



IBM Rational Software Architect, Version 7.0

Highlights

- ***Offers easy adoption and use***
- ***Enables simplified architectural modeling and specification***
- ***Supports Java, EJB, C++, DDL, Web, WSDL and XSD modeling with markerless graphical editing and visualization of structure and behavior in diagrams using UML-like notation***
- ***Offers a complement of software construction tools for a variety of implementation technologies***
- ***Leverages the open, extensible Eclipse 3.2 platform***
- ***Integrates with your software lifecycle and development team's processes***

IBM Rational® Software Architect, Version 7.0 software offers a powerful, integrated design and construction environment that helps software architects understand, design, manage and evolve enterprise solutions and services across the team, across the world and across different areas of technical expertise. Part of a flexible family of business driven development products that are built on the open Eclipse 3.2 integration platform, Rational Software Architect offers extended support for the Object Management Group's (OMG's) industry standard Unified Modeling Language (UML). And the software's many powerful visual modeling and editing features are designed to improve productivity, enhance architectural control and ease the design-to-code experience for Java™ and Java 2, Enterprise Edition (Java EE), Web services, service-oriented architecture (SOA) and C/C++ applications.

Model applications more productively than ever with UML 2

Rational Software Architect, based on UML, Version 2.1, includes new ease-of-adoption and ease-of-use features that raise the bar for user productivity in modeling workflows. Its support for a variety of diagram types, including freeform diagram support, aids in design, discovery and documentation activities.

The UML 2 modeling capabilities in Rational Software Architect include new object diagram support and extended capabilities for modeling activities, composite structures, components, interactions (sequence and communication diagrams) and state machines—in addition to providing use-case, class and deployment diagramming capabilities.

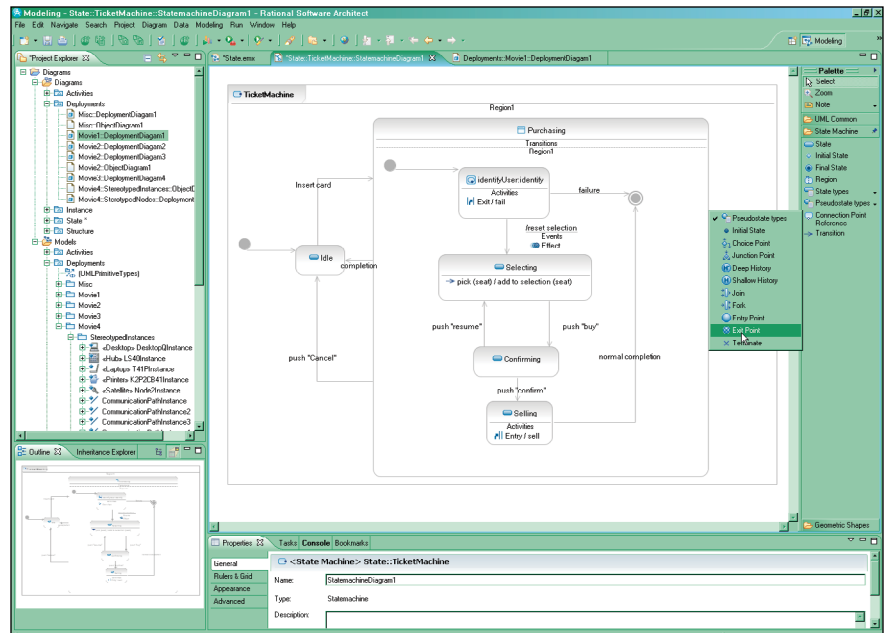
Exploit the latest in modeling-language standards

Rational Software Architect supports the OMG's Model Driven Architecture (MDA) initiative by allowing the user to

define multiple levels of models coupled with user-defined transformations between models and code, resulting in a clearer separation of concerns across the lifecycle. Rational Software Architect is also unique in its ability to support modeling in other domains such as Java, Enterprise JavaBeans (EJB), C++, Data Definition Language (DDL), Web Services Description Language (WSDL) and XML Schema Definition (XSD)—letting you create diagrams that combine elements from the UML and from these other semantic domains using consistent, UML-like notations.

Modeling flexibility for teams

Rational Software Architect offers flexibility in managing UML models. Capabilities for model fusing, i.e., “combine these models,” and model decomposition, i.e., “make this package into its own model,” let you evolve the logical organization of model content into model files that adapt to changing needs. Model files can be further deconstructed into smaller, transparent “subunit” files that are version controllable at a modular level to support team development approaches based on exclusive check-out policies. And if you prefer to follow true parallel development policies with nonexclusive checkouts, Rational Software Architect features fast and efficient model-merging capabilities to compare, merge and reconcile parallel changes as they are contributed. Extended model search capabilities round out the productivity features, enabling specific model information to be identified.

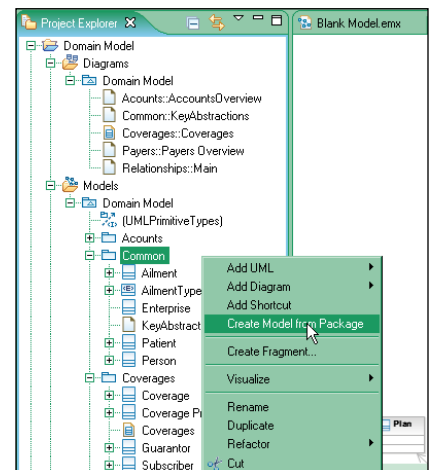


The Rational Software Architect product provides rich support for UML modeling based on UML, Version 2.1.

Tap into the power of model-based automations

Modeling alone adds value and helps reduce project risk, but even greater benefits are realized when models are used to automate the creation of other development artifacts, including other models, code and more. Use the design patterns included with Rational Software Architect to help you more quickly build the content of your UML analysis and design models. Or, create your own UML-based patterns to extend this benefit even further.

Rational Software Architect supports model-to-code and code-to-model transformations. Forward transformations go from UML to Java, C++, EJB, WSDL, XSD and CORBA Interface Description Language (IDL). Additional transformations go from UML to structured query language (SQL)–based logical data models as supported by IBM Rational Data



The Rational Software Architect product affords highly flexible model management. Logical models can be broken apart and recombined, as well as partitioned into fragments (multiple physical files) that can be independently version controlled.

Architect software, and from UML to SOA constructs, via the Business Process to Service Model transformation. This transformation takes a UML business analysis model that is authored in IBM WebSphere® Business Modeler software and opened in Rational Software Architect into a software services UML model. You can also use a software service UML model as the source for the UML-to-SOA transformation, which transforms the UML model into software services artifacts. The implementation includes a UML-to-Survivable Control Data Link (SCDL) extension that supports generating default service components and service components implemented with Business Process Execution Language (BPEL), along with the supporting WSDL and XSD artifacts. The output of the UML-to-SOA transformation can then be imported into IBM WebSphere Integration Developer software to perform more development, testing and deployment of the generated artifacts.

Reverse transformations go from Java and C++ to UML. And IBM Rational Modeling Extension for Microsoft® .NET software extends Rational Software Architect with transformations from UML to C#, and from C# to UML. Created from code, temporary models help you determine whether an implementation conforms to a specified architectural model. And Rational Software Architect includes tools for developing custom transformations that might target any type of implementation outputs and transformations between UML models at different levels of abstraction.

Bridge the gap between models and code in Java, Web services, SOA, portal and C++ applications

In a business driven development world, software matters. So you have to ensure that what is in the design is what ends up in the code—even as requirements change. Rational Software Architect supports modeling for Java Platform, Standard Edition 5 (Java SE 5) common constructs and enhanced graphical editing capabilities for Java, EJB and data using UML-like notations. The Rational software also includes application structural review and control features that directly address concerns for Java and C++ code. Significant tooling for Java EE security makes it easier to create, assign, integrate and test the application's security requirements.

The Rational Software Architect Web diagram editor provides a range of capabilities for designing complete Web applications in one easy-to-use visual editor. To enable your development team to save time and boost productivity, the Web modeling editor helps you create data- or service-driven applications within the Web diagram tool. Portlet design wizards support complex data types, as well as JavaServer Faces (JSF) and Struts portlets. The integrated IBM WebSphere Portal Test environment enables you to test business process portals with other resources, such as content management systems and embedded messaging.

SOA capabilities enable more robust and scalable development. Template models and UML profiles help jump-start the process of SOA development. A range of powerful Web service development capabilities such as easy-to-use wizards, WSDL and XML Schema editing, complex schema support, remote WSDL validation, and WSDL and XSD modeling all help to improve productivity. SOA best practices, included in the process advisor feature, complete the SOA design solution.

As for implementation technologies, Rational Software Architect includes features found in the IBM Rational Application Developer for WebSphere Software product for creating Java EE, Web and Web services applications. With support for the design and development of applications optimized for IBM middleware and IBM DB2® software, Rational Software Architect also includes IBM WebSphere Application Server, Version 6.1 and IBM WebSphere Portal Server, Version 6.0 software for testing. Support for the Sun Microsystems Java Runtime Environment enables optimization of applications for Apache Tomcat and middleware environments from other vendors.

Take advantage of the Eclipse open and extensible platform

Because Rational Software Architect is built on top of the open Eclipse 3.2 platform, you can more easily extend the software's features to meet



specific project requirements through an ecosystem of third-party plug-ins. Eclipse is written in the Java language, which means you can outfit your team across both Microsoft Windows® and Linux® desktops.

Rational Software Architect and the Eclipse foundation are modular, so installation is highly configurable. Install both Eclipse and Rational Software Architect, or install selected Rational Software Architect features into an existing Eclipse shell that, for example, supports an established development project or environment. This modularity also makes it possible to unify the data, application and software development environments on Eclipse.

Integrate with other facets of the lifecycle

Rational Software Architect products help you integrate with the full software development lifecycle. Requirements stored and managed in IBM Rational RequisitePro® software can be accessed, associated to corresponding modeling elements and synchronized with user-selectable rules. Users can generate reports highlighting traceability from requirements to design. Modeling files can be managed by the included IBM Rational ClearCase® LT software, IBM's robust

software configuration management product. Alternatively, Rational Software Architect integrates with Concurrent Versions System (CVS) for users who are already committed to that tool. And the integration with the IBM Rational Unified Process® (IBM RUP®) solution gives teams the ability to work through all of this with common, online and integrated process guidance.

Rational Software Architect integrates with these and other aspects of the IBM Rational Team Unifying Platform™ solution, providing requirements management, traceability, source code control, automated documentation and other team management functions throughout the lifecycle. This helps reduce the risk associated with software development and helps make application development more predictable.

For more information

To learn more about IBM Rational Software Architect software, or to download a trial version, visit:

ibm.com/rational/adc

© Copyright IBM Corporation 2007

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
09-07
All Rights Reserved

ClearCase, DB2, IBM, the IBM logo, Rational, Rational Unified Process, RequisitePro, RUP, Team Unifying Platform and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product and service names may be trademarks or registered trademarks or service marks of others.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.