
Skill Level: Intermediate

Bertrand Portier (bportier@ca.ibm.com)
IT Architect
IBM

Gregory Hodgkinson (greg.hodgkinson@7irene.com)
SOA Lead
7irene (IBM Tier 1 Business Partner)

22 May 2007

This tutorial introduces the concepts and tools involved with modeling service-oriented architecture (SOA) using IBM® Rational® Software Architect. In this series, you will learn how Rational Software Architect, along with IBM® WebSphere® Business Modeler, supports a model-driven development (MDD) approach to the specification of SOA. This tutorial walks you through these concepts using a fictitious online DVD rental case study.

Section 1. Before you start

Learn what to expect from this tutorial, and how to get the most out of it.

About this series

This series provides a detailed look at modeling service-oriented architectures (SOA) using the IBM® Rational® Software Architect tool. Although primarily targeted at software architects, this tutorial should also benefit other roles in the software development process. These roles could include business analysts (specifically for
Part 1), or software designers and developers who would use the architecture as input to perform their activities (architecture realization, design, and implementation). This series also covers a lot of core SOA concepts, which should be beneficial to a wide audience. See Resources for Part 2.

This series of tutorials focuses on teaching you how to do the following three things:

- **Architecture**: Describe what an architecture comprises, and where it fits into the overall software development process.
- **Services**: Architect a system using SOA. Services are central to this architecture.
- **Models**: Demonstrate how the Rational Software Architect tool supports a Model-Driven Development (MDD) approach to the specification of service-oriented architectures.

This series begins by describing software architecture and defining the place of services within software architecture. It then presents Rational Software Architect and its SOA- and architecture-related features.

Using a fictitious online DVD rental case study throughout, the series does the following:

- Describes the work products used as input to the service architecture activities, including the component business model, business process model, system use case model, and external systems part of the design model.
- Describes step-by-step how the service model representing the architecture is specified in Rational Software Architect, including service consumers, service specifications, service partitions, atomic and composite service providers, services, service collaborations, service interactions, and service channels.
- Explains how the service model is then used in the subsequent activities of the software development process, with specific attention to design and implementation.

**About this tutorial**

This tutorial, Part 1 of the series, introduces the video rental case study used throughout the series. It also introduces the tool, Rational Software Architect (Version 7 and later), and the features that you will use to model service
architectures. Finally, it describes two of the models used as input to the service modeling activity: the component business modeling (CBM) map, and the business process model.

Objectives

At the end of this tutorial, you should be able to:

- Describe the business reasons behind DVD2U's SOA architecture effort
- Describe what Rational Software Architect is
- Explain, at a high level, how Rational Software Architect can be used to model service-oriented architectures
- List what models are used as input to the service architecture activity
- Describe what a component business modeling (CBM) map is
- Describe the *Return Video* business process used for this project

Prerequisites

To get the best value out of this tutorial, it is recommended (but not necessary) to be familiar with:

- Business Process Modeling Notation (BPMN)
- IBM® WebSphere® Business Modeler
- Service-Oriented Architecture (SOA)
- Rational Software Architect

System requirements

- Rational Software Architect V7 (fix 002 recommended) or later
- WebSphere Business Modeler V6.0.2 or later
Section 2. The Video Rental case study

This tutorial series uses the example of a fictitious company named DVD2U throughout.

DVD2U is a DVD sales and rental company that was founded in 1998. They are in the business of providing their customers with movies that are difficult to find in the mainstream video stores, such as foreign or special interest ones. Since its inception, DVD2U has gone through two major acquisitions, one which allowed them to get a wider customer base from a traditional video tape company, and another one from which they got a lot of DVDs in foreign languages, but just a few new customers.

When you talked to the CEO, he explained that they had started a new online offering one year ago. Here is how it works: People sign up for the service, and pay a monthly subscription to become DVD2U members. Members build their wanted movie list online, typically a hundred titles, based on DVD2U’s available titles. DVD2U then sends DVDs to members via mail. After watching the movie, members return the DVD by mail using a postage-paid envelope. When DVD2U receives the DVD in their warehouse, they send another DVD from the member’s list.

Based on members’ subscription type (and price), they are allowed to have a limited number of DVDs out of the warehouse at a time (typical members choose the popular 3 DVD plan). DVD2U believes in online communities and strives to build one through its Internet site, where members can interact, rate movies, write reviews, and even find people who watched the same movies. People say that the DVD2U site is very effective for dating!

The CEO told you, however, that although there is a significant business opportunity in online rentals, there have been issues with the service. For example, people complained that they never seem to get the movies at the top of their list, or that it sometimes takes a couple of weeks to get a DVD after one has been mailed back, or that the Web site is down and they cannot update their movie list.

You discussed a number of other specific problems with the current IT solution:

• Their current solution is a point-to-point integrated set of disparate systems.
• The cobbled-together nature of the existing IT solution causes a number of breaks in continuity in the overall business process.
• There is a lot of duplication of functionality and data across the systems, combined with inconsistent usage to solve the same business problems.

• Often when changes are made to support new business processes variants, these changes break existing business process solutions.

• The lack of specification documentation, coupled with the lack of general understanding within IT as to how the systems support the business, makes it difficult to introduce changes.

Fortunately, the CIO is a very good friend of an IBM business partner who specializes in SOA, and a strong advocate of enterprise and solution architecture. After attending an IBM executive SOA summit with the enterprise architect, they decided to engage IBM for an SOA project. The project is contingent upon IBM delivering business value in the first three months, and this is where you come in!

Section 3. Service architecture positioning

Development process and methods

This section positions the core contents of this series using software development process and method terms. It defines what the activity is, where it belongs, who performs it, and what work products are used.

• **Work Product:** The core work product for this series is the Service model. It is represented as a Unified Modeling Language (UML) model in Rational Software Architect. The Service model describes the architecturally significant elements of an SOA, using the UML Profile for Software Services (UPSS). Part 1 of this series introduces Rational Software Architect and UPSS. Part 2 describes the service model in detail.

• **Discipline:** Figure 1 shows the IBM® Rational Unified Process® (RUP®) disciplines and phases for software development processes and methods. Business (modeling), Analysis, Design, and Implementation could also be interpreted as levels of abstraction. This tutorial focuses on the core of the Analysis and Design. (more on this can be found in the Learn part of the Resources section).
• **Phase:** The bulk of the architecture activities take place during the Elaboration phases of the project. During Inception, you (the architect) select the architecturally significant requirements. During Elaboration, you complete the "big-picture" architecture, along with a representative set of detailed architecture artifacts focusing on the high-risk use cases. You'll finish the remainder of the detailed architecture iteratively, all the way into the Construction phase.

• **Role:** The **software architect** performs the core tasks illustrated in this tutorial series. This role is responsible for making major technical decisions about the system being built in order to address all of the requirements. The result of these decisions is specified in the software architecture. Note, however, that the tasks described in this part of the tutorial series are primarily performed by the business architect and the business process analyst.

• **Activity:** The core activity detailed in this series is about modeling a service-oriented architecture (SOA, or **service architecture** for short). The RUP for SOMA (Service-Oriented Modeling and Architecture) plug-in, which includes SOA contents on top of classic RUP, calls this activity **Service Specification**. Please refer to the **Resources** section for links to the RUP for SOMA plug-in, which provides a way to combine RUP for SOA content with IBM’s Global Business Services (GBS) SOMA method.

• **Tool:** The tool used by the software architect to model is **Rational Software Architect**. Rational Software Architect V7 and later includes more SOA capabilities than earlier versions. These capabilities will be described in this part of the tutorial. This part also uses another tool called **WebSphere Business Modeler** Advanced Edition (V6.0.2 and later), for business analysts to model business processes. Please refer to the **Resources** section for links to trial downloads for these products.

**Figure 1: RUP Disciplines**
Please refer to the Architecture and Services developerWorks article series (under Learn in the Resources section) for more details regarding SOA and IBM products.

SOA Solution Stack

As defined in the SOA terminology article series (see the Resources section for a link), the SOA solution stack, shown in Figure 2, is an SOA reference model depicting the conceptual view of an SOA solution. Sometimes referred to as the SOA layered architecture, this model introduces layers (concepts such as business process, service, or service component) and the relationships between them. It is independent of the technology used for implementation.

Figure 2: The SOA Solution Stack
The 5 functional layers are as follows (bottom to top):

- **Operational systems** represent existing IT assets (later called *external systems*), and shows that previous IT investments are valuable and should be leveraged in an SOA.

- **Service components** realize services, possibly by using one or more applications in the operational systems layer. As you can see in Figure 2, consumers and business processes do not have direct access to components: they access services. Existing components can be internally reused, or leveraged in an SOA if appropriate.

- **Services** represent the services that have been deployed to the environment. These services are governed, discoverable entities. They are grouped on service providers and consumed by service consumers.

- **Business Process** represents the operational artifacts that implement business processes as service choreographies.

- **Consumers** represent the channels that are used to access business processes, services, and applications.

The 4 non-functional layers are:

- **Integration**: Provides capability to mediate, route, and transport service requests to the correct service provider.

- **Quality of Service**: Provides capability to address the non-functional requirements of an SOA (e.g., reliability, availability).
• **Information Architecture**: Provides capability to support data, metadata, and business intelligence.

• **Governance**: Provides capability to support business operational life-cycle management in SOA.

More detail is needed to explain what the SOA Solution Stack is. Please refer to the "Design an SOA Solution using a Reference Architecture" article, listed under Resources section, for detail on this.

This series focus on the central layer, the services layer. In the later tutorials in this series, you are going to spend time specifying the model elements of the service model work product. Part 1 (this tutorial) talks about the business process layer. This is often referred to as the top-down approach. Detailed design elements belong to the service component layer. Leveraging existing (non-SOA) systems from the operational systems layer is called the bottom-up approach. We recommend a combination of top-down and bottom-up called meet-in-the-middle: top-down business process modeling will define the functional scope of the solution; bottom-up existing asset analysis will discover the functionality provided by external systems (non-SOA) along with any constraints they impose. Note that existing service models may describe existing SOA functionality available.

The models used as input for the service modeling activity

The following models are used to perform the service modeling activity:

• **Component Business Model**: A strategic business-level componentization of the enterprise

• **Business Process Model**: The flow of tasks, the business items that are passed between tasks, and the roles that perform these tasks; also describes where automation is required

• **Domain Model**: The consolidated view of business information

• **System Use Case Model**: The interactions between actors (human and system), and the IT system of concern

• **External Systems Model**: The non-SOA systems that you can leverage

• **Service Model (analysis-level)**: Groups of identified and validated conceptual services

In this tutorial (the first in the series) we describe the component business model and the business process model.
Section 4. Rational Software Architect V7 and later

Rational Software Architect is an advanced MDD tool, which allows you to model software at different levels of abstraction; use transformations to move models from one level to the next; and generate, develop, and test code. This section describes the Rational Software Architect features that are relevant to SOA and architecture. You can skip this section if you are already familiar with Rational Software Architect.

Rational Software Architect is based on the open source Eclipse platform, and Version 7 is based on Eclipse Version 3.2. It supports team development using the following:

- Asset-Based Development (ABD)
- Pattern-Based Engineering (PBE)
- Unified Modeling Language (UML)
- Java™ Standard Edition
- Java™ Enterprise Edition
- The Reusable Asset Specification (RAS)
- Web Services Interoperability (WS-I) profiles

Rational Software Architect combines IBM® Rational® Software Modeler and IBM® Rational® Application Developer in an integrated environment. It also provides integration features with other products such as WebSphere Business Modeler, IBM® Rational® Clear Case®, CVS (concurrent versions system), IBM® Rational® ClearQuest®, and IBM® Rational® RequisitePro®.

If you are new to Rational Software Architect, we recommend that you take a tour from its Welcome page (as shown in Figure 3), which should be displayed when you launch Rational Software Architect and use a new workspace. You can also access it through Help > Welcome.

Figure 3: The Rational Software Architect Welcome page
The Welcome page allows you to enable and disable Rational Software Architect capabilities based on roles (bottom right), and provides pointers to Rational Software Architect-focused information such as:

- **Overview:** A description of the features supported by Rational Software Architect
- **Tutorials:** Tutorials that teach you how to use key features in the area of UML modeling or application development
- **Samples:** Sample projects containing model or code, illustrating key features of the tool across the same areas as covered by the tutorials
- **What’s new:** A description of major advances to features in this release
- **First steps:** First steps provide step-by-step guidance on performing key tasks supported by Rational Software Architect
- **Web resources:** Links to relevant online resources, mainly on the IBM® developerWorks® Web site or ibm.com
- **Migrate:** Information on how to import projects from other versions (for example, Rational Software Architect V6) or tools (for example, IBM® Rational Rose®)
If you are a software architect and fairly new to Rational Software Architect, we specifically recommend that you look at the following under Overview > Modeling Basics > Modeling life cycle support > Integrations for the development life cycle.

Perform the following steps:

1. If you have not already done so, install Rational Software Architect (Rational Software Architect) V7 (see the Resources section for a download link).


3. In the Workspace Launcher dialog, specify a directory for your workspace (for example, C:\rsa-workspace) and click OK.

4. Rational Software Architect should launch, and you should see the Welcome screen (Figure 3).

5. Explore the documentation listed in this section.

The modeling perspective

In User Interface (UI) terms, an Eclipse perspective is a set of views grouped together to support specific roles or activities. Rational Software Architect includes built-in perspectives (for instance, Modeling, Plug-in Development, or Java), and you can also create your own. In this tutorial, you will spend most of your time under the Modeling perspective, as show in Figure 4.

Figure 4: The Rational Software Architect Modeling Perspective
The Modeling perspective includes 4 main views:

- The **Project Explorer** view, where you can see the model elements and diagrams grouped under projects, models, and packages
- The **Diagram Editor** view, where you can view or modify diagrams, and create, delete, or update model elements
- The **Outline** view, which allows you to see what part of a large diagram is currently being displayed in the diagram view
- The **Properties** view, which provides editable detailed information on the selected model element

Perform the following steps:

1. From the Welcome screen, click **Go to the workbench** as shown in Figure 5.
2. By default, you should be under the Resource perspective. Switch to the Modeling perspective by selecting Window > Open Perspective > Modeling.

This is the last thing that you do with Rational Software Architect in this part of the tutorial series. You can close it for now.

**Figure 5: Go to the workbench**

*New SOA features in V7*

There are two SOA features that are now part of the V7 Rational Software Architect product. Note that in V6, these used to be assets available on developerWorks. As SOA becomes more mainstream, these features have been integrated in the product and are fully supported, and you’ll leverage them in this tutorial series:

- **The UML 2 Profile for Software Services (UPSS):** This profile defines the stereotypes that you use to model your service architecture (for example, «serviceSpecification», «serviceConsumer», «serviceProvider», and «service»). Further parts in this tutorial series will introduce each stereotype as you use it in the service model. If you want to start learning about it now, refer to one of the following in the Resources section for more details:
  - The developerWorks article on the UML 2 Profile for Software Services
  - The Building SOA Solutions with the Rational SDP Redbook, chapter 9, service model section

- **UML to WSDL transformation:** Core to MDD is the ability to use transformations to generate a target model or code, based on a source model or code. Rational Software Architect provides supported transformations out-of-the-box, along with a framework to build your own
custom transformations. The UML to Web Services Definition Language (WSDL) allows you to generate WSDL and XLM Schema, based on a UML model. In our case we use the service model as input. As the service model describes the architecturally significant parts of the system, it will be used to generate the architecturally significant parts of the implementation. Further detail is added by the design model, which can be used to generate further implementation detail (the internal implementation of the architecturally significant parts).

At the time of this writing, we used V7 with fix 001 applied. We recommend you use the latest fix available at the time you go through the tutorial.

Section 5. Component Business Model

As defined in the SOA terminology article series (see the Resources section for a link), the IBM® Component Business Model™ is a strategic method that allows businesses to focus on core competencies -- the parts of the business that differentiate it from its competitors, see how resources are consumed, and better align business and IT goals.

Service orientation achieves the necessary integration of these business components’ interaction, as well as their flexibility (such as outsourcing a component): business components have a unique business purpose, and collaborate through a set of business services that they provide to (or consume from) other business components.

Note that creating a full-blown Component Business Model is normally outside of the scope of a software development project: it is normally created by a business strategy and change effort. Where this input is not available, it is sometimes worthwhile creating a basic Component Business Model map to provide context for the business modeling efforts of the software development project. The advantage in doing so is that your CBM map will provide a useful context for the business processes that you define. It also acts as a useful check-point to check the coverage of your modeled processes: have you considered the full-breadth of the business that is included in your processes?

Figure 6 shows the Component Business Model Map for the DVD2U enterprise.

Figure 6: DVD2U Component Business Model Map
Business components (functional areas) belong to one of three accountability levels (the rows of the map):

- Direct (Strategy)
- Control (Management)
- Execute

DVD2U has four business competencies (the columns of the map):

- **Warehousing**: Stocks, shipments, and returns
- **Sales**: Mail sales, store sales, and promotions
- **Rental**: Store, online, or mail rentals
- **Marketing**: Pricing, marketing campaign, and advertising

By conducting a Component Business Model workshop, you have identified that the **Online Rentals** business component (under **Rental, Execute**) is the key business component of focus. Component Business Model calls it a *hot component*, and there
is a star icon attached to it on Figure 6. **Online Rentals** is thus the business functional area for which the Component Business Model effort will look at people, processes, systems and, most importantly, services.

---

### Section 6. Business Process Model

Figure 7 and Figure 8 show the *Return Video* business process, which belongs to the **Online Rentals** business competency. This is the business process that sets the scope of the SOA project for which you are modeling the architecture in this series.

**Figure 7: Return Video Business Process (1 of 2)**

The process flow is as follows, from Figure 7:

1. The DVD2U member, using a prepaid envelope, mails the video back to the DVD2U warehouse.

2. Optionally, the member logs into his DVD2U account from a Web browser, and updates his movie list by noting which videos he has mailed.
3. When this happens, the system retrieves the member’s standing.

4. After a day or two, the receiving clerk working at the DVD2U warehouse receives the video.

5. The receiving clerk then inspects the video.

**Figure 8: Return Video Business Process (2 of 2)**

The process flow from Figure 8:

1. If the member reports a video return and the member is in good standing, then the system updates the member’s profile to say that the next video on his list is due.
2. After inspecting the video, the receiving clerk records the receipt of the video in the system.

3. The system adds the video copy back to the overall warehouse stock.

Please note that the business process modeling effort identified and codified:

- **Roles**: Member (top swim lane), and Receiving Clerk (middle swim lane)
- **Automations** (as classifiers): Human Only (orange), Human-System (blue), System-Human (light blue, not shown), and System Only (grey)
- **IT Systems** (as classifiers): Membership Management, Stock Management, and Member Management (all part of Customer Relationship Management). Note that a common alternative to using classifiers to model IT systems is to instead model them as individual resources in WebSphere Business Modeler.

Note also that business process modeling involves other activities (such as process simulation) which you do not talk about in this series.

Perform the following steps:

1. If you have not already done so, install WebSphere Business Modeler Advanced Edition (v6.0.2 and later). See the Resources section for the download link.

2. Launch WebSphere Business Modeler.

3. In the pop-up dialog, click **Browse**, and then create a folder for your WebSphere Business Modeler workspace (for example, C:\temp\wbm-workspace). Click **OK**.

4. WebSphere Business Modeler should launch and automatically launch the Quick Start wizard. Name the new project **DVD2U Online Rentals** and click **Finish**.

You will now import the provided business process model.

1. Select **File > Import**.

2. In the Import wizard, select WebSphere Business Modeler Import and
click **Next**.

3. Select WebSphere Business Modeler Project (.mar, .zip) and click **Next**.

4. Click **Browse** and point to the directory where you saved the compressed file (see the **Download** section of **Resources**). Under **Files**, select DVD-Rental.mar. Select DVD2U Online Rentals as the project. Select **Overwrite existing elements** and click **Finish**.

5. Explore the model and take a look at the diagrams mentioned in this section.

Many things from this model are inputs to the service modeling activity, including the flow of tasks, the business items communicated between these tasks, descriptions of automation requirements for tasks involving IT systems, and the performing role for tasks performed by humans. As part of the business modeling work that you performed in this tutorial, you would typically also produce a domain model. The next tutorial in the series will describe this in more detail, as well as covering the next logical step in this top-down approach, which is the modeling of requirements in the system use case model.

---

**Section 7. Wrapping up**

**Conclusion**

This tutorial has set the stage for this series on modeling Service-Oriented Architecture with Rational Software Architect. This part of the series used a top-down approach and described the DVD2U company, its business, and its business issues. Then, it positioned software architecture and SOA relative to development processes, and described the focus of the series using the SOA Solution Stack. It next described the Rational Software Architect tool and its new SOA features. Then, after listing the models that are used as input for the service architecture activity, it described the two business models for this project: the DVD2U Component Business Modeling map, and the Return Video business process. This is just the foundation, and Part 2 will describe in detail the other input models, and start the initial structure of your software architecture. So stay tuned to developerWorks!
## Downloads

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
<th>Download method</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Business Modeler v6.0.2 project</td>
<td>DVD-Rental.zip</td>
<td>111KB</td>
<td>HTTP</td>
</tr>
</tbody>
</table>

Information about download methods
Resources

Learn

• Learn more about Model service-oriented architecture with IBM Rational Software Architect, Part 2: Modeling the business domain.
• Read the SOA terminology series to learn more about SOA.
• Visit the IBM Service-Oriented Architecture (SOA) Solutions area on IBM.com.
• Get training on service-oriented architecture with these IBM courses.
• Discover how to build SOA solutions with the Rational SDP Redbook to learn more about SOA.
• A four part tutorial series: Design SOA services with Rational Software Architect
• Find more information about IBM Component Business Modeling (CBM)
• Design an SOA Solution using a Reference Architecture.
• All you want to know about the Unified Modeling Language (UML) specification.
• Learn more about The UML 2.0 Profile for Software Services.
• In the Architecture area on developerWorks, get the resources you need to advance your skills in the architecture arena.
• In the Pattern Solutions area on developerWorks, get the resources you need to advance your skills in patterns-based development.
• Browse the technology bookstore for books on these and other technical topics.

Get products and technologies

• Download IBM product evaluation versions and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.
• Get the RUP SOMA plug-in.
• Download a trial version of WebSphere Business Modeler Advanced Edition (V6.0.2 and later).
• Download a trial version of Rational Software Architect (V7 and later).

Discuss

• Participate in the discussion forum for this content.
• Check out developerWorks blogs and get involved in the developerWorks community.

About the authors

Bertrand Portier
Bertrand Portier works for IBM SWAG SOA Advanced Technologies (formerly EIS). He is heavily involved in Service-Oriented Architecture, model-driven, and asset-based development. A regular speaker at conferences and the author of several technical articles, he has also co-authored an IBM Redbook about SOA solutions.

Gregory Hodgkinson
Gregory Hodgkinson is founder, director, and the SOA lead at 7irene, an IBM Tier 1 Business Partner in the United Kingdom (www.7irene.com). He has 10 years of experience in software architecture, initially specializing in the field of component-based development (CBD), then moving seamlessly into service-oriented architecture (SOA). His extended area of expertise is the software development process, and he assists 7irene and IBM customers in adopting RUP framework-based agile development processes and SOA methods. He is still very much a practitioner, and has been responsible for service architectures for a number of FTSE 100 companies. He presents on agile SOA process and methods at both IBM (Rational and WebSphere) and other events. He has also co-authored a Redbook on SOA solutions.

Trademarks

This is the first trademark attribution statement.
This is the second trademark attribution statement.