Comments On The Use Of RADs In Case Studies.

Peter Kawalek,
Informatics Process Group, University of Manchester.
Tel: +44 61 275 6183 Email: pk@cs.man.ac.uk

Introduction.

Role Activity Diagrams (RADs) originated through the study of coordination by Anatol Holt [Hol83]. They have of course been well described in IOPener before now [Oul92]. This article is a collection of observations based upon the author's use of RADs in work with insurance underwriters and software developers. Although intuitively simple and easy to read, RADs are a powerful modelling technique which use a number of notational primitives to express very complex process behaviour. In simple terms, the harder you work RADs the more you can express with them.

Usage.

In order to be clear about the purpose to which RADs are put we can borrow some ideas from the software process research community. Dowson and Fernstrom [DF94] usefully identify different "process domains:"

- The domain of *process definitions* which contains "characterizations of processes or fragments of processes, expressed in some notation, in terms of how they could or should be performed." Therefore this is the domain of RADs, Data Flow Diagrams and other graphical notations. It is also the domain of process modelling languages (PMLs) which are "..definitions that are enactable."
- The domain of *process performance* which "encompasses the actual activities or actions conducted by human agents." This is the domain of the workplace, where people carry out a range of activities which may or may not involve the use of technology and may or may not conform to a process definition.

To these two domains we can add a third; the domain of *process support*. This is concerned with the services that take place in IT systems to support and enable the greater process performance[1].

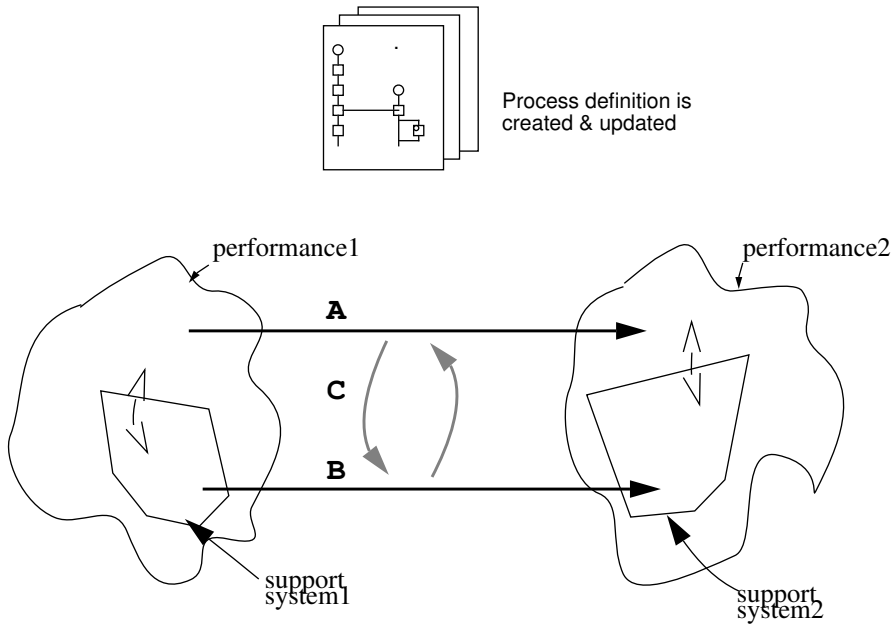
RADs can be used to describe an existing process performance in order that it can be better understood, analysed and changed. A new pattern of process performance which results can itself be modelled and changed. Thus, ever since the first version of our process modelling method [Kaw92] right through to our current SERC funded work [Rob93], the Informatics Process Group have emphasised that process modelling can be as continuous as the management function itself.

Process support can be seen as a system within and contributing to the greater system of process performance. RADs can be used to help understand the relationship between an existing support system and process performance. They can contribute to the development of a support system or a more general performance and to an appreciation of the implications of this development. Figure 1 is an expression of the interdependence of a process and its support. Changes which are made to a process performance may affect that part of it which is in the domain of the support system. The converse is also true. Therefore the purpose to which we set RAD modelling, and indeed process modelling more generally, is to do with understanding and managing the relationships between systems (existing support, exist-

1. This third domain differs from that proposed by Dowson and Fernstrom. They suggest a domain of process definition enactment which "is concerned with what takes place in a process centred support environment to support process definition-governed process performance. " For the purposes of this article this was thought to be restrictive . We can think of process definition enactment as contributing to the more general domain of process support.

ing performance etc). This is not best described as systems analysis but as *systems synthesis*.

Figure 1: The development of process performance and process support.



A. To change the process performance.

B. To change the process support.

C. Inevitable and useful cross-currents

Concurrency.

A vertical line in a RAD represents a state of the role body [Hol83], [Oul92]. To link activities vertically through a state is to say that the uppermost activity will precede the state in which the lower occurs. This is quite a powerful statement! If we are modelling process performance it means that we are confident of some kind of dependency relationship between activities within the real world.

The first stage of the author's process modelling work in case studies has always been to sit down with staff and discuss their work. Frequently, and helpfully, they often describe their activities as though they were doing the job at that time. Thus, they tell me "When I have finished doing *X*, I start doing *Y*." One of the first problems in building a RAD model is to try and understand the reasons why *X* precedes *Y*. Often there are no reasons which can be distilled into satisfactory pre-conditions to a state. When this is the case then a part refinement is used to denote potential concurrency (or perhaps more tellingly, *no ordering*) between the activities concerned.

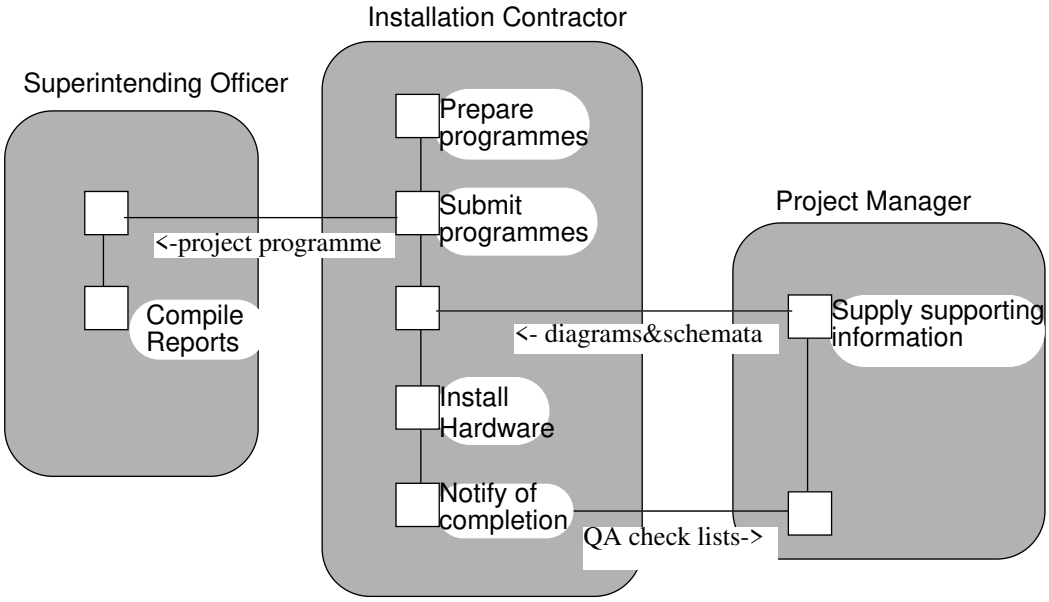
If you pay close attention to the reasons why activities apparently precede each other, then you tend to use the part refinement primitive a lot. This means that you tend to end up with a fatter role which captures and conveys the flexibility and contingency of work much more effectively. This can be very helpful when discussing the way work is done, or redesigning processes and is perhaps critical in helping to avoid the development of over-prescriptive and constraining process support.

Organisational Roles.

A RAD role has been described as being "..comprised of a set of activities which, taken together, achieve some particular goal" [Oul92]. Despite all that has been said about RAD roles not being akin to organisational roles (i.e. job titles) it is still common to encounter this misconception. Perhaps this is because to describe organisational roles through RADs is intuitively obvious. It may even be *useful* in some circumstances. For example, if you wish to update the bland text and flowcharts of your existing procedures manual with something a little more animated, then RADs which describe familiar organisational roles might be just what you need. What you are likely to do in these circumstances is display the contribution of organisational roles within a functional breakdown of work. It is unlikely that you will want to show whole organisational roles unrestrained by functional context.

A very simple RAD fragment is shown at Figure 2. It describes part of a fictitious process for installing hardware in a manufacturing plant. Three roles are shown. These are named according to organisational job titles. What is particularly nice about a RAD built this way is that the interactions relate so closely to the real process performance. If you happen to have knowledge of a similar process you may be able to put names to the people involved and visualise the 'project programmes,' 'diagrams' and 'schemata' that they pass between them:

Figure 2: RAD Fragment (1).

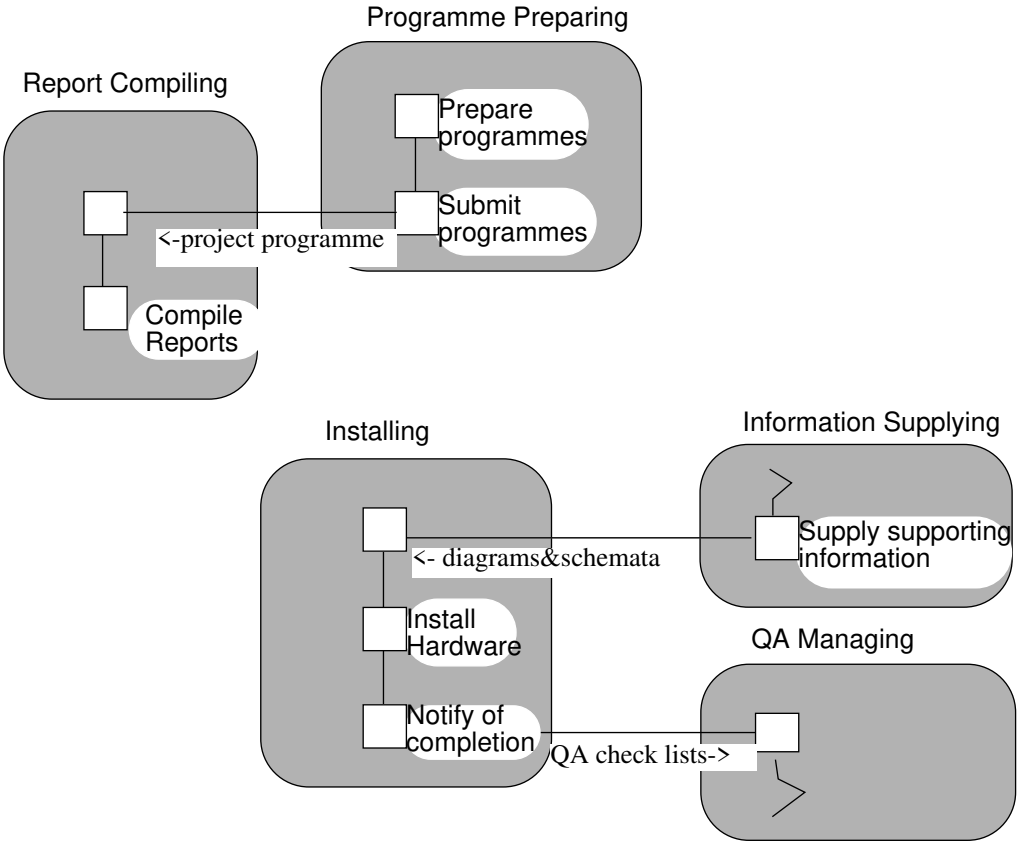


Working RADs Harder.

If we focus on RAD roles as logically related activities which contribute towards the fulfilment of a goal, then we can detach RAD roles from the organisational structures of the company in which they sit. In this way, with some effort, RAD roles can represent cohesive and decoupled modules of work. RADs then show the *what* of work and not the *who,* which is a dimension that can be treated separately. I am almost tempted to describe RAD roles of this kind as *normalised!*

The RAD fragment in Figure 2 (above) is very small and it is difficult to clearly show the impact of abstraction away from organisational job titles. Figure 3 attempts it. The zig-zag lines are used to denote incomplete role fragments:

Figure 3: RAD Fragment (2).



A RAD built this way is a more powerful modelling tool than a RAD built out of organisational roles. Why is this the case?

- *Expressing process performance:* A RAD which is detached from organisational context is more consistent with the view that work practices are very often contingent and rarely conform exactly to procedure manuals or process definitions [Fik82]. The non-organisational RAD is a more fluid mechanism than its organisationally bound counterpart. It is not a solution to the problems encountered when using formal representations (see, for example [Suc83]), and does not interpret work as a dialectical relation between rules and practice [WWK94]. However, these RADs are at least conformant with the view that consistent activity/ person relationships are very uncommon. For example, sometimes the Superintending Officer may compile reports, sometimes the Project Manager will do it.
- *Facilitating process redesign:* The isolation of process from organisational structure makes it easier to consider the organisational aspects of change projects. For example, a process redesign project may require that the report compiling role is undertaken, not by a Superintending Officer, but by a Project Manager. The organisational RAD fragment locks the activities in with the people who carry them out. The revised RAD fragment makes it easier to conceive of the many ways in which the logical groupings may be assigned to people.
- *Roles as reusable modules of work:* A RAD which is detached from organisational context makes it possible to consider roles as units of process which are reusable. This reuse is available in three dimensions. First, roles can be common to different processes. Secondly, roles within a single process can call

each other and thus reuse each other. Thirdly, when a relationship to organisational job titles is established the roles can be reused (e.g. both Project Manager and Superintending Officer may undertake 'QA Managing'). This is of more than academic interest. The concept of reusing roles is useful in the development of processes and process support. It may even prove to be essential to architectural approaches to reengineering such as that advocated in [Vea94].

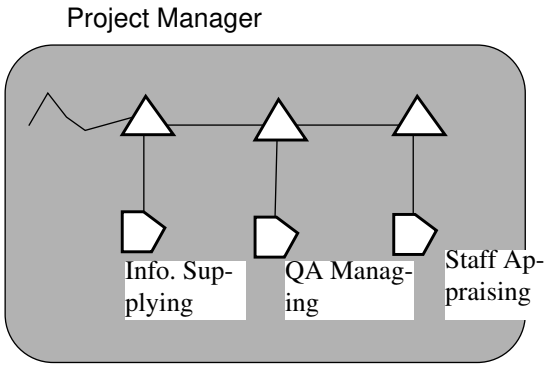
In one of the case studies the company were keen to use the RADs to assist the development of both process support and new organisational structures. Having detached the RADs from extant organisational structure the company is able to consider how the work portrayed in the roles can be assigned to people in a new project based team structure. The appropriate process support for each role can be thought of as being packaged within it. Therefore, as we assign roles to people we also know immediately the process support services that those people need. In this way we are using RADs as a mediator for "...exploration and description of the relationship between parts of a work system" [Kaw93].

Note that in abstracting away from an organisational context we have to consider what we mean by an interaction. In Figure 2, the first RAD fragment, the interactions clearly show people communicating with each other. The roles in Figure 3 can be assigned to different people. It may even be that the roles could all be assigned to the same person. Clearly then an interaction is best thought of as a *point of synchronisation* between two or more roles.

Organisational Roles Reconstructed.

Therefore, having detached roles from organisational context we may want to return to the issue and explore how roles map to people or job-titles. In one case study it has been decided that this shall be done by reconstructing organisational roles out of start roles as shown in Figure 4. This can be achieved by showing the start roles in a conventional ordered way, or without any ordering. The latter route is easier as it avoids the need for us to consider whether start roles have 'start-start' or 'finish-start' relationships with each other. It also creates a role which is akin to an *object* in so far as it describes a type (e.g. Project Manager) and the set of roles (or services) which can be undertaken *by* (not *on*) that type:

Figure 4: A Fragment Of An Organisational Role Object.



The Domain Of Process Support.

The primary concern of one case study was to develop a prototype support system. This was required to show benefit through reduced cost of work projects and popularity amongst its users. The project also involved some redesign of processes initiated in both the support and performance domains. Each role was regarded as a package containing its

own support services/requirements. The support system was developed by defining a process model which governed the availability of these support services. This was less constrained than the process performance model (thereby supporting flexible working practices) but shadowed it at a high level (which was described by IDEF0 diagrams). The result was very successful.

An interesting issue which arose during the development related to the use of Entity Relationship diagrams. The E-R formalism was used in the building of a database which was part of the overall support system. It was not used for anything else. Interestingly, the entities described in the E-R diagram were all derived from the RAD notes[1] and IDEF0 diagrams.

IDEF0

Reference has already been made to the fact that in one of the case studies RADs were used in conjunction with IDEF0. The use of the two notations together offers important benefits. Briefly, IDEF0 offers clear description of the data requirements of activities, their data output, the rules which govern their operation and the mechanisms needed for their support. They were used at a higher level than RADs, providing a structure into which the RAD roles can be fitted. Indeed, the RAD roles can be explicitly linked to the IDEF0 activities through calls.

By using IDEF0 to show a process at a high level and RADs to show detailed work activities we were able to assist our thinking about change on two levels. IDEF0 helped to structure our thoughts about radical change possibilities whilst RADs were useful for thinking about change at a more detailed level.

The Good & Bad Of RADs.

In conclusion, RADs have proved to be powerful in that they are simple to read, good at modelling how work is really done, able to express the modularity of work and can contribute to the redesign of processes in the performance and support domains. Against these must be the set criticisms made by some of the participants in the case studies: RADs give less information about what makes an activity happen than IDEF0, and are less well suited to expressing the overall flow and structure of an end to end process. Furthermore, RADs can be time consuming to build and to arrive at an optimal structure of roles is difficult.

Acknowledgements.

Thanks to Philip Veasey, F. Shams Aliee, Andy Garnett, David Redmond Pyle & Mike Ward for their comments during the course of case studies.

References.

[DF94] M. Dowson, C. Fernstrom; Towards Requirements For Enactment Systems. Proceedings of the Third European Workshop, EWSPT '94; Springer-Verlag Lecture Notes In Computer Science 772.

[Fik82] R.E. Fikes; A Commitment-Based Framework For Describing Informal Cooperative Work, Cognitive Science (6) pp. 331 - 347.

[Hol83] A.W. Holt, H.R. Ramsey, J.D. Grimes; Coordination System Technology As The Basis For A Programming Environment, Electrical Communication, Volume 57, Number 4, 1983.

[Kaw92] P. Kawalek; Introducing The Process Modelling Cookbook, IOPener Volume 1 Number 2.

1. The RAD diagrams were created using the RADitor tool developed by Coordination Systems Ltd. This allows notes to be attached to each RAD node. These notes can be used to describe conditions, data entities etc.

[Kaw93] P. Kawalek; Coordinating People & Technology Through Process, 2nd Information Technology and People Conference, Moscow, May 1993.

[Oul92] M. Ould; Process Modelling With RADs Parts 1 - 3, IOPener Volume 1 Number 5 to Volume 2 Number 2, 1992 - 1993.

[Rob93] I. Robertson; The Process Engineering Project, IOPener Volume 2 Number 2.

[Suc83] L.A. Suchman; Office Procedures as Practical Action: Models of Work and System Design, ACM Transactions on Office Information Systems 1 (4) pp. 320 - 328.

[Vea94] P. Veasey; Business Process Redesign in Multi-Business Enterprises, paper for BPR Academic Directions Conference, June 1994.

[WWK94] D.G. Wastell, P.R. White, P. Kawalek; A Methodology For Business Process Redesign: Experiences And Issues, Journal Of Strategic Information Systems, in press.