

An Overview of the RUP as a Process Engineering Platform

Bruce MacIsaac

IBM Rational Software

Abstract. The right software development process for a particular project depends on many factors, including size of project, formality, technology, applicable techniques, and development philosophy. The Rational Unified Process® (RUP®) enables a suitable software development process to be configured from a set of process components, and to be extended using process plug-ins.

1 Introduction

The Rational Unified Process (RUP) is a commercial software development process framework. The RUP is composed of:

- **Best practices.** The RUP includes a library of best practices for software engineering, covering everything from project management to detailed test guidance.
- **Process delivery tools.** RUP is delivered using web technology, allowing it be integrated with other software development tools, and making it easily accessible to developers.
- **Configuration tools.** The RUP is made up of components and plug-ins, that can be selected and configured to meet the needs of the project.
- **Process authoring tools.** An organization can extend or modify the RUP by creating its own plug-ins using the Rational Process Workbench product.
- **Community/Marketplace.** The Rational Developer NetworkSM(RDN) provides a place for process engineers in the software development community to share their process extensions.



Fig. 1. RUP is a Process Framework consisting of process delivery tools, configuration tools, process authoring tools, and a community/marketplace of process plug-ins.

The focus of this paper is on the configuration and process authoring capabilities of the RUP framework. Information on other aspects of the RUP product can be obtained from the IBM Rational website at <http://www.rational.com>.

2 Configuring RUP

2.1 Selecting Components and Plug-ins

The RUP process framework contains a vast amount of guidance, artifacts, and roles. Because no project can use all of these artifacts, you need to specify a subset of the RUP to use for your project. This is done by selecting or producing a RUP Process Configuration, which constitutes a **complete process** from the perspective of a particular project's requirements. You can use one of the ready-made configurations as is, use a ready-made configuration as a starting point, or create a process configuration from scratch.

To understand how to build a RUP Configuration, you need to understand the following concepts.

- A **RUP Process Component** is a coherent, quasi-independent “chunk” or module of process knowledge that can be named, packaged, exchanged, and assembled with other process components.
- A **RUP Library** is a collection of Process Components out of which a set of RUP Process Configurations may be compiled with RUP Builder. New Process Components can be added to a RUP Library by means of RUP Plug-Ins.
- A **RUP Base** is a collection of Process Components meant to be extended by applying plug-ins to generate RUP Process Configurations. It resides in a RUP Library.
- A **RUP Plug-In** is a deployable unit for one or several Process Components that can be readily “dropped” onto a RUP Base to extend it. A RUP Plug-In can be compiled into a single physical file, allowing it to be moved around and added to a RUP Library with a compatible RUP Base.
To explain this via a simple analogy, a **RUP Plug-In** is a “precompiled” **RUP Process Component**, ready to be “linked” into a **RUP Base** to create a **RUP Configuration**.

A RUP Process Configuration is produced using RUP Builder. RUP Builder is shipped with a number of predefined configurations, and you can create additional configurations as needed. Based on what plug-ins you choose, you can make the process smaller or bigger, and you can make it address the technology, domain, and tools relevant to your project or set of projects. You can also choose how formally you want to work — for example, whether to use more comprehensive document templates or lighter templates suitable for smaller teams. Once you have defined which plug-ins belong to a configuration, and which process components within those plug-ins and RUP Base you want to use, RUP Builder validates that the selected process components are compatible and publishes a RUP Configuration Web site from your configuration.

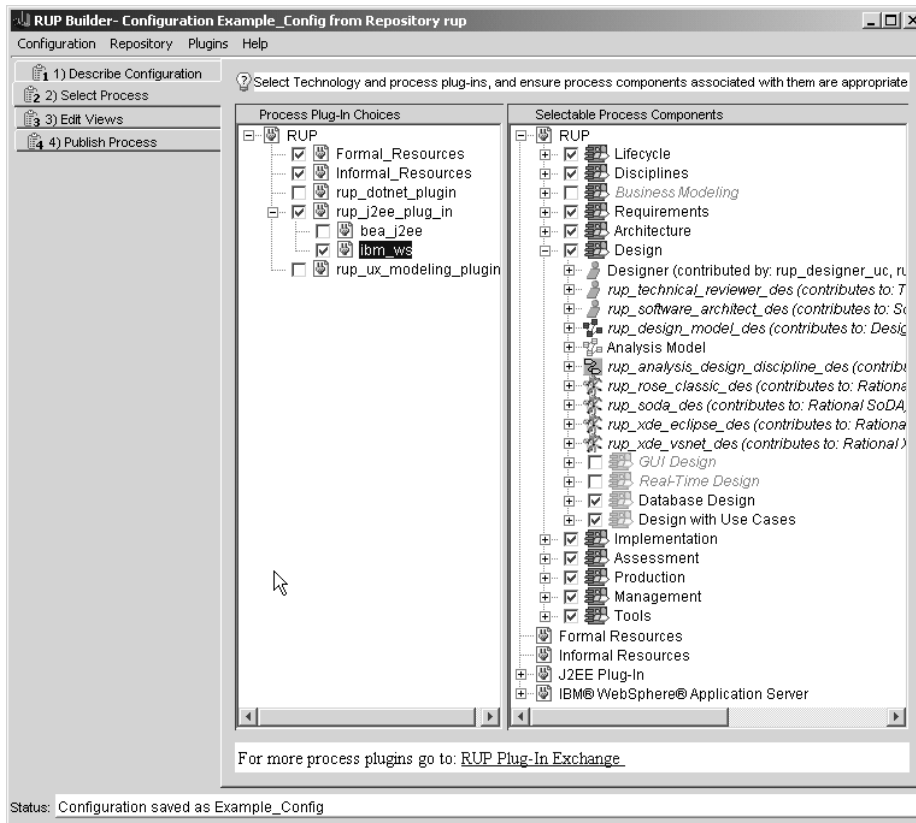


Fig. 2. RUP Builder allows you to select plug-ins and process components, and then publish the resulting configuration.

Using RUP Builder, you add more content to your RUP library by importing RUP Plug-Ins. This content can then be used to build RUP Process Configurations that better fit the needs of your project. There are many companies packaging their know-how in RUP Plug-Ins and making them available to RUP users through the RUP Exchange, a subsite on the Rational Developer Network (RDN) where RUP Plug-Ins and other process-related material are made available to the user community.

Producing a RUP Process Configuration takes only a few minutes. The tricky thing is to know what to select; understanding what is available for selection may take some time. As a first-time user, you should initially focus on reviewing the already-defined process configurations, and use them as a starting point.

2.2 Defining Process Views

Your RUP Process Configuration contains the parts of the RUP Process Framework that is applicable to your project. More than likely, however, all of that content is not

applicable to you as a project manager, analyst, architect, developer, tester, or configuration manager. For this reason, depending on your role and responsibilities, you will want your own window or view into your RUP Configuration, or what we call Process View.

A Process View is a role-based or personalized tree control containing links to desired elements in your RUP Process Configuration, as well as links to files or URLs external to your configuration. Process Views are created in RUP Builder. Each team member can further personalize Process Views using MyRUP, that is, the Web browser used to browse RUP. Figure 3 shows Process Views in MyRUP (in this case, a view for an analyst).

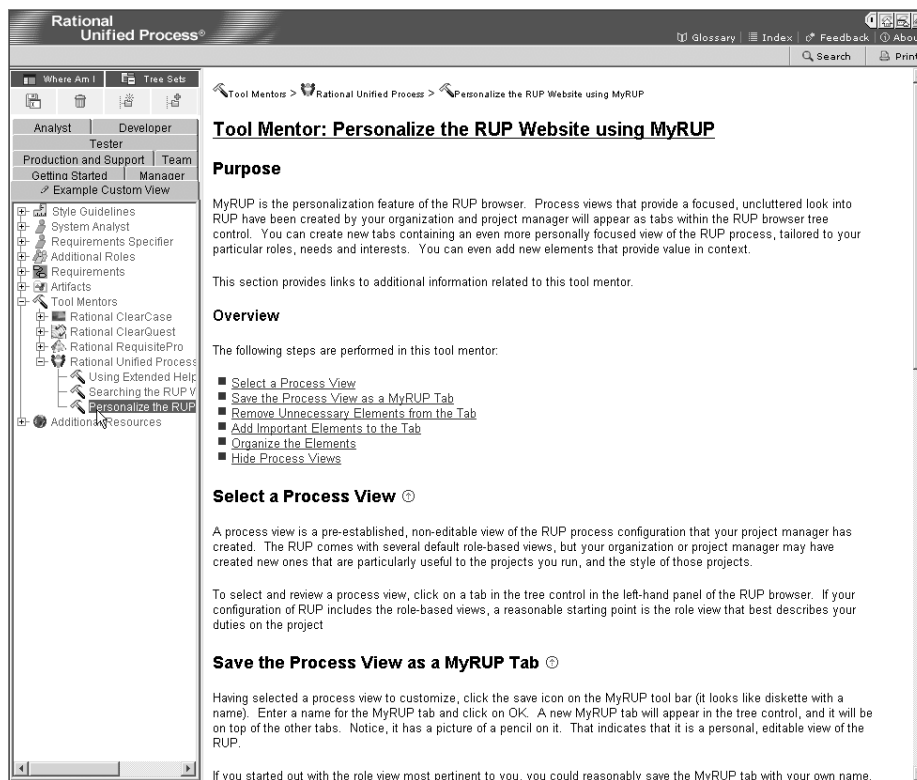


Fig. 3. Process views (tabs on the left) are trees of links to RUP content based on a particular perspective such as role. Process views can be created in RUP builder, or individually personalized (MyRUP).

Remember, even though each team member may use a personalized view, the entire project team is still using the same process that is, the same RUP Process Configuration. Each team member, however, can have his or her own personalized view into that common configuration. Each pre-made configuration in the RUP comes

with role-based Process Views that you can use as a starting point for your own Process Views.

3 Extending RUP

Some organizations may see the need to create RUP Process Configurations that are different from what can be produced with the RUP Plug-Ins available from Rational Software and its partners. These organizations can customize the RUP to accommodate the specific needs, characteristics, constraints, and history of its organization, culture, and domain. Maybe the organization needs to develop software according to a certain quality standard or management lifecycle. They can also customize the RUP to include examples or reusable assets from previous projects. Customizing the RUP should be done by creating RUP Plug-Ins; this enables you to create a different set of RUP Process Configurations. Plug-ins can be grouped into two major categories: Thin RUP Plug-Ins and Structural RUP Plug-Ins:

- **Thin RUP Plug-Ins** — changing content files for existing process elements. Plug-ins of this type allow customers to add, modify, or remove content files associated with existing process elements. This allows you to, for example, change the guidelines for how to produce a Vision Document based on experiences from previous projects, modify the templates for a certain artifact, add examples from your specific domain, or add reusable assets harvested from previous projects. These types of changes are done using a component of Rational Process Workbench (RPW) called RUP Organizer.
- **Structural RUP Plug-Ins** — changing process elements and their relationships. Plug-ins of this type allow process-mature customers, who have developed process content outside the RUP, to integrate that content into the RUP. You can add, modify, or remove process elements such as artifacts, activities, roles, disciplines, workflows, and tool mentors. This flexibility allows you to make major alterations to the RUP Library by, for example, adding your own process guidance on how to do package implementation or legacy system integration to the RUP. These types of changes are done using RPW. Modifications to the process elements and their relationships are done using an RPW component called RUP Modeler. Changes to the associated files for each model element are done using RUP Organizer.

3.1 Thin Plug-ins

RUP Organizer is an easy-to-use tool. You create content pages with your HTML editor of choice, create templates, and harvest examples and reusable assets. RUP Organizer allows you to associate these content files with the appropriate process elements in a process component by dragging and dropping them on a process

element (see Figure 4). You can then create a Thin RUP Plug-In containing all the changes. You can create a plug-in that consists of only a couple of files, or one that consists of a large number of files. In most cases, most of the time spent to create a plug-in involves editing the actual content files. Note that Thin RUP Plug-Ins will not change the structure of your process element — that is, it will not change what artifacts, activities, roles, workflows, tools, or tool mentors you have — nor will Thin RUP Plug-Ins change their relationships to each other, such as which artifacts are input or output to which activities.

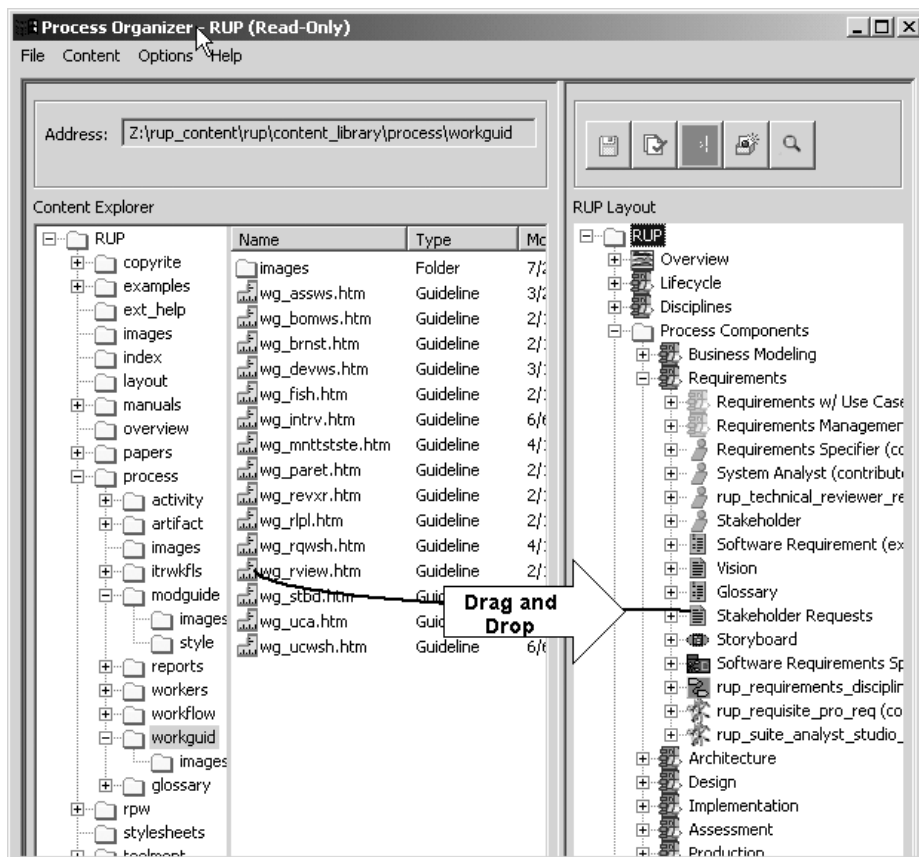


Fig. 4. RUP Organizer allows you to build thin RUP plug-ins. Content files, such as examples, guidelines, and templates, are associated to process elements by “drag and drop”.

3.2 Structural Plug-ins

RUP Modeler is a tool for the advanced process engineer. This tool allows process engineers to create or evolve an object-oriented model of the process, using UML. In

particular, they can introduce new process elements — roles, activities, artifacts, and the relationships between these process elements:

- Which artifacts are input or output of which activities.
- Which roles are responsible for which artifacts.
- Which roles perform which activities.

RUP Modeler allows you to define how these process elements should extend or override process elements in the RUP Base or other RUP Plug-Ins. It allows you to visualize the links you may have among these process elements.

Once you have created RUP Plug-Ins, you can import them into the RUP Builder, and create RUP Configurations containing the content you have developed in-house. RUP Builder uses the modeled information to automatically generate hyperlinks as it publishes a RUP Configuration.

Using the RUP Modeler requires a certain level of expertise. You need to be familiar with object-oriented modeling, Rational XDE, and the RUP product to build a Structural RUP Plug-In. It can be quite time-consuming to produce a structural plug-in, so you should undertake this effort judiciously. We recommend you use the RUP in a few projects before attempting to create Structural RUP Plug-Ins, to make sure that you understand what changes are necessary and what changes can be postponed or skipped completely.

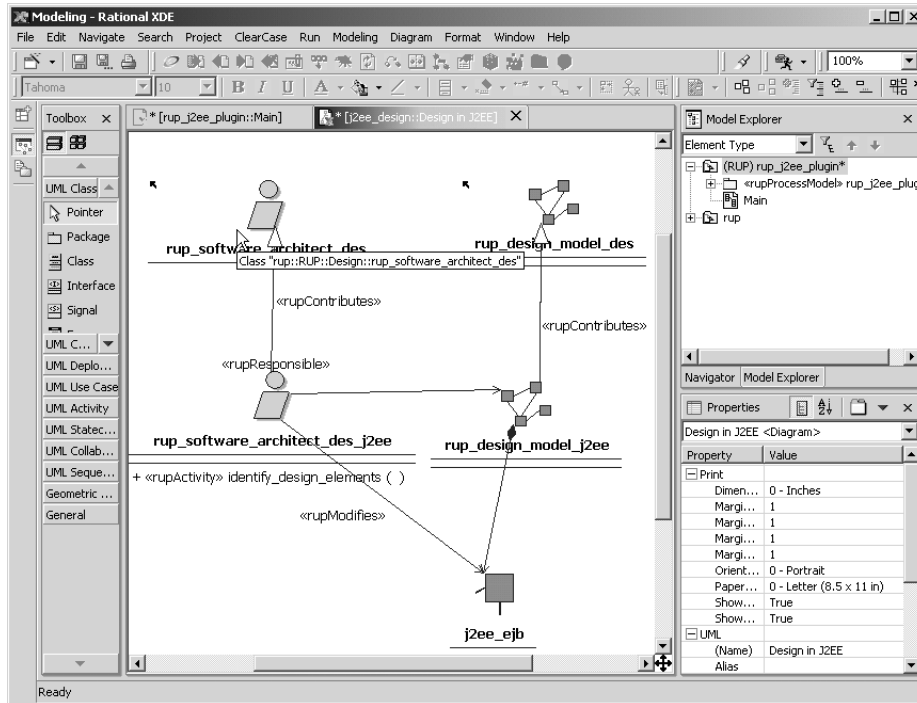


Fig. 5. RUP Modeler is a process modeling tool that allows the advanced process engineer to make major alterations to RUP.

4 Conclusions

The RUP with its supporting tools is a powerful Process Engineering Platform, which is both configurable and extensible.

The right software development process for a particular project depends on many factors, including size of project, formality, technology, applicable techniques, and development philosophy. For many projects, an out-of-the-box RUP configuration may be suitable. For others, you can configure a suitable process by selecting from the available plug-ins and components.

You can extend the RUP with your own plug-ins, and share them using the RUP exchange on the Rational Developer Network. In most cases, you should consider building Thin RUP Plug-Ins using RUP Organizer. Thin Plug-Ins allow you to add, modify, or delete guidelines, examples, templates, and reusable assets.

Process-mature customers with specific needs may also take on the bigger effort of producing Structural RUP Plug-Ins using RUP Modeler as well as RUP Organizer. Structural Plug-Ins allow you to make major alterations to the RUP.

The end result is one or more software development processes, suited to the current needs of your project and organization, and able to evolve to meet future needs.

Acknowledgements

Much of this article is excerpted with permission from [1]. Portions have been excerpted with permission from the RUP product itself [2].

References

1. Kroll, P., Kruchten, P.: The Rational Unified Process Made Easy, A Practitioner's Guide to the RUP. Addison Wesley (2003)
2. Rational Unified Process, version 2003. IBM Rational Software (2003)