Comments on Process Working GroupWhite Paper on Analysis & Design Process Engineering

I have made a quick pass through the paper and have a few spontaneous comments. I have not had the time to make a systematic review, but I felt that I should give the comments that came to my mind.

However, let me first say that the effort is clearly interesting. I sympathize as you can well understand very much with what you want to do. The main question we must ask us is: Is this something that OMG should do. You have done so and come to the conclusion: yes. Personally, I have some doubts, or let me say that I am not convinced.

This is – together with the UML effort – very similar to what I suggested to Richard Soley that we should do back in 1991. We should have done this instead of working with the Analysis and Design reference model. I never believed in that work even if I did not want to criticize OMG in public (as some other friends did). Richard knows all this. Today, I see less need for this. However, I am open. I would like to hear what others think before I make up my mind.

- Why do we need to talk about a new language. After my first quick read I could not find anything that you cannot model with UML with some simple extensions of stereotypes and tagged values. This will be a UML specialization similar to the one on Process-specific extensions and Business engineering extensions. OK, if you so want you could call this UML-SPE, but I think it gives the wrong message to talk about an SPE.
- You should take a deep look into how Rational Objectory and now the Rational Unified process has modeled itself. We used a specialization of UML to do so. In my mind, there is almost nothing more that you are asking for here, than what we have been doing for over 12 years, so this experience should be drawn upon.
- Why don't you extend the job to develop a UML-SPE that can be used to design other processes in the *software development business* (an organization) such as the Support process, Requirement gathering processes. A process is a workflow definition, an instantiation is approximately a project, characteristic for these processes are that their pace of instantiation is different. For instance the Reqt gathering process may be instantiated for each major customer, whereas the development process is instantiated for each product.
- The paper indicate that you are not developing a UML-SPE but also a method for using it. Be careful!!! When developing ROP/RUP we did both. The method that you seem to have in mind looks a bit too functional – the workflow is decomposed in activities. Activities should be work performed by workers or similar. In this way we can specialize the process by specializing these workers.
- On terminology: use workflows (collaborations among workers and artifacts – work products) to describe important flows in the process. Combine workflows to get different kinds of complete flows within a development cycle.
- More on terminology: the term role is overloaded. We use the term worker and at the end of this PM I have attached what I am writing in the Unified Process book.

As said this was a very quick pass through the paper, which was very well written.

Thanks

Ivar

From the Unified Software Development Process book:
People fill many different positions in a software development organization. Preparing people for these positions takes education and pin-pointed training followed by careful assignment supported by mentoring and helpful supervision. An organization faces a substantial task when it moves a person from a latent "resource" to a particular position as a "worker."
We have selected the word *worker* to stand for the positions to which people may be assigned and which they accept. Some examples of worker types are use-case specifier, architect, component designer, tester, and user. A worker type is a role that an individual may play in software development. We did not select

the term role for primarily two reasons: it has a precise and different meaning in UML, and workers need to be very concrete; we need to think in terms of individual workers which are the positions taken by single individual persons. We also need to use the term role to talk about roles of a worker. A worker may play roles in relation to other workers in different workflows. For instance the worker component designer may participate in several workflows and in each participation the worker plays a role.
Each worker (a worker instance) is responsible for a whole set of activities, for instance, the activities involved in the design of a subsystem. To work effectively, he or she needs the information that is required to carry out those activities. He needs to understand his role relative to that of other workers. At the same time, if he is to get his particular work done, the tools he employs must be adequate. They must not only help him carry out his own activities, but also they should shield him from information that is not relevant to his position. To accomplish these objectives, the Unified process formally describes the positions – that is the workers -- that people can take in the process.
The figure below illustrates how individual people may be different workers in a project.

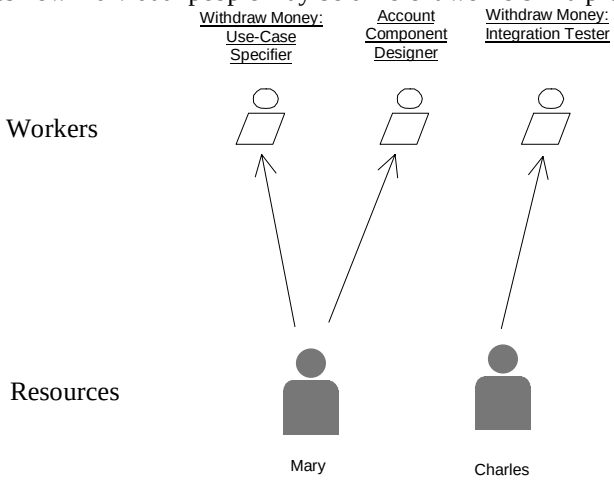

Figure 2-2: Workers and resources that "realize" them.

A worker may also be "realized" as a set of individuals working together, for instance an Architect worker may be realized as an architectural board.
Each worker has a set of responsibilities and performs a set of activities in developing software.
When allocating resources to workers in a project, the project manager needs to identify the competencies of his/her resources and match them with the required competencies of the workers. This is not an easy task especially the first time the Unified process is used. The skills of the resources (real people) must be match against the competencies specified by the different workers needed by the project. The competencies needed by some workers may be achieved through training while the competencies needed by other workers may only be gained through experience. For example, the skills needed to be a Use Case specifier may be learned through training while those of an Architect are typically learned through experience.
One individual person may be many workers during the life of a project. For example, Mary may start as a Use Case Specifier and later become a Component Designer.
When allocating resources, the project manager should minimize the hand-off of artifacts from one resource to another in a way that makes the flow of the process as seamless as possible. For example, the Use Case Designer for the Withdraw Money use case will acquire a lot of knowledge about the responsibilities of the class Account which means that he/she would be a logical choice to be the Designer of the Account class. The alternative would be to train a new person to take on this work, which could be done but it would be less efficient – loss of information, risk for misunderstandings, etc.