Role

Firstly, 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). Secondly, this points to the need for further investigation of the prototyping role.

For one of the processes that had a prototyping role, no data on development team size and participants was available. Table 5 shows the prototyping role-coupling factor, development team size, average project participant size and associated correlation, for each of the eight remaining processes.

Proc	Prot CpF	Team size	Part. size	Correlations	
1	0.65	4	8	Part. & ProtCpF	0.782
2	0.71	4	30	Team & ProtCpF	0.645
3	0.55	1	15		
4	0.50	1	5		
5	0.45	1	4		
7	0.56	12	15		
8	0.75	65	80		
9	0.63	30	30		

Table 5: Relationship between Project Characteristics and Coupling in Prototyping Role

The results show that the prototyping role-coupling factor is correlated (at the 5% level) with the number of participants and the development team size. Chart 3 shows a scatter plot of participant size (Y) against prototyping coupling factor (X); in this case, the stronger of the correlation results. However, the scatter plot also reveals that the relationship may not be linear.

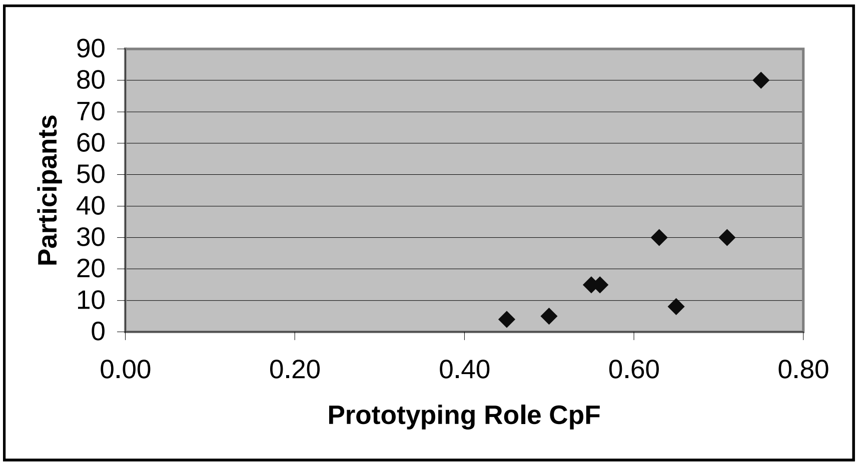

Chart 3: Scatter plot of participant size (Y) against prototyping coupling factor (X)

Proc.	Part. size	Team size	Part. Rk	Team Rk	Prot. CpF Rk	Correlations	
1	8	4	6	4	3	Part. & ProtCpF	0.80
2	30	4	2	4	2	Team & ProtCpF	0.79
3	15	1	4	6	6		
4	5	1	7	6	7		
5	4	1	8	6	8		
7	15	12	4	3	5		
8	80	65	1	1	1		
9	30	30	2	2	4		

Table 5: Rankings of Project Characteristics and Prototyping CpF and associated correlation

Table 6 shows the associated rankings for coupling factors and team and participant size. Note the much stronger correlation of the rankings, showing a strong monotonic (if non-linear) relationship between size (both of team and participants), and the coupling factor.

These findings suggest that for larger projects, particularly where there are more participants involved in the process, more communication mechanisms are employed. It is not just that more communication takes place, but that more types of communication take place. For example, suppose a management role has a communication with a designer role, called ‘allocate work’. Irrespective of whether there were a single manager and a single designer or multiple designers and managers having this type of communication the depiction of the process in RADs would be the same, since roles show type of behaviour, not instances of it. Hence, behaviour in larger projects (and processes) is different, using more communication mechanisms.

5.3 Summary of Findings for Prototyping

This 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 initial study is limited in its coverage; however, it is likely that guidelines need to developed for role types, and that these guidelines could be used to set threshold values and to aid identification of outliers. The authors note there is likely to be significant variation across different organisations or sites, and that general guidelines across organisations are less feasible. However, the need to consider and distinguish different role types within an organisation appears to be an implication of this study.

6 Conclusions

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 the business model. This has been illustrated by describing a coupling measure for Role Activity Diagrams. The utility of this coupling measure has been demonstrated by application to two real world case studies, the latter being part of a major study of prototyping roles across ten software development processes.

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.

However, there were specific findings from the empirical studies, notably from the study of prototyping processes. Study of these prototyping processes then found that the same role types exhibited similar coupling levels across organisations. Furthermore, instances of roles that did not conform to tentative threshold values were explained by unusual circumstances. Finally, the coupling metric enabled the identification of a role type (prototyping) which appeared to include different kinds of behaviour. Importantly this role accounted for much of the variation in the system coupling for each process. Further study found that prototyping role coupling was correlated both with the team size, and with the average number of participants in the process. Hence, that the prototyping role was hiding different behaviour, and that more communication mechanisms were employed when team size, or participants size was increased.

These empirical studies suggest that the coupling metrics 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.

The major finding of this paper is that there is potential for the application of simple counts to static process models. An example measure has been described, and then applied to industrial case studies. Findings from these empirical studies seem to support the view that such a simple quantitative approach aids the analysis of business processes.

The authors recognise the need for more controlled empirical trials to assess the usefulness of metrics in restructuring business processes. These empirical trials would take the form of case studies to test hypotheses relating features of the process model to measures of the process (DESMET 1994), (Pfleeger 1994). However, it is felt that the preliminary work described in this paper suggests that there is merit in such further research. Hence, the paper demonstrates supports the general proposition, that there is merit in applying simple counts to complement analysis of static business process models.

References

Basili, V. R., L. C. Briand, et al. (1996). “A Validation of Object-Oriented Design Metrics as Quality Indicators.” IEEE Transactions on Software Engineering **22**(10): 751-761.

Briand, L., L. Bunse, et al. (1997). An Experimental Comparison of the Maintainability of Object-Oriented and Structured Design Documents. Proceedings of Empirical Assessment in Software Engineering: EASE'97, Keele, UK.

Briand, L. C., J. W. Daly, et al. (1996). A Unified Framework for Coupling Measurement in Object-Oriented Systems, Fraunhofer Institute for Experimental Software Engineering.

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.

DESMET (1994). Case Study Design and Analysis Procedures (CSDA), National Computing Centre Ltd (NCC).

Fenton, N. E. and S. L. Pfleeger (1996). Software Metrics, A Rigorous and Practical Approach.

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

Henderson-Sellers, B., L. L. Constantine, et al. (1996). “Coupling and Cohesion (towards a valid metrics suite for object-oriented analysis and design).” Object Oriented Systems **3**(3): 143--158.

Hitz, M. and B. Montazeri (1996). Measuring Coupling and Cohesion In Object-Oriented Systems. Proceedings of the International Symposium on Applied Computing.

Kitchenham, B., S. L. Pfleeger, et al. (1995). “Towards a Framework for Software Measurement Validation.” IEEE Transactions on Software Engineering **21**(12): 929-944.

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.

Ould, M. A. and C. Roberts (1986). Modelling Iteration in the Software Process. Proceedings of the 3rd International Software Process Workshop, Breckenridge, Colorado, USA, IEEE Computer Society Press.

Pfleeger, S. L. (1994). “Design and Analysis in Software Engineering.” ACM SIGSOFT Software Engineering Notes **19**(4): 16-20.

Pfleeger, S. L., J. C. J. Fitzgerald, et al. (1992). “Using multiple metrics for analysis of improvement.” Software Quality Journal **1**: 27-36.

Phalp, K. T. (1998). “The CAP Framework for Business Process Modelling.” Information and Software Technology **To appear**.

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

Phalp, K. T. and S. J. Counsell (1998). On the Relationship between Business Models and object oriented systems. Available at: http://xanadu.bournemouth.ac.uk/staff/kphalp/papers/oobpr/oobpr.html.

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

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 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/. Homepage at: http://www.staff.ecs.soton.ac.uk/~ph/sebpc/.

Yourdon, E. (1989). Modern Structured Analysis, Prentice Hall.

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