

With the release of version 7.2 of the RUP, the included SOA materials have profited from important changes. How service-oriented is RUP actually?

SOA gains importance to the RUP

Looking at the world of service-oriented architectures (SOA), we see all IT giants fighting for market share for their infrastructure products, their development tools and their consultancy. IBM is an obvious actor in this arena. Almost every one of its products is positioned somewhere in Big Blue's SOA strategy. One of those products is the Rational Unified Process (RUP), the famous software development method. Nowadays, the RUP provides extensive and valuable information on how to shape an SOA when developing new systems. Unfortunately, the integration of this information with the rest of the method is limited.

From hype to hype

The RUP has grown up in the nineties, as a method for object-oriented software development. As soon as the OO hype was succeeded by component-based development (CBD), the component paradigm was eagerly adopted and appointed one of the six best practices of the method. In fact, the concept only had a limited influence on the details of the process. From the use cases, one still created an object-oriented analysis model, without components. The component diagram was deferred to chapter 8 of the Software Architecture Document, the so-called implementation view. Nowadays, the six best practices have been replaced by six key concepts and CBD has been put aside into the collection of 'supporting materials'. Now, the next train is passing by: service-oriented architectures. How did IBM react as far as the RUP is concerned?

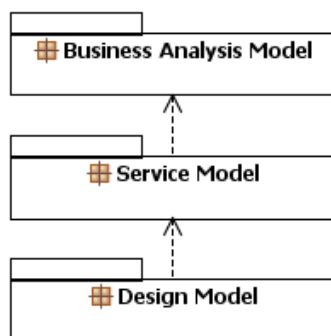


Figure 1. The Service Model: a bridge between business and technology.

Since 2006, IBM has an SOA plug-in for the RUP available. The word "plug-in" already indicates, that SOA principles are not regarded fundamental for the RUP. The plug-in contains extensive

information on the concept and a lot of SOA aspects are discussed in separate guidelines and white papers. Tasks and work products have been added to the method, giving SOA concepts a stronger role in the process than CBD. There is a remarkable work product called Service Model, that functions as a bridge between business and technology (see figure 1). But these tasks and products remain additions, while necessary adjustments to the core of the method don't come off. Existing tasks like Architectural Analysis and Use Case Design and work products like the Software Architecture Document have not been touched. This is still the case, even with the recent advent of version 2.4 of the SOA plug-in, released with version 7.2 of the Rational Method Composer, which is the product that covers the RUP.

The SOMA-method

Version 2.4 did show quite some progress nevertheless, since it is now based on the SOMA method developed elsewhere within IBM. SOMA's framework for SOA development consists of four activities (figure 2): Service Identification, Service Specification, Service Realization and Service Deployment, although the latter has not been covered by the RUP as yet. The circle of arrows in figure 2 implies, that these activities should be regarded as an iterative rather than a linear process.

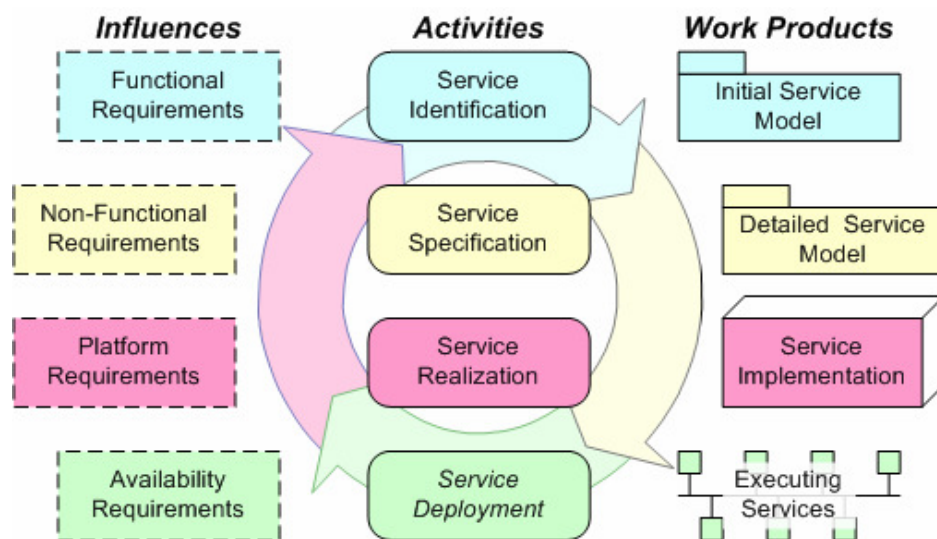


Figure 2. The SOMA framework

The activity **Service Identification** is focused on the selection of candidate services, based on business characteristics and existing IT assets.

During the **Service Specification**, the set of services to be developed is determined and accurately specified in the Service Model, including their mutual interaction and including the non-functional requirements imposed on each service.

Now, what is meant by **Service Realization**? Figure 2 suggests that this activity results in the software (or source code) that implements the services, but from the description it becomes clear, that it is mainly about designing the service components in the Design Model. In addition, a proof of concept is developed, which usually consists of working software, but this work product is only meant to validate the architecture.

Next, one would expect an activity called Service Implementation, to get to the source code eventually (generated as much as possible, of course), but such an activity doesn't exist. The authors have probably reasoned, that the general activity Implement Components, present in the RUP, already fulfills this need.

Service Deployment represents the transfer of the services to the production environment.

As already mentioned, the last activity is not covered by the RUP, but the others are described as a hierarchy of subactivities, with tasks on the bottom level, linked to roles and work products. The tasks describe in practical terms what needs to be done. The SOA plug-in adds fifteen tasks, eleven work products and one new role (the Security Architect) to the RUP.

Phases

Every RUP project follows a strictly linear scheme of four phases: Inception, Elaboration, Construction and Transition. Each of these phases, except for Inception, encompasses all the activities Service Identification, Service Specification and Service Realization, but the plug-in's documentation states, that Service Identification and Service Specification are primarily Elaboration-time activities and that Service Realization is primarily performed during Construction. Still it is odd, that these activities are even present in the Transition phase, during which significant software modifications are not allowed. Another remarkable fact is, that in the Inception phase, when the scope of the project is determined, no attention is paid to services, although, in a SOA context, the scope is often defined in terms of services. Moreover, Inception is the time when a proof of concept is normally developed, but the creation of the SOA proof of concept is now positioned in all phases but Inception, namely as a part of the Service Realization activity!

Guidance

Apart from activities, tasks and work products, the SOA plug-in offers thirty guidance documents. These documents provide explanation on certain tasks, work products or SOA in general. They are hyperlinked to the corresponding tasks and work products. There is much interesting guidance material, like the view on the concept "Service Portfolio", explaining how a portfolio could be structured and how a SOA is only really effective, if all the organization's services are

administered well. Another document called “Service Composition and Choreography” describes how to combine services to higher-level, more business-oriented services en how a design for that would look like, for example using BPEL. The guidance documents together form a valuable set for the SOA specialist, though not a tutorial for the beginner.

Work Products

The Service Model is the most important work product of the SOA plug-in. This model specifies the services with their operations, including input and output parameters, the messages exchanged, their protocols and also the non-functional requirements they have to adhere to. The SOA plug-in provides an explanation of the UML symbols to be used to benefit from IBM Rational Software Architect’s code generator, later on. The Service Model is thus created preferably using a tool like that. If there is a need for a paper document, it can be generated from the tool.

The Service Model consists of smaller work products. These come in eight flavors: Message, Service, Service Channel, Service Contract, Service Gateway, Service Partition, Service Provider en Service Specification. The RUP proceeds on the unpractical tradition of labeling all kinds of model elements as separate work products. By now, we have 129 different work products in total.

Another work product, the Service Component, is not a part of the Service Model, but of the Design Model. A component like this realizes one or more services. It is not a part of a service, as the word might suggest.

Finally, the SOA plug-in also defines the Goal-Service Model. This is a mapping of services on business goals, usually in the form of a table.

Conclusion

SOA is a spearhead for IBM, naturally, which is translated into a more and more improving SOA plug-in for the RUP. Has it now become a successful method for SOA projects? No. The web of information is too much of a labyrinth and insufficiently integrated into the whole. It is difficult to grasp the complete picture and once it has been grasped, it appears to be not perfectly consistent. The guidance however, deserves a compliment. This set of documents is a useful reference library for the software architect.

Hans Admiraal (freelance IT architect and RUP trainer)
admiraal@aol.nl

www.admiraalit.nl