Software Quality Journal, 9, 223–240, 2001

Coupling Trends in Industrial Prototyping Roles: An Empirical Investigation

KEITH PHALP kphalp@bournemouth.ac.uk
Empirical Software Engineering Group, Bournemouth University, Bournemouth, UK

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

Abstract. Prototyping in the development cycle claims to improve analyst understanding of system requirements leading to systems which match those requirements more closely. The quality of the end systems, from a user perspective, is thus improved. In this paper, the results of an empirical investigation into the use of prototyping in the development of various systems is described. Nine processes across eight different sites were analysed and data relating to each process was collected. The notation of Role Activity Diagrams (RADs) was used to capture each of the nine processes. Five hypotheses were then investigated: these related the prototyping role to features of other roles such as extent of interaction during the prototyping process, the effect site size had on the prototyping role and the dependence of the prototyping role on other roles in the prototyping process. Two coupling metrics were introduced to quantitatively analyse these RAD features. Results indicated a strong similarity between large and medium-sized sites in terms of interactions and behaviour. The prototyping process in small sites, however, was found to be different to large and medium-sized sites in both respects. The study demonstrates how measures of business models can aid analysis of the process as well as the products of systems development and highlights the need for more empirical investigation into this and other facets of the software development process.

Keywords: prototyping, coupling, requirements, empirical

1. Introduction

Various claims have been made of the prototyping process as part of the software development process. Through participation and communication with the user, prototyping claims to reduce the risk of building the wrong product. Through greater understanding of requirements by the prototyper, it also claims to enhance the quality of the end software product. The delivered system stands a greater chance of meeting user expectations of the system, easing the maintenance burden as a result.

As well as throw-away prototypes, significant benefits have also been suggested for developing prototypes that evolve over time (Davis, 1992). A number of previous studies and research projects have also investigated the claims made of prototyping. For example, (Gordon and Bieman, 1995) claims that in thirty-three of thirty-nine industrial applications of prototyping, improvement in factors such as developer productivity and the end product itself were found. The research extends earlier work which focused on software quality issues only (Gordon and Bieman, 1993).

In (Coleman and Verbruggen, 1998), a prototyping method is proposed which the authors claim can be used to help small companies achieve Capability Maturity Model (CMM) level two. In (Lichter et al., 1994), a study of five industrial software projects using prototyping was made. The conclusion reached was that an understanding of the aspects of the development process which prototyping was intended to cover and proper user involvement were of paramount importance to the success of a project using prototyping. The importance of user involvement in the prototyping process is also stressed in (Brooks, 1987).

A point and counter-point argument on prototyping was described in (Reilly, 1995) and (Carmel, 1995). The argument in favour of prototyping was that it improved the requirements gathering process and could bring reductions in the time it took to deliver software systems. The counter-argument was that prototyping needed proper tools and qualified staff to carry out the prototyping; without these prototyping is unlikely to be a worthwhile exercise. Some doubts have been cast on prototyping as a method; (Card, 1995) suggests that Rapid Application Development (RAD) may be just a passing fad, and that at best, only in certain cases will adoption of prototyping result in any tangible benefits.

In this paper, an underlying assumption is that the interactions (i.e., communication coupling) between a prototyper and other parties involved in the development process is likely to be a key factor in the success or otherwise of the prototyping process. Little research has been undertaken whether certain types of agent in the development process exhibit similar levels of coupling, show consistently differing levels of coupling or whether characteristics of the development process dictate the amount of coupling between agents. A case study of the prototyping process in the software development process of eight sites of varying sizes and application domains is described. At each of the sites studied, the motivation for carrying out prototyping was to enable clarification of end-user or customer requirements. As such, the prototyping process at each site was exploratory in nature. The modelling notation of Role Activity Diagrams (RADs) (Ould, 1995) was used to model the processes investigated.

Data was collected from nine RAD processes containing a prototyping role, and five hypotheses related to coupling then tested. These hypotheses related to whether prototyping roles had greater numbers of actions and interactions than other role-types; whether the size of site affected the number of interactions of prototyping roles and whether the prototyping and project management roles were more coupled to each other than any other two roles; two further hypotheses related the coupling measures (coupling factors) used in this paper to features of the prototyping role. While large and medium-sized sites tended to be similar in nature, a distinct difference was noted between these two types of site and those of small sites. The prototypes developed by the prototyping process and described in this paper were, in most cases, intended to be throw-away. However, as is often the case, the prototypes produced were deemed useful enough to be used at later stages of development, that is, they became evolvable prototypes. The results in this paper build on earlier work (Phalp and Counsell, 2000), in which analysis of the interactions in the same processes revealed that the project manager interacted with the prototyper far more often in large developments than in small or medium-sized developments.

However, greater numbers of interactions between the project manager and end-user were found in small-sized developments.

In Section 2, we describe the notation of Role Activity Diagrams. In Section 3, we describe two coupling metrics which we collected from each of the nine RADs investigated. We then describe the applications domains investigated and the five hypotheses tested (Section 4). In Section 5, we discuss the results from our analysis. We then discuss the implications of the results (Section 6) and, finally, draw some conclusions and point to further work (Section 7).

2. Role Activity Diagrams

Role Activity Diagrams were initially developed for software process modelling (Ould and Roberts, 1986). RAD notation reflects a move away from functional depiction of organizations towards understanding of behaviour and interactions of individuals or groups (Handy, 1976). Role Activity Diagrams have been used extensively within process modelling and reengineering, and are considered a "state of the art" technique for modelling the business process (Miers, 1994).

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.

Roles are like *types* or *classes* (as found in object-oriented (OO) languages) in that they describe a particular kind of behaviour. They are not, however, 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 shop assistant roles, and a number of manager roles. A single role may be acted out by a number of different people at different times. It is therefore useful to think of *instances* of the role. In the RADs used as a basis of our study, the prototyping roles usually represent a single person within the organization, and hence there is only usually one instance of that role.

The RAD in Figure 1 is taken from Ould (Ould, 1995), and is generally considered close to the ideal RAD for that process (coupling is at a minimum). An aim in the design of RADs would be, typically, to minimise the degree of coupling, hence allowing roles to become more autonomous. As such, they no longer have to synchronise with other roles, giving them the opportunity to complete their tasks more quickly.

Earlier work has shown that RADs can be a useful means of capturing organizational processes (Phalp and Counsell, 1997; Phalp and Shepperd, 1999); simple measures can be used to complement and guide expert analysis of process models (Phalp and Shepperd, 1999). Before defining our coupling metrics, we make

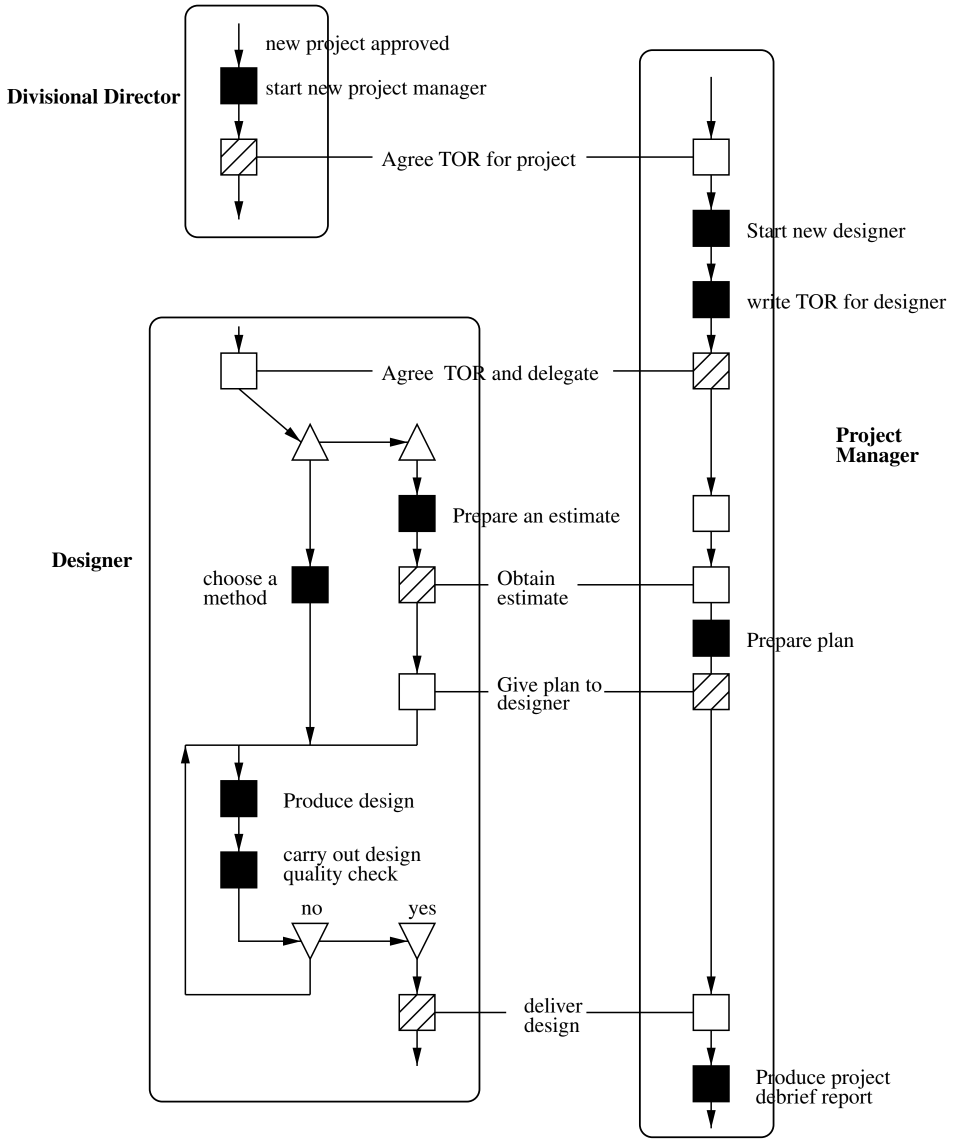

Figure 1. Example Role Activity Diagram.

the important distinction between a *product* and a *process* metric. Both measure different aspects of the development process. A product metric is a measure of a software artifact, for example, lines of code; it relates to the product itself. A process metric is a measure of the process of software development rather than the product itself; for example, the number of hours taken by a developer to carry out a maintenance task. The distinction between these two types of metric is important in this paper, where we are analysing the process of prototyping through the use of product metrics. The activities within the RADs reflect the process.

3. Quantitative measures of RADs

In the software engineering community, empirical evaluation can be used to investigate the association between proposed software metrics and other indicators of software quality such as maintainability or understandability. Thorough qualitative or quantitative analyses can be used to support these investigations (Basili et al., 1996; Basil, 1996). Various metrics have been proposed for the analysis of software; more recently the focus has been on OO software (Harrison et al., 1998; Lorenz and Kidd, 1994). In this paper, we use two coupling metrics applicable to RADs, both of which were proposed in (Phalp and Counsell, 1997). Both metrics are classified as ratio metrics, the chief advantage of which is that, in the context of this study, such metrics permit RADs of different sizes to be compared. For an in-depth discussion of theoretical metric issues including the different measurement scales, the reader should consult (Fenton and Pfleeger, 1996).

For each prototyping role, a Coupling Factor (CF) was extracted from its respective RAD. The CF is calculated as:

$$I/(A+I)$$

where I is the number of interactions a prototyping role engages in and A the number of actions the prototyping role engages in. We note that an interaction between role X and role Y is counted twice (one for each role). If a role engages in only interactions then its CF value is one; if a role engages in only actions then its CF value is zero. The latter case is unlikely, since the role would play no part in the business process. Based on its definition as a ratio scale metric, it is possible for the CF metric to take the same value for differing numbers of actions. We note, however, that this is a feature of any metric defined as such. We also use the System Coupling Factor (SCF) of (Phalp and shepperd, 1999) calculated as:

$$I_{sys}/(A_{sys}+I_{sys})$$

where I_{sys} is the number of interactions in the system and A_{sys} is the number of actions in the system. Consider the Project Manager role in Figure 1. It has 5 interactions and 4 actions. Its CF is therefore 5/9. The total number of interactions is 10 and total number of actions 9. The SCF is therefore 10/19. In the next section we describe details of the evaluation carried out. The analysis described in this paper reflects a static (as opposed to a dynamic) analysis of RADs. The metrics are therefore a view of role action and interaction *types* in the prototyping process.

4. Evaluation details

4.1. Data capture

To produce the RADs on which our evaluation rests, a number of visits were made to each site to validate models and to conduct further interviews over the course of two to three years. Data collected from interviews was documented on formatted forms for later analysis. Further process evidence was gathered from both documentary sources and observation (Chen, 1997). The data collected formed the basis of the RAD diagrams reflecting the prototyping roles in each of the organizations. The hypotheses then investigated are described in the following section.

4.2. Hypotheses investigated

The following five hypotheses were investigated:

1. **H1**: Prototyping roles have greater numbers of actions and interactions than other role-types.

   This hypothesis is based on the belief that to carry out its function properly, prototypers must not only carry out a wide range of actions, but must also communicate extensively with various other roles.
2. **H2**: The larger the size of the organization, the greater the number of interactions undertaken by prototyping roles.

   This hypothesis is based on the belief that larger organizations cause additional staff to become involved in the prototyping process and, hence, cause more interaction. Larger organizations will tend to have more rigid lines of communication as part of the overall prototyping process.
3. **H3**: Coupling Factors for prototyping roles are smaller than Coupling Factors for other types of role.

   This hypothesis is based on the belief that whilst prototyping processes interact extensively, they also carry out a higher proportion of internal actions than other roles (the CF value reflects the ratio of interactions to all activities of a role).
4. **H4**: A higher number of interactions take place between prototyping roles and project management roles than interactions between prototyping roles and any other roles.

   This hypothesis is based on the belief that the prototyping role is more dependent on management roles than on any other role to achieve its function. In all sizes of developments, there is likely to be certain controls imposed by management on the role of the prototyper.
5. **H5**: System Coupling Factors for large organizations are greater than System Coupling Factors for medium and small-sized organizations.

   This hypothesis is based on the belief that in large organizations, a RAD will contain a higher proportion of interactions than at medium or small-sized sites. The nature of communication in large and medium-sized developments is likely to be more structured and rigid than in small-sized developments. In other words, the more people involved, the more communication patterns introduced.

4.3. Application domains

Table 1 contains descriptions of the nine RADs analysed. Altogether, three small projects, three medium projects and three large projects were analysed as part of

Table 1. Descriptions of the nine RADs investigated

RAD	Site size	Application domain
1. International banking	Large	Information system
2. Telecommunications	Large	Intelligent networking
3. Telecommunications	Large	Intelligent networking
4. Airway service	Medium	Information system
5. Air traffic control	Medium	Air traffic control
6. Electronic eng.	Medium	Circuit testing
7. Software house	Small	Information system
8. Hotel service	Small	Staff scheduling
9. Univ. computer centre	Small	Network monitoring

the investigation. Each project was labelled by size according to the number of staff on the project, the number of staff involved in the prototyping process and the nature of the roles themselves (see Table 1). Two of the RADs (from projects 2 and 3) were drawn from the same telecommunications organization. The application domains ranged from a typical information system application to a circuit testing system.

Table 2 summarises the roles found in each of the nine RADs. Every RAD contained a management role and a prototyping role. RADs 3 and 9 differed slightly from other roles in that they had Design Managing and Managing roles respectively (rather than a Project Manager role). However, for the clarity of analysis, in this paper, all managing roles were considered equivalent in the functions that they carried out. Every RAD also contained a customer or end-user role (or in some cases both). The customer and end-user roles were also similar in function (since they both represented the target domain for the application and hence were important stakeholders in the system). RADs 2, 3, 5 and 6 all contained a customer role rather

Table 2. Roles in each of the nine RADs investigated

Role	1	2	3	4	5	6	7	8	9
Project managing	•	•		•	•	•	•	•	
Prototyping	•	•	•	•	•	•	•	•	•
Customer	•		•			•	•	•	•
End-user	•			•			•	•	•
Marketing		•				•			
Business board						•			
Managing									•
DBA	•								
Customer/user					•				
Commercial group			•						
Component engineering		•							
Design managing			•						
Engineering					•				
Proving		•	•						
System design		•							
User group				•					
External customer		•							

than an end-user role. In RAD 2, the role was called an external customer and in RAD 5, the role was amalgamated. Again, for the purposes of our analysis, the role of customer and end-user are considered the same. To justify this assumption, we note that for every RAD which contained both a customer and end-user, the same interactions were made between the prototyper and those two roles. In other words, the customer and end-user were both present during the prototyping process. We also note that the Proving role (RADs 2 and 3) is better known as a Quality Assurance (QA) role, responsible for the quality of the process and products throughout the development process.

5. Empirical evaluation

In the following five sections, each of the hypotheses is evaluated using appropriate collected data. We note that in analysing the data collected, the median value is preferred over the mean since a median is not influenced as much by the presence of large values in the data set (as would be the case with the mean). We also note that, because of the small sample size (9), only limited statistical analyses could be undertaken.

5.1. Hypothesis ***H1***

Hypothesis **H1** assesses whether prototyping roles have greater numbers of actions and interactions than other role-types. To test Hypothesis **H1**, the actions and interactions in each of the nine prototyping roles were collected. The data relating to *actions* in prototyping and non-prototyping roles is illustrated graphically in Figure 2. Data relating to *interactions* in prototyping and non-prototyping roles is illustrated graphically in Figure 3.

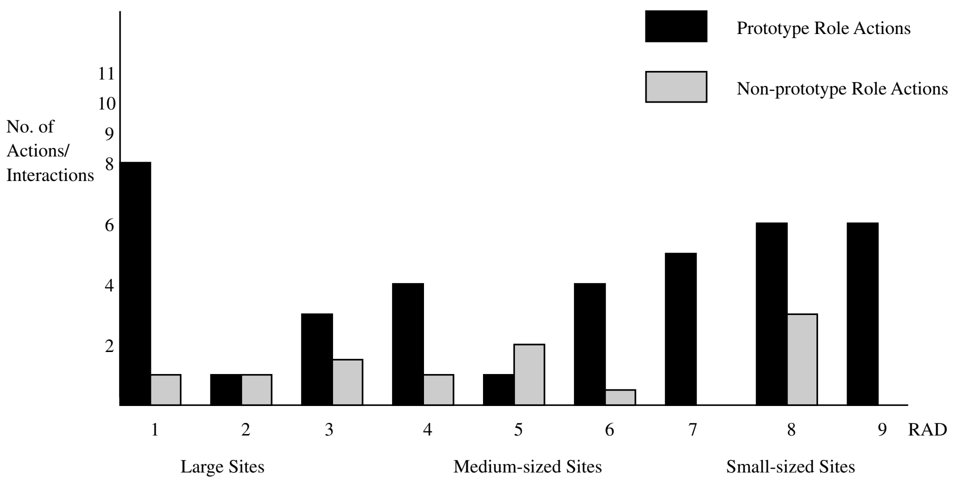

Figure 2. Histogram of actions for all roles.

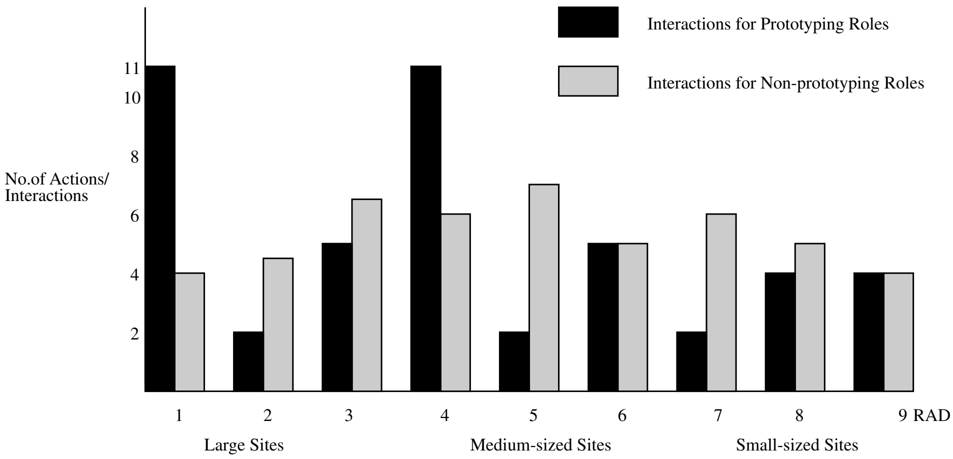

Figure 3. Histogram of interactions for all roles.

The prototyping roles contained actions ranging in number from 1 to 8 (Figure 2, dark shaded bars); for non-prototyping roles, the median number of actions ranged from 1 to 3 (Figure 2, lightly shaded bars). Seven of the prototyping roles therefore had a greater number of actions than the median for non-prototyping roles. RAD 5 was the only process where a greater number of actions was found for the median non-prototyping roles; one RAD (number 2) showed the same number of actions for both prototyping and non-prototyping roles. Inspection of the RADs shows, as would be expected, that the prototyping roles undertake extensive actions as part of the prototyping role. For example, actions such as recording user comments, modifying and extending prototypes, testing and validating prototypes and writing reports appear frequently across the nine RADs studied. One explanation for the wide range in actions between prototyping roles and non-prototyping roles is that some roles communicate and do nothing else. The relatively small number of actions found in prototyping roles for RADs 2 and 5 can be explained by the existence of Systems Design and Engineering roles which undertook some of the work normally associated with the prototyper (e.g., analysing requirements, modifying designs).
In terms of differences in the number of interactions, no clear pattern emerges in differences between prototyping roles and the median for other roles.

5.1.1. Statistical support for Hypothesis H1. From Figure 2, the mean difference between prototype role actions and the median number of actions for other roles is 3.05. This value includes the one exception (that of RAD 5), where the median number of actions is greater than that of the prototype role. Inspection of the individual RADs indicates that across all of the nine RADs, only three of the forty-two roles analysed had a greater number of actions than the prototyping role (i.e., 7.1%).

For interactions (Figure 3), the mean difference between prototype role actions and the median number of interactions for other roles was 0.1 in favour of the median interactions for non-prototyping roles. Thirteen of the forty-two roles analysed had a greater number of interactions than the prototype roles (i.e., 31%). While there may be support for the hypothesis based on the number of actions in a prototyping role being consistently greater than interactions therefore, the hypothesis relies on **both** actions and interactions in the prototyping roles being greater than in other roles. Hypothesis **H1** can not therefore be totally supported and we conclude that prototyping roles do not necessarily have more actions **and** interactions than other roles.

*5.2. Hypothesis **H2** data*

Hypothesis **H2** tries to clarify whether the larger the size of the organization, the greater the number of interactions undertaken by prototyping roles. To assess Hypothesis **H2**, we consider Figures 2 and 3 further. Prototyping roles in the three large systems engage in 11, 2 and 5 interactions respectively; the three medium-sized RADs engage in an identical number of interactions; the three small RADs engage in 2, 4 and 4 interactions. This data is illustrated graphically in Figure 4. In at least two cases, a great difference in the number of interactions occurs between small sites and large and medium-sized sites (sites 1 and 4). At these sites, large and medium-sized sites seem to require extra interaction on the part of the prototyping roles.

One explanation for this difference maybe that in some larger organizations, there are a larger number of roles involved in the prototyping process. Inspection of Table 2 does reveal that large organizations tend to include more roles in the prototyping process; in the three large RADs studied, a total of seventeen roles were identified, compared with thirteen in medium-sized developments and twelve in

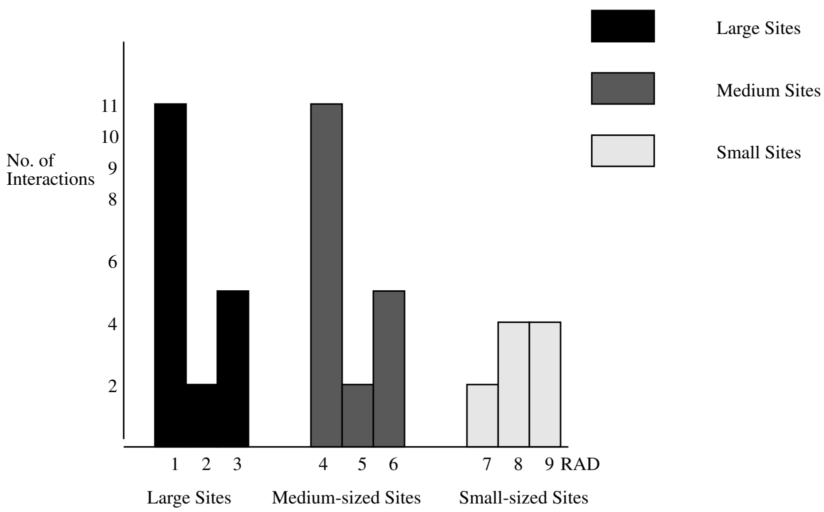

Figure 4. Histogram of prototyping interactions at all sites.

small developments. Inspection of all nine RADs also reveals a significantly higher level of complex interactions in large and medium-sized developments than in small-sized sites. Using the RAD notation, a complex interaction is one engaged in by more than two roles (as opposed to an interaction involving just two roles).

Another explanation for the difference between large and medium-sized sites and those of small-sized sites might be that in larger organizations, larger developments are more likely to involve higher numbers of staff in the prototyping process. Hence, by their nature, the size of the prototyping process creates the need for greater interaction. Larger organizations are more likely to have the available resources to devote to the prototyping process. Stricter lines of control and authority may mean lines of communication of the prototyper in carrying out the prototyping process are rigid, well-documented and have to be enforced as part of site standards. As well as identifying the reasons why large and medium-sized sites contain more interactions than small-sized sites, it is also important to establish why, in this study, large and medium-sized sites contain an identical number of total interactions. One explanation may be that, above a certain size of organization, levels of interaction are always similar. Below that threshold, (as shown by the three small sites), differences in the underlying process mean that levels of interaction do not need to be so extensive. This suggests that the three medium-sized sites analysed were above that threshold value.

5.2.1. Statistical support for Hypothesis H2. The mean number of interactions by the prototyping role for small sites (value 3.3) is just over half of that for the same role at medium and large-sized sites (value 6). It is also worthwhile analysing the effect on prototyping role interactions in RADs where the role of the prototyper has been partially usurped by the interactions of other roles. For the large-sized sites, the mean value of interactions would have been 7.7, had the prototyping role in RAD 2 not been partially usurped by the System Design role. For the medium-sized sites, the mean value of interactions would have been 3.25, had the prototyping role for RAD 5 not been partially usurped by the Engineering role.

For Hypothesis **H2**, a similar conclusion can drawn to that of Hypothesis **H1**. There is evidence to suggest that large and medium-sized organizations have greater numbers of interactions. However, it is not true to say that as a site increases in size, the number of interactions increases (large and medium-sized sites had identical numbers of interactions). As such, Hypothesis **H2** can not be totally supported. We conclude that it is not necessarily the case that the larger the size of the site, the greater the number of interactions undertaken by prototyping roles.

*5.3. Hypothesis **H3** data*

Hypothesis **H3** examines whether Coupling Factors for prototyping roles are smaller than Coupling Factors for other types of role. To assess Hypothesis **H3**, the Coupling Factor (CF) metric for each prototyping role and median CF value for other roles in the same process were calculated. These values are shown in Figure 5. The coupling factors in prototyping roles are smaller than the median CF value for other

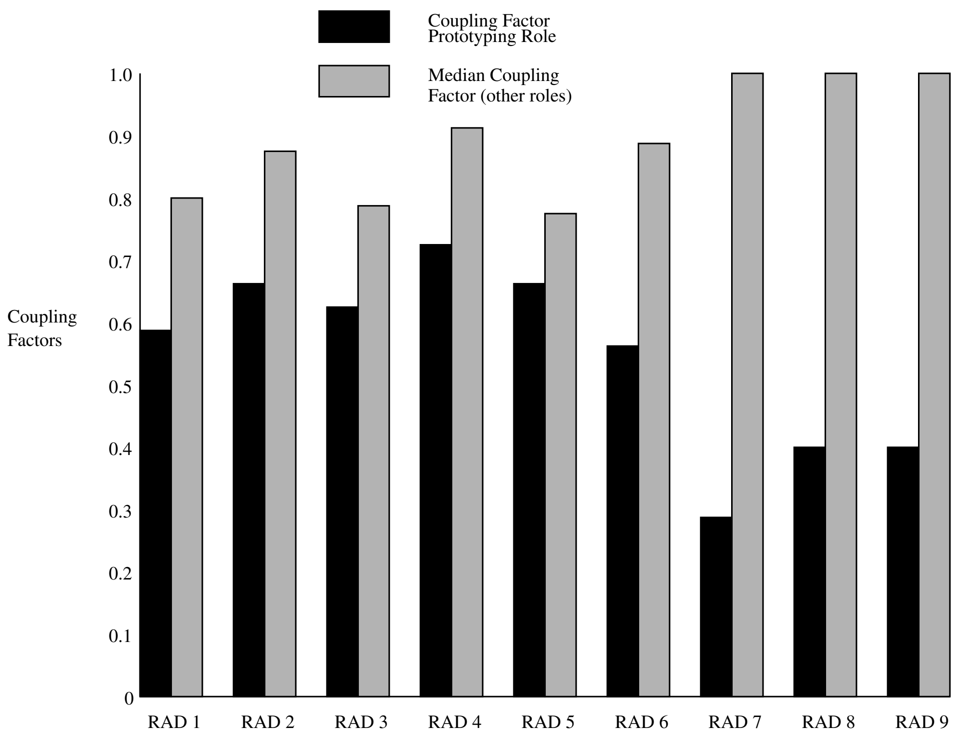

Figure 5. Coupling Factors for the nine RADs investigated.

roles in every case. This can be explained by the fact that every prototyping role analysed contained a more even balance between actions and interactions. Figures 2 and 3 illustrate this point. The Marketing role, present in two of the RADs (2 and 6), contained zero actions. Inspection of the RADs revealed that only four of the forty-two roles had a CF lower than the prototyping role (representing 9.5%). Three of these roles were project management roles, the other was the Component Engineering role (RAD 2).

5.3.1. Statistical support for Hypothesis H3. From Figure 5, the mean difference between the CF's for prototype roles and the median CF's for other roles is 0.35. Inspection of the nine RADs revealed that fifteen of the forty-two roles (35.7%) contained no actions. Thirteen of those fifteen roles were either Customer or End-user roles. At small sites, where there tended to be fewer roles, the RADs contained both a Customer and an End-user role. The median CF value therefore tended to be very high at these sites. At medium and large sites, the Customer and End-user roles tended to perform more actions of their own (but only marginally more, it should be stressed).

Hypothesis **H3** is supported and we conclude that Coupling Factors for prototyping roles are smaller than Coupling Factors for other roles. Interestingly, the median CF values at all the small sites was one; this implies that the non-prototyping roles in small-sized sites tended to engage in interactions only. The CF values for prototyping roles in small sites were all less than any of the other CF's. This suggests that prototyping roles in small organizations tended to engage in more actions than interactions, suggesting a greater level of autonomy in those prototyping roles carrying out the prototyping process (autonomy not afforded the prototyping roles in large and medium sized sites). Inspection of the RADs showed that at small sites, six of the nine non-prototyping roles had CF's of one (i.e., those of the Customer and End-user at each of the three sites).

*5.4. Hypothesis **H4** data*

Hypothesis **H4** examines whether a higher number of interactions takes place between prototyping roles and project management roles than with any other roles. The key to success of the prototyping process (and perhaps a project) can be the relationship between a prototyper and the project manager. In terms of time and cost pressures, this inter-relationship can be crucial. Table 3 shows, for each of the nine RADs, the number of interactions between each of the prototyping roles and project management roles, together with the median number of interactions between other roles and project management roles. For clarity, each RAD is also annotated with its site size—large (l), medium (m) or small (s) as given in Table 1. The two zero values are for prototyping roles in RADs 2 and 5, both of which interact with the project managing role via an intermediate role. In the case of RAD 2, it is via a System Design role and in RAD 5 via an Engineering role. Interestingly, it can be seen from Table 3 that in the large and medium-sized sites, there tends to be more interaction between the prototyper and project management role than in the small sites. This suggests that, in the former, there is a higher degree of management involvement in the prototyping role. In small sites, the prototyper appears to have more freedom in carrying out its functions and hence does not need to communicate with the project manager role to such a large extent. It may also be the case that, being at a small site, the product itself may be smaller and hence, more manageable (understandable) by the prototyper concerned.

The level of communication between prototyper and the project manager in large and medium-sized sites may be due to a number of factors. Firstly, there may be extra emphasis and importance placed on establishing user requirements in large and medium-sized organizations, particularly if the developments in that organization require extensive investment of resources. Systems with a safety-critical aspect to their domain would also fall into this category (e.g., RAD 5). Secondly, the role of the prototyper in large and medium-sized sites may also have to be justified financially, requiring greater controls being placed on the prototyping process by

Table 3. Interactions between prototyping roles and project management

RAD no.	1 (l)	2 (l)	3 (l)	4 (m)	5 (m)	6 (m)	7 (s)	8 (s)	9 (s)
No. interactions	7	0	3	5	0	3	1	2	2
Median (other roles)	2.5	0	3	2.5	0	2	4	3	2

management; at large and medium-sized sites, there is also likely to be added pressure for success on the prototyping process in terms of future maintenance. Finally, large systems may just be too complex for a prototyper to get an overview of the whole project; hence there may be a need for higher-level communication between prototyper and project manager to achieve that understanding.

5.4.1. Statistical support for Hypothesis H4. The mean difference between the interactions for prototyping roles and the median for other roles is 1.33. In three of the RADs (1, 4 and 6), the prototyper does interact more with the project manager than it does with any other role. In three of the RADs, (at the three small sites), the prototyping role does not interact as much with the Project Manager as it does with other roles. One RAD ties in terms of interaction (RAD 3), and there are two RADs (2 and 5) with zero interactions.

Returning to the hypothesis, it is true that a higher number of interactions takes place between prototyping roles and project management roles than with any other roles, but this tends only to apply to large and medium-sized sites. Small sites exhibit different properties. Hypothesis **H4** can not therefore be supported totally, and hence we reject Hypothesis **H4**: it is not true to say that a higher number of interactions take place between prototyping roles and project management roles than between other roles and the project management role.

*5.5. Hypothesis **H5** data*

Hypothesis **H5** examines whether System Coupling Factors for large sites are greater than System Coupling Factors for medium and small-sized sites. Table 4 lists the System Coupling Factors (SCF's) for each of the nine RADs. Interestingly, values for the sites 7, 8 and 9 are smaller than any of the other SCF values. In terms of the hypothesis, it can be seen that medium-sized sites (RADs 4-6) have the highest mean SCF value followed by large sites (and then small sites). From the definition of the metric, this implies that either there are a relatively low number of actions or a relatively high number of interactions (or both) in these RADs.

Inspection of the three RADs reveals that sites 7, 8 and 9 do have a relatively low number of actions when compared with the other six RADs. Hypothesis **H5** can not therefore be fully supported, and we conclude that SCF's for large sites are not necessarily larger than those for medium and small sized sites. It is interesting to note that the highest SCF's were in the medium-sized sites. This suggests that at those sites, a lower proportion of actions were undertaken by roles. One explanation for this may be that, at medium-sized sites, there is neither the autonomy found at small sites nor the extent of control structures found at large sites; this leads to a relatively low number of actions compared to interactions at those sites (and hence

Table 4. System Coupling Factors for the nine RADs

RAD no.	1	2	3	4	5	6	7	8	9
SCF value	0.73	0.81	0.73	0.83	0.81	0.79	0.72	0.56	0.52

higher SCF values). In the following section, some of the issues raised in the five hypotheses are discussed.

6. Discussion

From the original five hypotheses stated in Section 2, the following summary can be made. Only Hypothesis **H3** was supported: coupling factors for prototyping roles are smaller than coupling factors for other types of role.

Hypotheses **H1**, **H2**, **H4** and **H5** could only be partially supported. Prototyping roles do not necessarily have greater numbers of actions and interactions than other role-types (**H1**), although a hypothesis that prototyping roles have greater numbers of actions than other types of role would be supported. There is a link between the size of the site and the number of interactions undertaken by prototyping roles (Hypothesis **H2**) but not of sufficient strength to fully support the hypothesis. Interactions between prototyping roles and project management roles are not necessarily greater than any other roles (**H4**); however, this does not give a true representation of what is actually occurring: a hypothesis that interactions between prototyping roles and project management roles are greater than with any other roles for large and medium-sized developments would be fully supported. Finally, system coupling factors for large sites are not greater than system coupling factors for medium and small-sized sites (**H5**). Data showed that medium-sized sites had the highest SCF's.

A number of issues have to be considered in terms of strengthening the results in this paper. One criticism that can be levelled against research of this kind is that the systems studied are similar in size and application domain. However, the systems investigated in this paper represent a mix of projects of different sizes and application domains; our results are more generalisable as a result.

Another criticism of this study relates to prototyping role actions (particularly with respect to Hypothesis **H1**). It could be claimed that only the interactions of the prototyping role are of interest. However, it must be borne in mind that every action is a potential interaction if responsibility for that action is subsequently delegated to another role. This delegation is quite possible in the relationships a prototyper has with the End-user and Project Manager roles alike.

A further criticism that could be levelled at the approach of applying measures to RADs is that the RADs themselves do not represent a true picture of the process, that is, inaccuracies in representation could overshadow the results. However, the RADs were constructed over a period of two to three years using documented investigation techniques; clarification was obtained on any aspects of the RADs that the investigator felt needed clarification. This reduced the threat of inaccuracies and misunderstanding in constructing the RADs. This also reduced the threat posed by too much emphasis being placed on interviews with the prototypers themselves, which could lead to a biased reflection of the role of the prototyper.

The RADs modelled represent a static view of the prototyping process. For simplicity of the case-study, data relating to the number of occurrences of a particular interaction was not collected. Collecting such data would have required a great deal more time and effort, with the possibility of counting errors influencing the analysis.

The static view of the prototyping process provides an insightful and useful means of understanding the behaviour of the prototyper and other roles during the prototyping process. It is also noted that, from inspection of the interview transcripts, the views on the impact of doing prototyping varied enormously across the eight organizations, and this is reflected in the RADs themselves. For example, RAD 3 had no project managing function; instead it had a design managing role, indicating that project managers played no part in the prototyping process. In a study of industrial type situations, inconsistencies of this sort are always likely to appear.

7. Conclusions and future research

In this paper, an empirical investigation has been carried out into nine prototyping processes for different application domains at eight sites. Five hypotheses were investigated: these related the prototyping process to features of other roles such as extent of interaction during the prototyping process, the effect site size has on the prototyping role and the dependence of the prototyping role on other roles in the prototyping process. Results indicated a strong similarity between large and medium-sized sites in terms of interactions and behaviour. The prototyping process in small sites was found to be significantly different to large and medium-sized sites in both interaction and actions suggesting that the informal nature of prototyping in small developments leads to less coupling between the roles involved.

In this paper, features of actions and interactions from the RADs have been examined. One intended avenue for future research is to analyse features common to particular application domains. For example, between those of information systems and other types of application domain such as network applications. This may reveal that certain application domains exhibit particular coupling (interaction) trends. Further research will also focus on development and analysis of metrics to capture other features of the RADs studied in this paper. One challenge would be to assess the nature of cohesion within the roles of a RAD (i.e., the extent to which actions within roles are related). Also of interest would be to study the relationship that cohesion has with coupling across RADs.

Acknowledgments

The authors acknowledge with thanks the useful comments of the anonymous referees, inclusion of which has greatly improved the paper. They also express their gratitude to John Wiley and Sons Limited, for allowing Figure 1 to be reproduced from: Business Processes: Modelling and analysis for reengineering and improvement, M. Ould, 1995. Finally, we note that the RADs used as a basis of this study are available to view on-line at: www.dcs.bbk.ac.uk/˜steve/rads.html.

References

Basili, V.R. 1996. The role of experimentation in software engineering: Past, current, and future. *Proc. 18th Int. Conf. Software Eng.* pp. 442–449.

Basili, V.R., Briand, L.C., and Melo, W.L. 1996. A validation of object-oriented design metrics as quality indicators, *IEEE Trans. Software Eng.* 22(10): 751–761.

Brooks, J. 1987. People are our most important product. *Software Engineering Education*, Gibbs, E. and Fairley, R. (eds.), Springer-Verlag.

Card, D. 1995. The RAD fad: Is timing really everything? *IEEE Software*, Jan.: 19–22.

Carmel, E. 1995. Does RAD live up to the hype? *IEEE Software*, Jan.: 24–26.

Chen, L. 1997. *An Empirical Investigation into Management and Control of Software Prototyping*. Ph.D. dissertation, Department of Computing, Bournemouth University.

Coleman, G. and Verbruggen, R. 1998. A quality software process for rapid application development, *Software Quality J.* 7(2): 107–122.

Davis, A. 1992. Operational prototyping: A new development approach, *IEEE software*, pp. 70–78.

Fenton, N.E. and Pfleeger, S.L. 1996. *Software Metrics, A Rigorous and Practical Approach*, International Thomson Computer Press.

Gordon, V. and Bieman, J. 1993. Reported effects of rapid prototyping on industrial software quality, *Software Quality J.* 2(2): 93–108.

Gordon, V. and Bieman, J. 1995. Rapid prototyping: Lessons learned, *IEEE Software*, Jan.: 85–95 (reprinted in: Software Project Management: Readings and Cases, First Edition by C. Kemerer, Irwin Publishing, 1997).

Handy, R. 1976. *Understanding Organisations*, Penguin.

Harrison, R., Counsell, S.J., and Nithi, R. 1998. An evaluation of the MOOD set of object-oriented software metrics, *IEEE Trans. Software Eng.* 24(6): 491–496.

Lichter, H., Schneider-Hufschmidt, M., and Zullighoven, H. 1994. Prototyping in industrial software projects: Bridging the gap between theory and practice, *IEEE Trans. Software Eng.* 20(11): 825–832.

Lorenz, M. and Kidd, J. 1994. *Object-Oriented Software Metrics*, Prentice Hall Object-Oriented Series.

Miers, D. 1994. *Business Process Re-Engineering: Myth and Reality*, Kogan Page.

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

Ould, M. and Roberts, C. 1986. Modelling iteration in the software process. *3rd Int. Software Process Workshop*, Colorado, USA.

Phalp, K. and Counsell, S. 1997. Counts and heuristics for static process models. *Proc. ICSE Workshop on Process Modelling and Software Evolution*, Boston, USA.

Phalp, K. and Counsell, S. 2000. The application of metrics to industrial prototyping processes: An empirical study. *Software Process Technology*, Ed. R. Conradi, Springer-Verlag Series, 1780: 239–248.

Phalp, K. and Sheppered, M. 1999. Analysing process models quantitatively. *Empirical Assessment in Software Engineering (EASE'99)* Keele, Staffordshire, UK.

Reilly, J. 1995. Does RAD live up to the hype? *IEEE Software*, Jan.: 24–26.

Keith Phalp graduated from the University of Kent with a BSc in Mathematics in 1986. Having taught mathematics, he returned to further study, gaining a Masters in Software Engineering (1992) and a PhD in Computer Science (1995), both from Bournemouth University. He is a Chartered Mathematician and a Member of the Institute of Mathematics and its Applications. Dr. Phalp's research concentrates on the empirical study of the software development process, and specifically upon the efficacy of requirements techniques. He currently teaches software design and process modelling at Bournemouth University, where he is the course leader for the Masters in Computing.

Steve Counsell graduated from the University of Brighton with a BSc in Computer Studies in 1987. He gained an MSc from the City University in 1988 and worked in industry as a developer for three years before undertaking a PhD in Computer Science at Birkbeck College, University of London, which he recently completed. His research focuses on the empirical study of the software process, in particular that related to object-orientation and requirements capture. Since 1999 he has been a Lecturer in the School of Computer Science and Information Systems at Birkbeck.