Rational Unified Process

Björn Gustafsson, John Smith
Process Development
Rational Software
February 26, 1999

Response to RFI

This note responds to the OMG, Analysis & Design PTF, *Software Process Engineering, Request for Information,* OMG document ad/98-10-08.

The Rational single point of contact regarding this response is:

> John Smith,
> Rational Software
> #638-650 West 41st Ave.
> Vancouver, B.C. V5Z 2M9
> Canada
> Email: jsmith@rational.com

References

[1] The Rational Unified Process, version 5.0, Rational Software, Cupertino, Ca. (1998).

[2] *The Rational Unified Process, An Introduction,* Philippe Kruchten, Addison Wesley Longman (1999) ISBN 0-201-60459-0.

Brief Summary of the Rational response

The Rational response is based entirely on the existing Rational Unified Process - its conceptual model, the modeling language used, and the boundaries it defines for 'process engineering'.

The Unified Process covers the full lifecycle of development including engineering activities for Business Modeling, Requirements Capture, Analysis & Design, Implementation, Test, Deployment, as well as supporting activities for Configuration and Change Management, Project Management and Environment.

The Unified Process is modeled entirely within the boundaries of UML. UML stereotypes are used to define the modeling concepts used in process modeling.

The Unified Process is a *process framework*. This means two things: it provides a framework within which *any* software development process may be defined; and it provides a standard process 'out-of-the-box'. Capabilities of the framework allow the standard process to be extended to describe the specific process used by a particular development organization or in a specific project.

In every respect the process model, and the principles for its modeling, is equivalent to 'a design model' defined in UML and the principles for object-oriented design applied in its modeling.

An important conclusion of our work so far is that we do not see the need to standardize a new set of CORBA interfaces or XMI persistence format for process modeling. While a standard profile/extension of UML should be defined, there is no need that we currently see to define a process repository that is more specific than a general-purpose OA&D facility or storage format.

Specific areas

This chapter identifies areas where there are differences between our response and the ideas presented in the RFI.

'Logical model' vs. 'dynamic model'

Modeling of the Unified Process is limited to describing what we call the 'process model', which is a logical model of process used for software development - as opposed to the 'dynamic model', which describes the instances of process elements such as specific instances of artifacts, workers, etc. as elements of a specific project.

The 'process model' describes the *structure* of 'a process'. The goal with such a description is to describe the structure of the textual contents of the Unified Process, which is presented in the form of a web site. Hence, the 'dynamic model' aspect of process engineering falls outside what is currently covered by the Unified Process.

The creation of a 'dynamic model' may be regarded as an instantiation of a (logical) 'process model': a 'process model' describes the 'design' of process whereas a 'dynamic model' describes the instantiation of a 'process model' in a specific context of a project. Hence, the 'process model' constitutes the 'blue-print' for the 'dynamic model'.

The nature of the 'facility'

The RFI assumes that the facility for software process engineering will be used solely for the purpose of creating models of actual projects, and includes both instances of system elements as well as instances of resources as elements of the meta-model.

This response does not take process engineering quite that far, it stops at the creation of a **logical model** of process, called a 'Process Model'. By limiting itself to creation of this model only, the language used for modeling of the Unified Process may be constrained to using only *UML and its defined extension mechanisms*.

By limiting the responsibilities of the facility to only supporting modeling of the 'process model', the facility can provide the foundation for many different process modeling objectives.

'Project Profile Parameters'

A number of questions in the RFI imply that there can exist a set of 'project profile parameters' to support the characterization of processes.

Our response does not include the identification of such a set of parameters. Project profile parameters as a **concept** should be supported by the facility, but the actual set of parameters and associated values should not be dictated by the facility. This is equivalent to UML's tagged values as an extension mechanism. It may be possible to go as far as identifying core parameters (e.g. project size, domain), but not enumerating values. It should also be possible to add parameters.

Hence, the facility should support the definition of *any set* of project profile parameters, so that the specific context of a particular facility can utilize a particular set for the purpose of that facility.

Answers to questions

This section gives answers to questions asked in section 4 of the RFI.

4.1. General Questions raised by the White paper

4.1.1. Is this something OMG should be doing?

Yes – it seems natural, having defined UML as process-neutral, to create a framework for its use. This is not to underestimate the difficulty of making this widely applicable (and acceptable).

4.1.2. Should the focus ....?

There is limited value in covering only A & D. We believe that real value is added when the facility can support the whole lifecycle. It is important to recognize that a process that addresses just the narrow scope of analysis and design does not necessarily scale to cope with the production and deployment of working software systems - which has been our focus. Other issues, such as Project Management, Requirements Management and Configuration Management, assume different dimensions in a full lifecycle context.

4.1.3. Should the facility specify work products .... ?

The logical model(s) should specify the work products. The facility should also provide the language for the specification of the work products. However, we believe that any future OMG standard should not itself specify (and so constrain) the work products of a process.

4.1.4. Do the following need to be addressed since the UML does not address them?

- *A minimum set of work products*:

The logical model should specify what is mandatory and what is optional.

- *Version management*:

  If this refers to the work products of the process, then the logical model should specify, as part of the total lifecycle coverage, how configuration management of these is performed.

- *Traceability*:

  (it is here assumed that ‘traceability’ means traceability between related work products)

  Yes, with our approach it is already supported by UML, in the form of stereotypes.

- *Phasing*: ?

  The logical model should specify generically how a project should progress over time, and the expectations for the state of work products at important milestones, such as end of phase.

4.2. Proposed clarifications to the White Paper Concepts

4.2.1. Is the Concept described in the White Paper correct?

- *a meta-model*:
  Yes – this aligns with our notion of a conceptual model.

- *models to be stored using the meta-model*:

  Yes – this aligns with our notion of a logical model (or set of these).

- *a facility to build processes for specific projects by matching project profile parameters against the meta-model ....*:

  The perceived facility builds on the expectation that there can be defined a universal set of project profile ‘parameters’ that apply to *all* process definitions. This is not likely to occur. However, ‘project profile parameter’ should be defined as a modeling concept and thus enable their definition in a particular context (like tagged values in UML).

4.2.2. What other services should be expected from such a facility?

We have not considered this question at this stage.

4.2.3. Is meta-model sufficient to support ‘plug-and-play’ ....?

No, the conceptual model does not have enough in it to enable that. Our approach suggests that using the interface specifications of activities and the associated artifact (work product) state machines, in the logical model, it *may* be possible to determine if process modules can be swapped out and replaced.

4.2.4. Should there be a registration/certification process .... ?

Unclear what this means. It would certainly be useful to automate the ability to check against an assessment standard such as CMM or ISO 15504 and predict a potential maturity rating, and to check compliance with any future OMG standard. There would need to be some clear return on investment to justify the overhead and bureaucracy of formal certification/registration.

4.2.5. Should the facility have the capability to provide standard patterns .... ?

This question is related to the project profile parameters. In particular contexts, the definition of sets of project profile parameters can be envisaged, from which standard patterns can be produced and stored for reuse. It would be useful to have some default configurations to 'prime the pump'.

4.2.6. What should the process and profile components used by the facility be?

We have no response at this stage.

4.2.7. Should there be a language? ...a graphical notation ....?

The language *is* UML. Hence, we don't need a different language.

We define stereotypes to denote our process modeling concepts, which may be graphically represented in any tool that supports UML.

4.2.8. Are the UML and MOF sufficient to ....?

Yes, they are sufficient.

4.2.9. Terminology correctness and completeness?

The Unified Process defines its own terminology, which is documented in [2]. In particular chapter 3 of this book introduces the fundamental elements of the conceptual model.

The meta-model used for modeling of the Unified Process resembles the meta-model outlined in the OMG White Paper. The OMG meta-model is wider in scope since it takes into account the dynamic aspects of process, but the concepts in the overlapping portion agree well.

4.2.10. How can the facility be used to bind to CAPE and CASE tools? (how can the process definition and UML models be related?)

The fundamental level of binding between process definition and UML models occurs at the meta-model level, where the Unified Process identifies UML *model* and *model elements* as the two variants of underlying representations of artifacts. At the next level of binding the Rational Unified Process, as an instance of process, defines the exact set of models and model elements whose development are described by the process. At this level, UML models and model elements may be stereotyped, in the CASE tool, to denote artifacts defined in the process described in the CAPE tool.

Hence, our response is that process definition should be related to UML models as the most primitive level, to enable the binding, and that UML standard extension mechanisms should be utilized to describe bindings at higher-order levels.

4.3. Roadmap recommendations

We suggest two separate RFPs are created and issued in sequence:

First RFP: Limit the capabilities of the requested facility to only include those that *enable* process modeling, i.e. modeling of the logical model of process. Since we envision that the standard definition of UML provides sufficient support for process modeling, we recommend that this RFP should require responses to be expressed as UML extensions using only the extension mechanisms defined in UML.

Second RFP: Based on the facility defined based on the first RFP, a second RFP should specifically request extensions to the defined facility to include production of the dynamic model of process.

The rationale for this roadmap is that the first RFP will strive to reach consensus on modeling principles and language and at the same time capitalize on the effort spent on standardizing UML as well as the maturity of UML-enabled design tools. This step will satisfy every possible stakeholder in ‘process’ whether it is for modeling of process dynamics, description of process structure, or for other purposes.

The second RFP will address in isolation the issue of modeling of ‘project dynamics’. This area of process engineering is wide and currently not well defined. By institutionalizing process modeling as per the first RFP, a context is created which bounds the problem of ‘dynamic modeling’ and constrains its possible solutions.

It is possible to envisage a third RFP that addresses *enactment* as the final step in bringing the dynamic model to life.

4.4. Future Needs

No comments.

4.5. Existing Implementations

See [1].

4.6. Standards

No comments.