A Technical Discussion of Rational Rapid Developer
Technical Overview

July, 2003

IBM® Rational® Rapid Developer Technical Overview

James R. Farrell
jrfarrel@us.ibm.com

Executive Summary

The Internet has brought a major revolution in business and technology. It provides the ubiquitous business dial tone that has the potential to automate and transform every aspect of business. Businesses are connected on a real-time basis with customers, vendors, partners and employees. Every aspect of your business is likely to change including business models, types of products and services offered, pricing models, and business relationships.

A parallel change is occurring on the technology side. Your applications use *n*-tier deployment architectures to achieve 24 x 7 availability, and need to be highly scalable and secure. The technology to power your applications is likely to keep changing rapidly. This has an obvious impact on your application development environment. As an application developer you need to anticipate a changing environment in which business and technology become more complex.

The vital business processes of today's organizations increasingly depend on software. Organizations must be able to adapt their software assets rapidly to meet challenges and opportunities as they come along. The ability to generate reliable, scalable business applications has a direct correlation with the ability to respond to both internal and external customer demands. For many organizations, business applications are the most critical software assets, because they reflect the core competency of an organization and are essential for maintaining a competitive edge.

Today's smart organizations are connecting business applications to their customers, vendors, and other partners; they're connecting the many departments, locations, and people within the enterprise; and they're making the growing enterprise responsive to the needs of the business. These applications need to be agile, so they can adapt to changes in the business and advances in technology. They need to extend existing mainframe systems and enterprise applications to the Web and to the outside world. They need applications that are high performance, scalable, manageable, and highly secure. Building such applications is a tall order for many in-house development teams, which typically have essential domain knowledge about how the business works, but may also have legacy technology skills. IBM® Rational® Rapid Developer leverages the skill sets of your existing developers, and allows developers to quickly utilize the latest technologies. Rational Rapid Developer enables you to:

- Quickly extend existing mainframe systems and enterprise applications to the Web
- Develop high performance, scalable, manageable, and secure applications
- Maximize developer productivity

About IBM Rational

IBM Rational, formerly an independent company and now one of the IBM software brands, offers a comprehensive software development solution. The IBM Rational platform combines software engineering best practices, market-leading tools, and expert professional services, all of which drive rapid and continuous improvement in software development capability for on demand businesses.

In addition, IBM Rational offers more than 20 years experience in promoting and delivering integrated and open software systems, both of which are key characteristics of the on demand operating environment:

Integrated — IBM Rational has contributed considerable thought leadership and expertise in the areas of Service-Oriented Architecture (SOA), enterprise and software architecture, and heterogeneous platform support.

Open — IBM Rational has a long history in developing and supporting the goals of open computing. This includes development of the Unified Modeling Language (UML), now a standard for modeling applications, database design, and business processes. IBM Rational has promoted and participated in the development of a wide variety of open computing standards. It offers support for major programming languages and operating platforms, and it provides an extensive set of application programming interfaces for third-party tools interoperation.

Thousands of companies around the world have realized the benefits of the approach advocated by IBM Rational. Their processes are results-oriented, the artifacts they produce are well-designed and reusable, and they are working at higher levels of capability now required by the on demand era.

Contents

Introduction 4

Rational Rapid Developer Is Architected RAD 4

Empower a Broad Class of Developers to Succeed on J2EE and *N*-tier Projects...... 5

Simplify Integration of Legacy and Enterprise Systems 5

Realize Agile, Architected, Scalable *N*-tier Applications...................................... 6

Gain Optimum Performance from IT and Staff Assets to Accelerate Application Delivery.. 6

Types of Rational Rapid Developer Applications 7

Application Development Process 8

Functional Requirements........................ 8

Leverage Your Assets............................. 9

Highlighted Product Capabilities 12

Business Modeling 12

User Interface Modeling....................... 14

Intelligent Controls and the ObjectSpace 15

Section 508 Web Accessibility 16

Multi-lingual, Localized Applications.. 16

Database Architect................................ 17

Security Architect 19

Messaging 20

Legacy Integration................................ 22

Components 22

Web Services 22

Team Development............................... 23

Automated Deployment 23

Partition Architect................................. 24

Logic Architect 24

Conclusion.. 25

Introduction

IBM® Rational® Rapid Developer, enables teams to add sophisticated J2EE development capabilities to their toolbox without extensive retraining. This whitepaper focuses on how this new product can help teams leverage "*architected rapid application development*" (ARAD) in the new on-demand environment.

With Rational Rapid Developer, IT shops can deliver on-demand applications in less time and at a lower cost. Staff developers with valuable business domain skills but limited J2EE skills (or outdated technology skills) can quickly train to become skilled *n*-tier developers using the latest technologies. They can deploy applications with high scalability and performance, and with the flexibility to mix and match technologies and adapt to new ones. The repeatable ARAD process helps developers to deliver applications with predictability and a high rate of success. Rational Rapid Developer, with its plug-in for the IBM Rational Unified Process, or RUP®, enables an IT shop to transform itself into a powerful strategic enabler for the on demand enterprise.

Rational Rapid Developer Is Architected RAD

- Offers a single, integrated application development environment
- Combines model-driven development, RAD techniques, and automated code construction
- Delivers agile, well-architected, executable *n*-tier business applications rapidly
- Insulates developers from most of the traditional complexities of *n*-tier development and underlying technology platforms
- Enables organizations struggling with Web or business application backlog to leverage a broad class of developers for *n*-tier, legacy integration, business modernization, and J2EE development projects.

Simply described, ARAD is rapid application development (RAD) that has grown up to meet the needs of development teams charged with creating enterprise-class applications for on-demand business. Picture a team of eleven people working on an enterprise application – a project leader, one analyst, eight developers, and one technology architect. Whereas RAD focused only on the developers, ARAD provides a unified environment for all of these team members. The unified environment comprises all elements and capabilities essential to software development success:

ARAD enables organizations to leverage the valuable skills of legacy developers and enable these developers to quickly develop *n*-tier applications

- A unified process
- Business modeling via the Unified Modeling Language (UML)
- Rapid application development
- Integration of legacy and enterprise systems
- Scalable n-tier development.

Early RAD (and non-RAD) approaches lacked a focus on the business processes that drive software requirements. Without a clear view of how such processes mapped to business needs, development teams could not accurately target competitive opportunities or improve efficiencies. With ARAD, however, analysts on the project team can use UML to analyze business requirements, define business use cases, define the information model for the application – consisting of business objects and their storage in databases – and define the business processes and business rules that will drive the application.

Empower a Broad Class of Developers to Succeed on J2EE and *N*-tier Projects

In today's organizations, most project teams consist of application developers with experience in a variety of coding environments, such as Visual Basic, PowerBuilder and Oracle Forms, Java, C++, and COBOL. Their experience also covers a diverse array of deployment scenarios, from *n*-tiered architectures, to client/server, to legacy/mainframe environments. Although they have diverse technology skills, these developers all understand the need to write code at an increasingly rapid pace, and to support requirements that change ever more rapidly. With ARAD, organizations can leverage the valuable technical and domain skills these developers have acquired through the years to maintain vital legacy applications, but also enable them to quickly develop *n*-tier applications – without dealing with the complexities of the underlying *n*-tier platform.

Simplify Integration of Legacy and Enterprise Systems

Enterprise applications for on-demand enterprises are never standalone; they connect to other legacy and mainframe systems across the enterprise and serve as the vital link between the enterprise and its customers and partners. On demand organizations need the ability to quickly construct Web applications that integrate with databases, messaging protocols, and information from a variety of application servers. Then, after the Web application is deployed, the organization needs an efficient approach for maintaining the underlying technologies, which are subject to constant change.

ARAD enables the technology architect to design and fine-tune agile, scalable applications

The ARAD process addresses these needs. It supports rapid application integration with J2EE and other new techniques that help teams quickly leverage information from disparate sources. Users can then select target technologies for deployment without extensive knowledge of J2EE.

Realize Agile, Architected, Scalable *N*-tier Applications

IBM WebSphere Application Server and other *n*-tier runtime systems allow enterprise applications to scale to the needs of on demand computing. For example, connecting a large business to millions of customers. With this power, however, comes a level of complexity that challenges some application developers. To fully leverage an extensible *n*-tier architecture, a project team needs a technology architect – a person who understands the architecture and knows how to scale up from supporting 1,000 users on a single server to serving millions of users via a 100-server farm.

The ARAD process allows the architect to design and fine-tune *n*-tier patterns, using *n*-tier components such as JSPs, servlets, stateless and stateful session beans, and entity beans. Developers who keep their focus on the business can then apply these patterns as they create applications to support business processes.

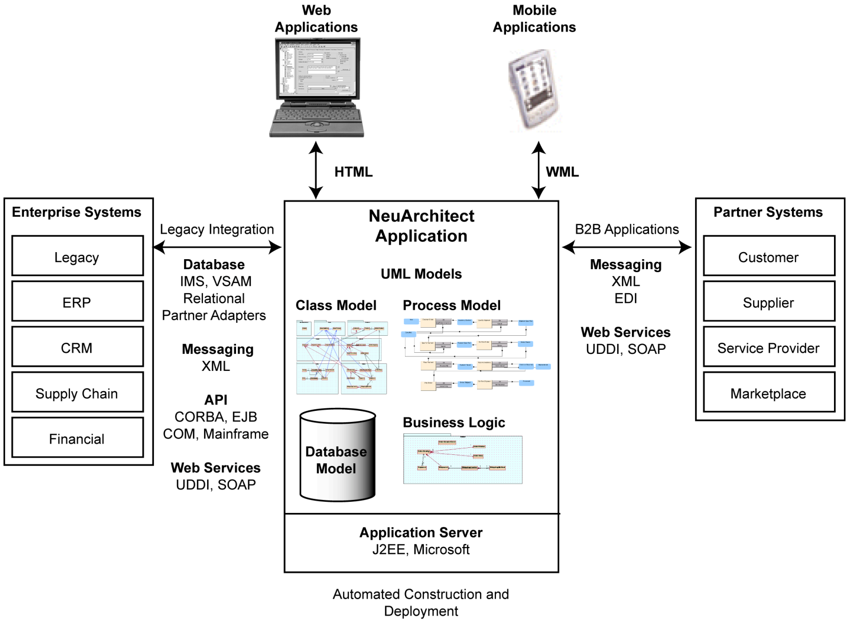
Gain Optimum Performance from IT and Staff Assets to Accelerate Application Delivery

Rational Rapid Developer empowers organizations to leverage IT assets more efficiently. That is, leverage human assets—the people who have deep domain expertise of your business, and your highly skilled but scarce J2EE developers and architects—across more projects, enabling you to effectively meet your businesses objectives. Rational Rapid Developer also enables you to rapidly access your legacy data and logic without requiring deep legacy expertise. All organizations need to leverage their existing IT assets in addition to their human assets. Legacy assets, be it yesterday’s implementation or a 20-plus-year-old legacy data source or program, can be effectively harnessed when building or extending applications using Rational Rapid Developer.

Types of Rational Rapid Developer Applications

Rational Rapid Developer is a versatile development platform suitable for a broad range of IT business applications. You can create new or composite applications that leverage your legacy IT assets.

Figure 1. Enterprise Application Features



With Rational Rapid Developer you can connect to other enterprise systems—logic or data—through a variety of mechanisms. You can rapidly access your relational data, hierarchical data such as IMS and VSAM, and business logic in CICS applications. You can easily incorporate Web services, standard platform components such as EJBs, and integrate them using XML messaging. The same capabilities can also be leveraged with integration with your partner’s systems. Rational Rapid Developer enables you to build high quality Web and wireless applications by highly leveraging your modeled application.

Application Development Process

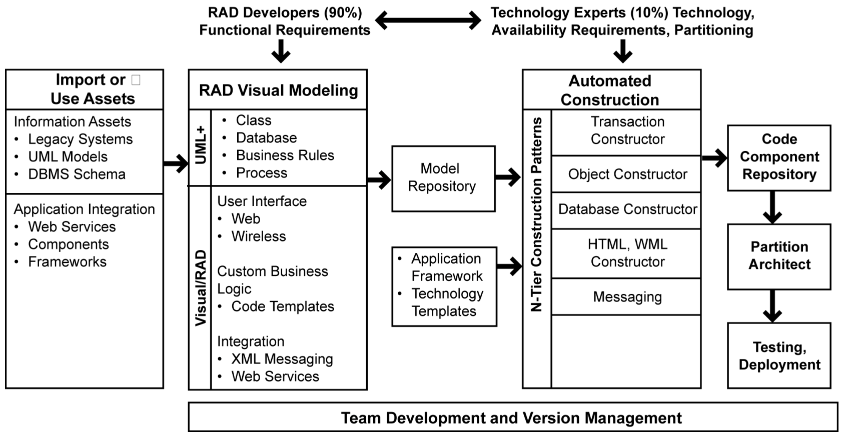


Figure 2. Rational Rapid Developer Development Process

The following is a step by step description of the primary Rational Rapid Developer development activities.

Step 1. **Import or Use Existing Assets**: UML class models from IBM Rational XDE®, IBM Rational Rose® or other UML modeling tool; DBMS schema and other legacy systems; Web services, and components.

Step 2. **Create Visual Models:** Creating the core UML models is twenty percent of the effort. Rational Rapid Developer provides essential UML models: class, database, business rules, and process models. As shown, most of the effort– typically about eighty percent – focuses on visually modeling the application: designing Web, wireless, XML messaging, and Web services transactions.

Step 3. **Make Application Deployment Decisions:** At this point, technology domain experts will decide how the applications will be constructed and deployed – for example, which databases, messaging platforms, and construction patterns to use.

Step 4**. Automatically Construct and Deploy Application:** Based on choices by the team's technical experts, Rational Rapid Developer automatically constructs the final application in a matter of minutes, and can automatically compile and deploy it locally.

Functional Requirements

Functional requirements are the starting point of your application. They are typically captured in use-case models or other textual documents. Developers take these functional requirements and translate them into a modeled application. Your focus is primarily on designing and implementing functional requirements—the business task at hand.

Save time by leveraging existing relational databases; data from IMS and VSAM; CICS applications; and existing UML models

Leverage Your Assets

Rational Rapid Developer can also save tremendous amounts of time by jumpstarting your application by leveraging your existing data sources. You can point to a data source and import a database schema from one of the leading relational databases into your class model. With this database reverse engineering capability, you can then either use these classes and existing underlying database tables and data in your application, or you can forward-construct that model into a new schema for a different database vendor and version. Rational Rapid Developer also provides access to IMS and VSAM data as well as CICS legacy mainframe application environments. You can have any number of data sources as part of your Rational Rapid Developer application. The developers only need to focus on invoking the appropriate data access methods on the business objects and not worry about the underlying data access (i.e., be concerned about whether data access is to a newly persisted data source, existing database table, or hierarchical data source).

A frequent requirement in enterprise application development is to harness business functionality that has been modeled in UML modeling tools such as Rational Rose or Rational XDE. Rational Rapid Developer enables development shops that have adopted UML modeling as a standard to accelerate the implementation phase of the development lifecycle by importing and synchronizing these UML class diagrams into Rational Rapid Developer, enabling you to rapidly build or maintain your information model. This is accomplished using the open, industry-standard XML Metadata Interchange (XMI) interface, as well with a native synchronization feature with Rational Rose and Rational XDE. Rational Rapid Developer also has native class modeling capabilities.

Rapid Visual Design

Once you have leveraged existing assets into your information model, you can extend these classes; add attributes, business rules, and business logic or methods to support your application. Rational Rapid Developer constructs system methods (e.g., data accessor methods) for you. These methods are constructed based on your underlying business model and transaction model. You can always see what system methods will be constructed and override them, if desired. You can view and override system-generated code on all n-tiers, giving you true flexibility.

After you have created your information model, you can focus on developing your business transactions or interactions. This could be rapidly developing Web or wireless pages, or designing messages to handle XML message transactions in/out-bound to your application. It could be focusing on creating components that you will eventually construct as services for other applications to invoke, or modeling Web services that you will expose and optionally have automatically registered to a UDDI registry.

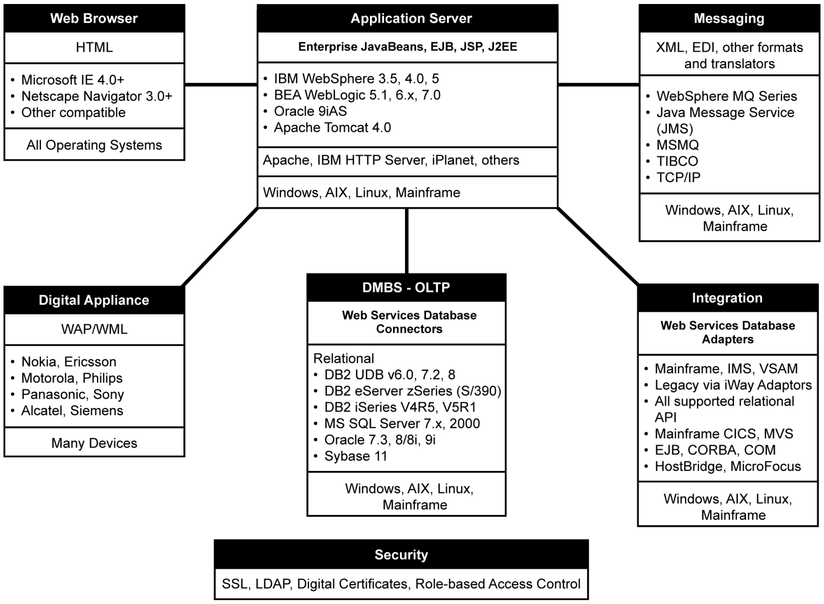
At this point the agile application model is complete. You have yet to specify technology construction requirements, or construct or deploy code.

Technology Selection

Now with your modeled application, you can specify the construction technologies, construction patterns, and deployment packaging. With Rational Rapid Developer, you can re-construct and re-deploy your application to other platforms, other vendors, and other versions of technologies at any time. It is important to note that the technical requirements of the application are independent of the functional requirements. Technical requirements start with identifying the specific *n*-tier deployment technologies that will be used to deploy your application. Technical requirements also include the capacity, performance, scalability and availability needs of the application that drives the design of a deployment model. Technical requirements are specified before you construct the application.

Using your selected technology configuration, Rational Rapid Developer automatically converts your model into high-quality, *n*-tier deployed code within minutes. The complexity of *n*-tier applications is removed from application development and is encapsulated as an engineering discipline in the construction system. Construction is optimized to your specific deployment technologies. Your constructed application, in conjunction with the Rational Rapid Developer Application Framework, enables the ability to build high quality, highly scalable, enterprise-class applications.

Figure 3. Rational Rapid Developer Technology Selection

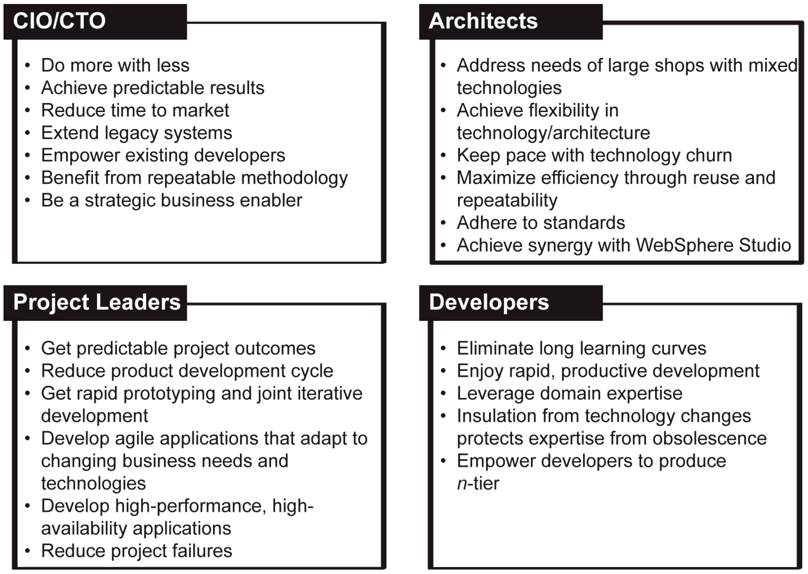

Web Browser
HTML
• Microsoft IE 4.0+
• Netscape Navigator 3.0+
• Other compatible
All Operating Systems
Application Server
Enterprise JavaBeans, EJB, JSP, J2EE
• IBM WebSphere 3.5, 4.0, 5
• BEA WebLogic 5.1, 6.x, 7.0
• Oracle 9iAS
• Apache Tomcat 4.0
Apache, IBM HTTP Server, iPlanet, others
Windows, AIX, Linux, Mainframe
Messaging
XML, EDI, other formats and translators
• WebSphere MQ Series
• Java Message Service (JMS)
• MSMQ
• TIBCO
• TCP/IP
Windows, AIX, Linux, Mainframe
Digital Appliance
WAP/WML
• Nokia, Ericsson
• Motorola, Philips
• Panasonic, Sony
• Alcatel, Siemens
Many Devices
DMBS - OLTP
Web Services Database Connectors
Relational
• DB2 UDB v6.0, 7.2, 8
• DB2 eServer zSeries (S/390)
• DB2 iSeries V4R5, V5R1
• MS SQL Server 7.x, 2000
• Oracle 7.3, 8/8i, 9i
• Sybase 11
Windows, AIX, Linux, Mainframe
Integration
Web Services Database Adapters
• Mainframe, IMS, VSAM
• Legacy via iWay Adaptors
• All supported relational API
• Mainframe CICS, MVS
• EJB, CORBA, COM
• HostBridge, MicroFocus
Windows, AIX, Linux, Mainframe
Security
SSL, LDAP, Digital Certificates, Role-based Access Control

Rational Rapid Developer provides optimized support and automated construction for a comprehensive selection of leading deployment technologies across all tiers of an *n*-tier system.

Construction and Deployment

Once you have developed your application, or even just parts of it, and have selected your preferred technologies and construction patterns, you can automatically construct the *n*-tiers of your application with a mouse click. Rational Rapid Developer leverages the appropriate technology construction templates to automatically construct your application. It can construct DDL and sample data into a database or to scripts. It constructs all required Java classes, beans, JSPs, HTML, WML, and Web Services on demand.

When you want to deploy your application to a remote environment such as a QA staging or production environment, Rational Rapid Developer enables this quickly. Rational Rapid Developer lets you define or model your deployment technologies and functional and physical deployment partitions, and creates optimized deployment enterprise archives (EAR files) as required. These can then be deployed in your environments.



CIO/CTO

- Do more with less
- Achieve predictable results
- Reduce time to market
- Extend legacy systems
- Empower existing developers
- Benefit from repeatable methodology
- Be a strategic business enabler

Architects

- Address needs of large shops with mixed technologies
- Achieve flexibility in technology/architecture
- Keep pace with technology churn
- Maximize efficiency through reuse and repeatability
- Adhere to standards
- Achieve synergy with WebSphere Studio

Project Leaders

- Get predictable project outcomes
- Reduce product development cycle
- Get rapid prototyping and joint iterative development
- Develop agile applications that adapt to changing business needs and technologies
- Develop high-performance, high-availability applications
- Reduce project failures

Developers

- Eliminate long learning curves
- Enjoy rapid, productive development
- Leverage domain expertise
- Insulation from technology changes protects expertise from obsolescence
- Empower developers to produce *n*-tier

Figure 4. Value Proposition for IT Stakeholders

Rational Rapid Developer provides deep value to a broad set of IT stakeholders – whether you are the IT executive needing “to do more with less”; the development manager needing to harness existing assets and domain expertise of your non-J2EE developers; the software or system architect needing to ensure architecture consistency and be enabled to adopt new technologies; or the developer who wants to rapidly build business applications leveraging cutting-edge technologies and platforms – you can highly leverage your J2EE and deployment expertise broadly across many projects. You can immediately enable your domain and broad-skilled corporate developers (HTML, JavaScript, Visual Basic, PowerBuilder, Oracle Forms, COBOL) to participate in building robust, enterprise-class applications.

Highlighted Product Capabilities

Business Modeling

Class Modeling

Class Architect is Rational Rapid Developer’s object-oriented, visual class modeling tool. It allows you to focus on analyzing the business domain and translating the business domain into an appropriate object design. The object-oriented focus produces smaller systems through the reuse of common components and produces systems that are more resilient to change and better able to evolve over time. Class Architect’s design features directly address the inherent complexities of software by helping you make intelligent decisions.

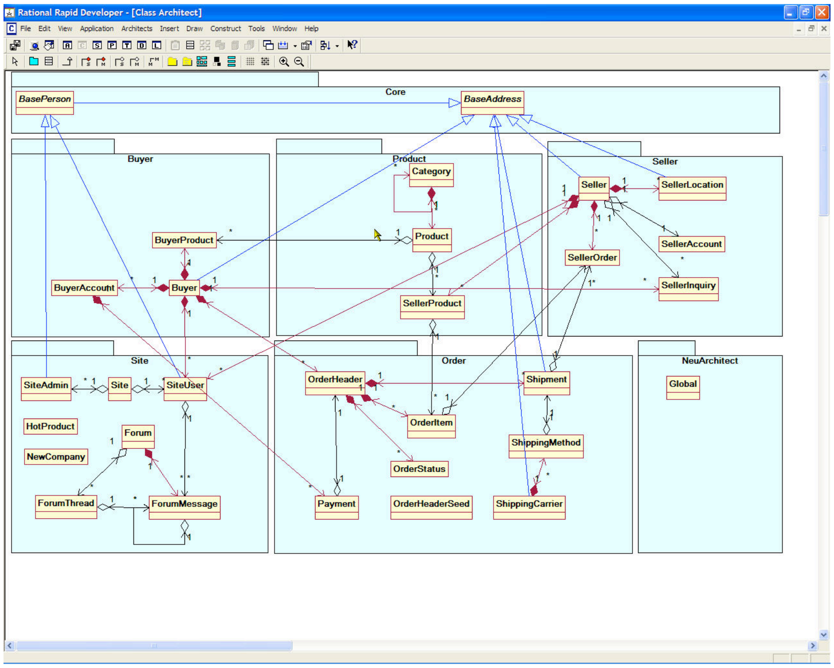


Figure 5. Class Architect

Using Class Architect, a business expert’s understanding and requirements are directly captured as class models. Class models support an unlimited number of abstract, non-persistent, and persistent classes. You define class attributes and methods, define inheritance relationships, and capture associations such as multiplicity, navigability, and composition or ownership relationships. The resulting class models form the superstructure of the application (e.g., data, middle-tier and client user interface) and allow developers to leverage this information directly and efficiently throughout the integrated visual tools in Rational Rapid Developer. Whether you are designing a Web page, extending your object with custom business logic, or designing XML messages, modeled class information is available to rapidly and accurately build your application.

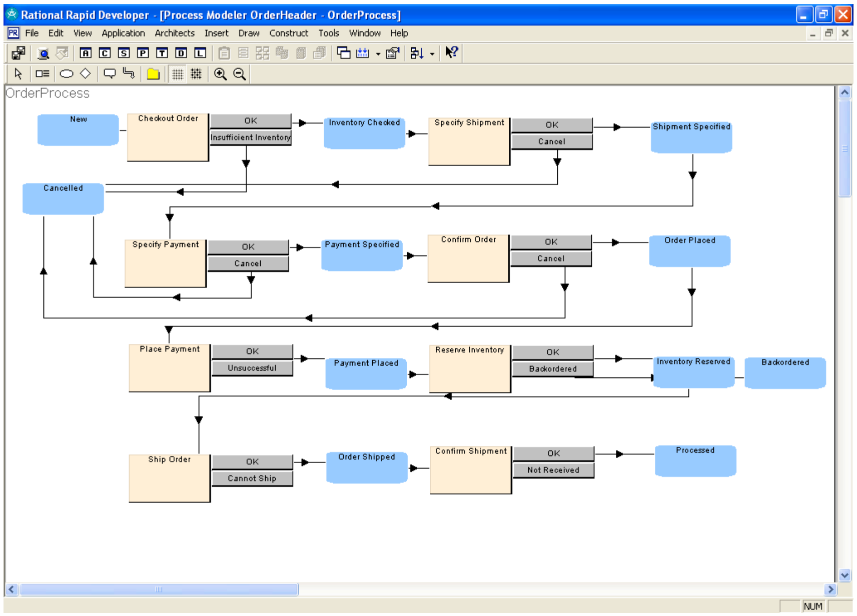
Business Rules

The Business Rules modeling in Rational Rapid Developer provides the ability to attach language-based expressions to an attribute so you can extend the semantics of an attribute at the object level. Business rules are simply business logic and should be maintained by business analysts. Typical business rules that can be quickly defined and deployed are rules for initial attribute value, derivation value, and validation rules.

Process Architect

The visual modeling features incorporated in Process Architect help an application designer or analyst to accurately model the business processes that must be embodied and supported by the application. You use the Process Architect to describe a process or sequence of activities and related states. The Process Architect captures state and activity. Modeling elements include states, methods, outcomes, branches, merges, and decisions.

Figure 6. Process Architect



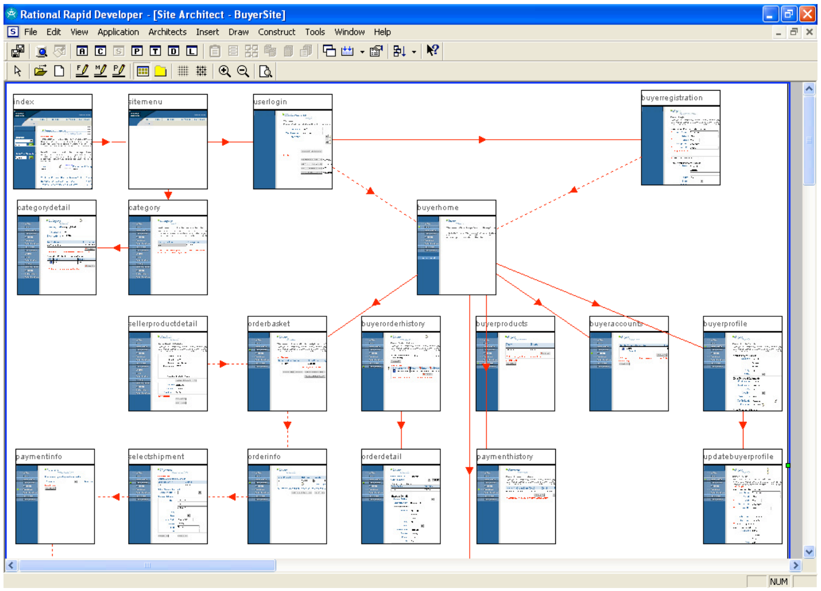
User Interface Modeling

Rational Rapid Developer provides a comprehensive system for user interface modeling. It uses the Site Architect to model user navigation, the Theme Architect for consistent look-and-feel, and the Page Architect to visually design the user interface. The most unique component is the transaction model embedded in each page that links it to business objects and data. This enables the page to provide full application interaction.

Site Architect

The Site Architect provides the ability to visually model the user experience at your Web site. It lets you create a storyboard or outline of the website using graphical tools to specify the individual pages, define linkages or relationships between pages, and show navigation paths. These pages can be displayed within a frameset using a menu selection or as part of a process chain.

Figure 7. Site Architect



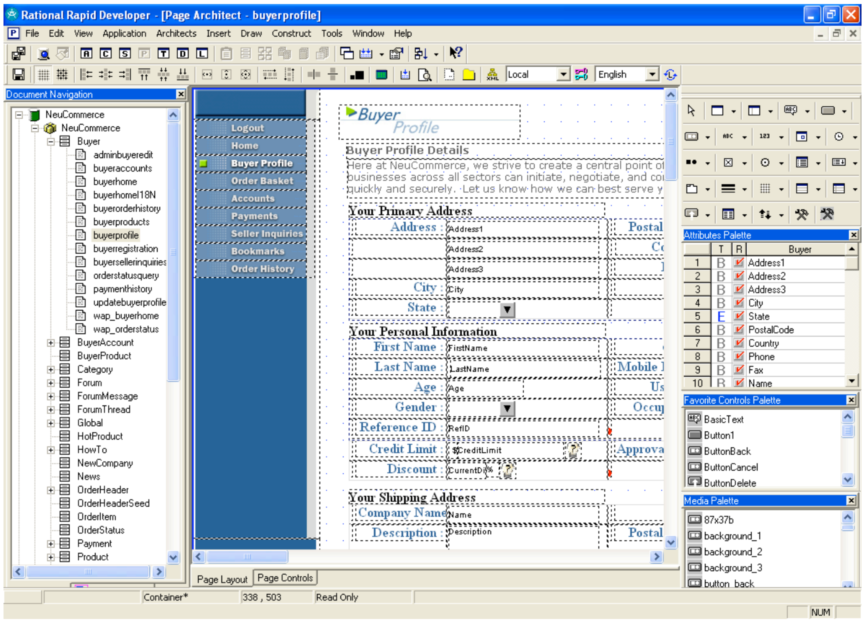
Theme and Style Architects

Rational Rapid Developer enables the graphic designer to create and select any number of presentation formats from which to construct a site's Web documents. Theme Architect offers the capability to dynamically change the look-and-feel of a site's pages. A theme is composed of several components, each defining a specific aspect of the presentation layer: color (defines sets of colors for controls), font (defines sets of fonts to be used in controls), and image (defines different sets of images to be used in the site). All colors, fonts and images defined in their respective themes are identified by a user-defined name.

Page Architect

The WYSIWYG Web page authoring tool, Page Architect, contains a rich set of graphical controls to support state-of-the-art user interface design. In addition, documents created with Page Architect are automatically connected to the middle-tier through the code construction facility. This powerful feature eliminates the need to write presentation logic for building data-connected Web pages. Page Architect supports both HTML-based browsers and WML/WAP-based devices. With Rational Rapid Developer, it is also easy to integrate existing visual components such as Java applets, custom HTML controls, and ActiveX components. These are simply placed on the document pages and can be easily customized to meet specific requirements.

Figure 8. Page Architect



Here at NeuCommerce, we strive to create a central point o
businesses across all sectors can initiate, negotiate, and co
quickly and securely. Let us know how we can best serve y

Intelligent Controls and the ObjectSpace

ObjectSpace refers to the Rational Rapid Developer-created “workspace object” produced by the constructed code. The ObjectSpace handles data mapping and transaction processing between browser, business objects, and database. The ObjectSpace for a web page determines the definition of business object instances created on the web server or application server at run-time. The business object instances are optimized to serve the web pages and handle presentation and persistence logic, doing things like populating the page with browser data (either data being sent to the browser or being sent by the browser via form posting), and reading or writing data to and from the database.

The localization model provides a layer of abstraction that greatly eases the task of developing and maintaining multi-lingual, multi-locale applications

This feature provides tight integration between graphical controls and the underlying data. For example, writing code to display data is unnecessary because this logic is done automatically by Rational Rapid Developer's ObjectSpace engine.

Section 508 Web Accessibility

Rational Rapid Developer provides support for Section 508 Web accessibility requirements. In 1998, the United States Congress amended the Rehabilitation Act to require Federal (central) agencies to make their electronic and information technology accessible to people with disabilities. Section 508 was enacted to eliminate barriers in information technology, to make available new opportunities for people with disabilities, and to encourage development of technologies that will help achieve these goals. Rational Rapid Developer has RAD support for these requirements. Support includes items such as ALT tag or help field text support, column and row header tags, cell headers, frame titles, and more.

Multi-lingual, Localized Applications

Rational Rapid Developer provides built-in capabilities for rapidly creating, deploying, and maintaining multi-lingual, multi-locale applications. Just as with all other features of Rational Rapid Developer, the ability to create applications that support multiple lanaguage, formats, and locale specifics is a rapid and easy process. The Rational Rapid Developer localization model abstracts away the many pains in creating these applications, giving you an unparalleled competitive advantage and cost savings in creating and maintaining international/localized applications.

Table 1. Rational Rapid Developer Localization Model Solution

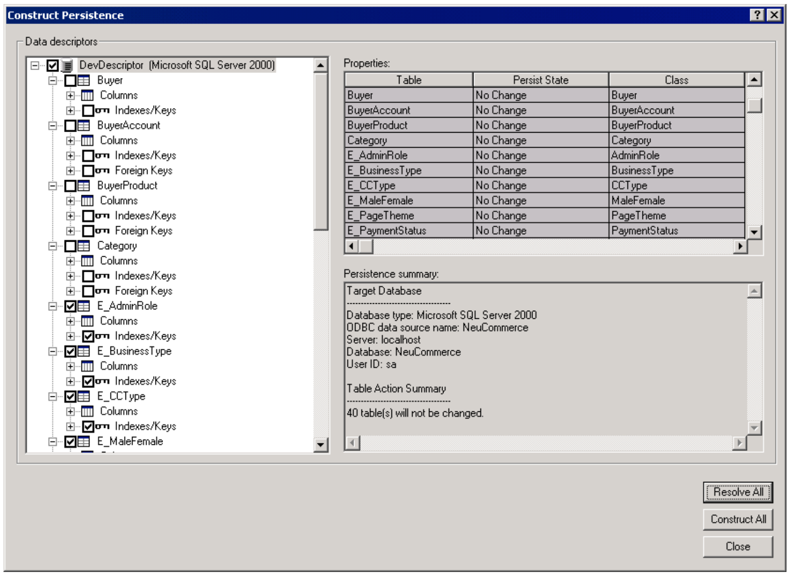
Problem	Solution
Difficult and costly to train an entire team on the complexities of internationalization (i18n).	The localization model insulates the application developer from the complexity of i18n without sacrificing flexibility. The application developer can focus on the task of building the application without worrying about the complexity of localization.
The process of translation can be cumbersome, costly, and inefficient.	Rational Rapid Developer provides an integrated system to send and receive translations from translators. The information is sent in industry standard formats (TMX, XLIFF), Excel spreadsheet format, or CVS files. Translated strings are automatically integrated back into the application model.
It is difficult to create and manage text strings across languages.	The developer does not have to work with resource bundles. Resource bundles are automatically generated and deployed.
It is difficult to determine the effect of translation on page design and layout	In traditional development environments, you must do this at runtime. With the visualization provided by the Rational Rapid Developer Page Architect, a developer or translator can immediately see the effects of translation on the page layout. We have added another dimension to WYSIWYG, that is multiple language support.
Images are hard to manage across languages.	Rational Rapid Developer automatically manages images across multiple languages.
It's very difficult to localize an application once it has been written.	There is no need to decide to localize up front. The Rational Rapid Developer localization model facilitates the migration of non-localized applications to a multi-language applications.
Multiple versions of pages, each in a different language, are very difficult to maintain	Rational Rapid Developer allows you to design one page that displays in different languages (determined by the operating system language settings on the user's computer), greatly reducing maintenance.

For more information, see the IBM Rational Rapid Developer, Globalization and Multi-Locale Application technology paper.

Database Architect

You define class persistence using database modeling. Rational Rapid Developer's database modeling enables you to map the logical model to a physical model, while providing the necessary abstraction that gives you the ability to quickly change your deployment database. Rational Rapid Developer's database modeling capabilities allow you to:

- Construct physical database persistence layers as defined by the class model.
- Resolve differences between the modeled and an existing constructed database layer, and then alter an existing database to reflect new models.
- View all DDL so that you can create and drop statements before constructing persistence.
- Set database-specific optimization parameters.



Data descriptors

- ☑ DevDescriptor (Microsoft SQL Server 2000)
  - □ Buyer
    - Columns
    - □ Indexes/Keys
  - □ BuyerAccount
    - Columns
    - □ Indexes/Keys
    - □ Foreign Keys
  - □ BuyerProduct
    - Columns
    - □ Indexes/Keys
    - □ Foreign Keys
  - □ Category
    - Columns
    - □ Indexes/Keys
    - □ Foreign Keys
  - ☑ E_AdminRole
    - Columns
    - ☑ Indexes/Keys
  - ☑ E_BusinessType
    - Columns
    - ☑ Indexes/Keys
  - ☑ E_CCType
    - Columns
    - ☑ Indexes/Keys
  - ☑ E_MaleFemale

Properties:

Table	Persist State	Class
Buyer	No Change	Buyer
BuyerAccount	No Change	BuyerAccount
BuyerProduct	No Change	BuyerProduct
Category	No Change	Category
E_AdminRole	No Change	AdminRole
E_BusinessType	No Change	BusinessType
E_CCType	No Change	CCType
E_MaleFemale	No Change	MaleFemale
E_PageTheme	No Change	PageTheme
E_PaymentStatus	No Change	PaymentStatus

Persistence summary:

```
Target Database
-----------------------------------
Database type: Microsoft SQL Server 2000
ODBC data source name: NeuCommerce
Server: localhost
Database: NeuCommerce
User ID: sa

Table Action Summary
-----------------------------------
40 table(s) will not be changed.
```

Figure 9. Constructing Persistence Using the Database Architect

Importing Data Models

Rational Rapid Developer protects legacy investments by supporting reverse engineering of existing databases. With Rational Rapid Developer’s database import wizards, you can take existing database schemas and create classes from them for use in your own application. You can accomplish this using a direct connection to the database, or using an XML-based catalog import. Class relationships can also be created, based on foreign key relationships in the database.

Multi-Database Support

Rational Rapid Developer provides the ability to specify the data source for each class being modeled, facilitating integration with existing legacy systems. The data source can be any database supported by Rational Rapid Developer. For example, Product and Category classes may be persisted in a DB2 database, while Customer and Account classes can be persisted in an Oracle database. Rational Rapid Developer ensures that the persistence interface is directed to the appropriate data source.

Referential Integrity and Multiple Key Support

Referential integrity describes the mandatory existence of parent-row primary key values for dependent-row foreign-key relationships. Rational Rapid Developer enforces referential integrity if it is defined in the business model. Rational Rapid Developer provides the capability to create relationships between entities based on multiple keys. This allows modeling of complex relationships in which a parent entity’s key becomes part of its child’s composite key, and so on. Rational Rapid Developer also:

Integrated data entry and sample data generation speed up prototyping and reduce the time required to set up application tests

- Supports cascaded relationships and generates appropriate triggers if the database requires it, and generates appropriate DDL.
- Maintains relationships and supports complex primary keys and custom matching relationships.
- Provides built-in support for relationships via surrogate keys.
- Supports domains for reducing joins.
- Abstracts JDBC complexities from the user.

Data Entry

The Rational Rapid Developer Data Entry feature facilitates testing by allowing you to directly enter data without using external tools or scripts for data creation. You can use this feature in conjunction with the Sample Data Generation feature to quickly set up any amount of test data.

Sample Data Generation

Finding adequate sample data is a typical problem in the development and testing of sites. You can use Rational Rapid Developer's Sample Data Generation feature throughout the development cycle for prototyping, development, testing, and training. From the developer's point of view, this feature facilitates prototyping by eliminating data issues. From the tester's point of view, there is always the need to generate test data for sites having varying volumes of data. Rational Rapid Developer eliminates the need to find additional tools to generate data or the need to write a program to generate the data. For applications with a complicated object model, you can use Sample Data Generation to instantaneously test the site.

Security Architect

Applications leveraging Rational Rapid Developer's security framework are not constrained to any particular vendor of Directory Services or technology. A common gateway allows Rational Rapid Developer-constructed applications to work with the most popular Directory Service technologies at runtime. These include LDAP server products (e.g., from Sun, Netscape), Microsoft's Active Directory, or other relational database directory services. The Rational Rapid Developer security framework supports simple password, X.509 digital certificates, smart cards, biometrics, authentication timeout techniques, and additional security standards. Multiple Security Provider instances allow an application to connect to multiple types of Directory Services.

Rational Rapid Developer also provides built-in support for data and role-based security.

The security framework provides the ability to change schema mapping or directory service settings without regenerating the application or rebuilding runtime components

Authentication

Each generated Web site creates and maintains a user profile that includes information such as name, title, address and phone number. Profiles can also include global information like departments, groups, or divisions. The Rational Rapid Developer platform provides (among other authentication methods) role-based management of access rights and their presentation to the developer. A role is a set of access privileges assigned to a user when their profile is created. Profiles and roles are maintained through the directory manager's administration facility.

Authentication is performed at runtime. Each resource request generates a search against the Rational Rapid Developer Security Provider to verify that the user has current authentication privileges.

Authorization

Rational Rapid Developer application developers can personalize content down to the control level within a document based on the user's identity. Users see only the resources that they are entitled to use while restricted functions remain hidden. If the user has insufficient privileges, the developer can set the action to automatically redirect the user to another resource.

Encryption is performed using a Secure Sockets Layer (SSL) connection to maintain secure data transfer during authentication and authorization operations. Data such as username and passwords are protected during transmission by an infrastructure that keeps unauthorized parties from reading or modifying data. URL parameter and XML message encryption is also supported.

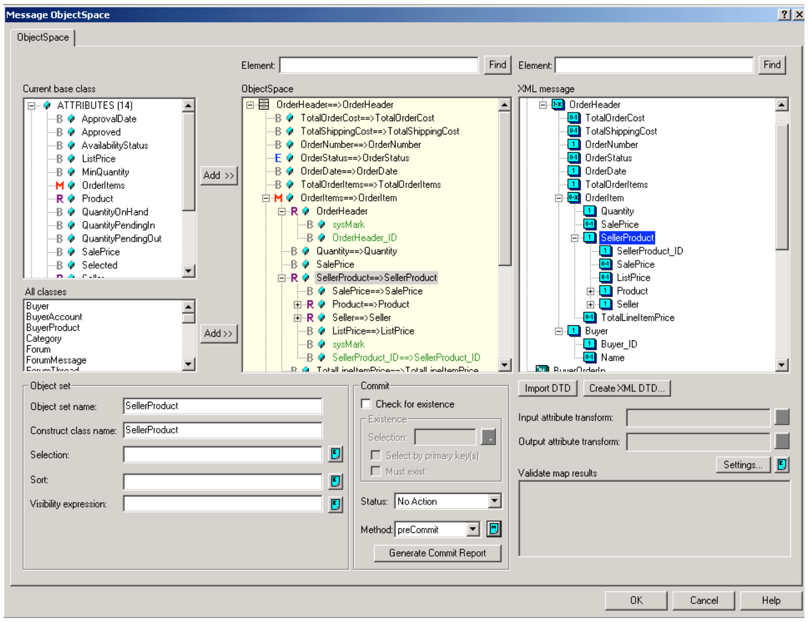
Directory Services Administration

The Rational Rapid Developer security framework provides the ability to change schema mapping or directory service settings without regenerating Rational Rapid Developer documents or objects or rebuilding runtime components. This is especially important during the development, testing, and deployment phases of application building.

Messaging

Rational Rapid Developer supports the ability to integrate with other enterprise or partner systems in a variety of ways including leveraging messaging systems and technologies. You can rapidly model your inbound and outbound messages using both XML or custom to support your specific business needs. The developer will focus on definining what entities, attributes, and methods are required to support the specific transaction (such as processing an inbound customer order), what business methods to be automatically invoked, and mapping the message to the appropriate ObjectSpace or transaction. Rational Rapid Developer will handle the rest.

Figure 10. Message Modeling



Object set name: SellerProduct

Construct class name: SellerProduct

Status: No Action

Method: preCommit

Rational Rapid Developer's messaging framework provides a sophisticated multi-threaded message broker (or you can utilize J2EE Message-Driven Beans) to process inbound messages and automatically routes the message to the correct component. Inbound message queues can be configured for any number of concurrently executing threads, depending on machine resource limits. Outbound messages are sent directly through the messaging system to their destination. Users can define as many different input queues as needed, limited only by the hardware resources. A variety of messaging systems can be supported, including:

- Native support for JMS providers, IBM WebSphere MQ, TIBCO, and MSMQ
- Support for XML messages and DTDs
- Support for plain text messages
- Supports synchronous and asynchronous messaging
- Supports complex transformations
- Constructs parser and formatters based on the XML to the DTD mapping

Developers can integrate with legacy applications without having to understand mainframe details or technologies

Legacy Integration

Traditional legacy systems have evolved with limited thought for future integration requirements. Typically, these older systems were developed or purchased for a specific department within the company or designed to solve a narrow set of problems. Most legacy systems are unable to interface beyond their own environment. In this era of diverse technology solutions, any integration issue quickly becomes an integration problem. Rational Rapid Developer addresses these problems.

Rational Rapid Developer provides the ability to interface with existing IBM mainframe legacy applications (databases and enterprise application services). The Rational Rapid Developer mainframe integration adapters enable you to access IMS or VSAM data without having to understand the mainframe details or technologies. They are treated as yet another data source for your application developers to access. Likewise, you can access CICS applications from Rational Rapid Developer in the same easy fashion.

Rational Rapid Developer also provides support for many commercial mainframe solutions in order to rapidly leverage 3270 terminal applications in your new or composite application. So the problem of getting information buried deep in green screen transactions and integrating this information with the rest of your application or enterprise gets solved easily with Rational Rapid Developer. For more information, see the IBM Rational Rapid Developer, A Guide to Legacy Integration technology paper.

Components

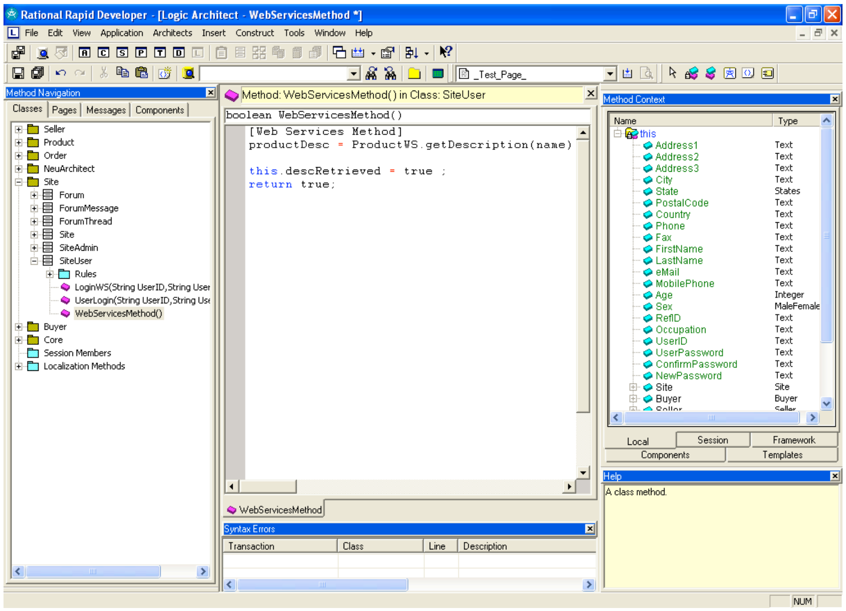
With Rational Rapid Developer, you can model components that enable support of a services-oriented architecture. These component leverage your information model and can be optimized for the services they support. For J2EE, these components at construction time translate to Session beans. These can be stateless or stateful, or even construction to non-J2EE, Java packages. These components have user-defined interfaces and can be automatically constructed and deployed. These constructed components can be consumed by any J2EE application, and are ideal for reuse across the enterprise.

Web Services

Rational Rapid Developer supports the discovery and usage of existing Web services. In addition, Rational Rapid Developer modeled components can be exposed as Web services with a click of the mouse. These Web services can also be automatically registered to a UDDI registry of your choice. Using Web services in Rational Rapid Developer is rapid and easy. The developer is insulated from SOAP protocols and underlying APIs.

For more information, see the IBM Rational Rapid Developer, Components and Web Services technology paper.

Figure 11. Web Services



- Seller
- Product
- Order
- NeuArchitect
- Site
  - Forum
  - ForumMessage
  - ForumThread
  - Site
  - SiteAdmin
  - SiteUser
    - Rules
    - LoginWS(String UserID,String User
    - UserLogin(String UserID,String Use
    - WebServicesMethod()
- Buyer
- Core
- Session Members
- Localization Methods

Method: WebServicesMethod() in Class: SiteUser

```
boolean WebServicesMethod()
[Web Services Method]
productDesc = ProductWS.getDescription(name)

this.descRetrieved = true ;
return true;
```

Name	Type
this	
Address1	Text
Address2	Text
Address3	Text
City	Text
State	States
PostalCode	Text
Country	Text
Phone	Text
Fax	Text
FirstName	Text
LastName	Text
eMail	Text
MobilePhone	Text
Age	Integer
Sex	MaleFemale
RefID	Text
Occupation	Text
UserID	Text
UserPassword	Text
ConfirmPassword	Text
NewPassword	Text
Site	Site
Buyer	Buyer
Seller	Seller

A class method.

Transaction	Class	Line	Description

Team Development

Rational Rapid Developer supports a true multi-user development environment. Team Development support enables project team members to perform version control activities such as check-in/out and update functions on project components. The team development environment supports IBM® Rational® ClearCase as well as other commercial version control systems that utilize the Microsoft® SCC API (Microsoft Visual Source Safe, Merant® PVCS, and MKS® Source Integrity, etc.). Integrated features include the ability to: Add and Remove from Source Control, Get Latest Version, Check In, Check Out, Undo Check Out, Disconnect from Source Control, Open Source Control Project, and Run Source Control. Support for automated backups is also available.

Automated Deployment

Deployment of business applications is an iterative development process that can be a time-consuming and frustrating activity. Rational Rapid Developer removes of burden of deployment so that the developer can concentrate on the task at hand, namely application development. When developing on a local machine, the developer simply presses the construct button. Your code is generated, compiled, and then locally deployed automatically for you to any combination of virtually all leading deployment technologies. Rational Rapid Developer makes it easy.

For remote deployment to staging and production environments, Rational Rapid Developer creates the appropriate deployment artifacts, easing your remote deployment tasks. This is done using the Partition Architect. The Partition Architect enables you to specify any number of deployment environments and technolgies and will optimally construct the deployable artifacts for these environments. Functional and physical partition capabibiites are enabled.

Partition Architect

The Partition Architect provides unmatched flexibility in deploying Rational Rapid Developer generated artifacts. With Partition Architect, application developers can divide the application artifacts into archives along functional boundaries (e.g., order entry, purchasing, inventory) or along application tiers (presentation, middle tier, database) or any other way that makes sense for the application. For instance, session beans that represent high-use business transactions can be packaged as a deployment archive and deployed on a high volume server.

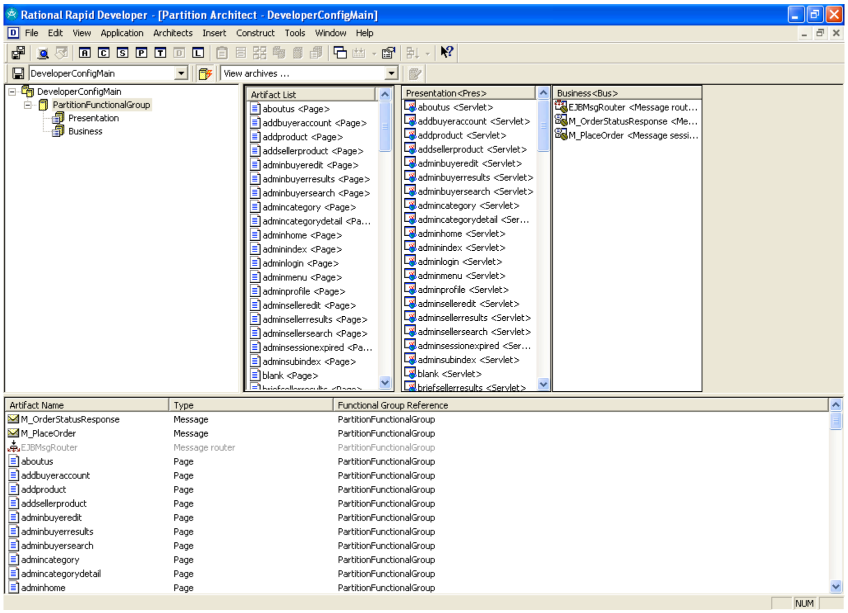

Rational Rapid Developer - [Partition Architect - DeveloperConfigMain]
File Edit View Application Architects Insert Construct Tools Window Help
DeveloperConfigMain
View archives ...
DeveloperConfigMain
PartitionFunctionalGroup
Presentation
Business
NUM

Figure 12. Partition Architect

Artifact List
aboutus <Page>
addbuyeraccount <Page>
addproduct <Page>
addsellerproduct <Page>
adminbuyeredit <Page>
adminbuyerresults <Page>
adminbuyersearch <Page>
admincategory <Page>
admincategorydetail <Pa...
adminhome <Page>
adminindex <Page>
adminlogin <Page>
adminmenu <Page>
adminprofile <Page>
adminselleredit <Page>
adminsellerresults <Page>
adminsellersearch <Page>
adminsessionexpired <Pa...
adminsubindex <Page>
blank <Page>

Presentation<Pres>
aboutus <Servlet>
addbuyeraccount <Servlet>
addproduct <Servlet>
addsellerproduct <Servlet>
adminbuyeredit <Servlet>
adminbuyerresults <Servlet>
adminbuyersearch <Servlet>
admincategory <Servlet>
admincategorydetail <Ser...
adminhome <Servlet>
adminindex <Servlet>
adminlogin <Servlet>
adminmenu <Servlet>
adminprofile <Servlet>
adminselleredit <Servlet>
adminsellerresults <Servlet>
adminsellersearch <Servlet>
adminsessionexpired <Ser...
adminsubindex <Servlet>
blank <Servlet>
briefsellerresults <Servlet>

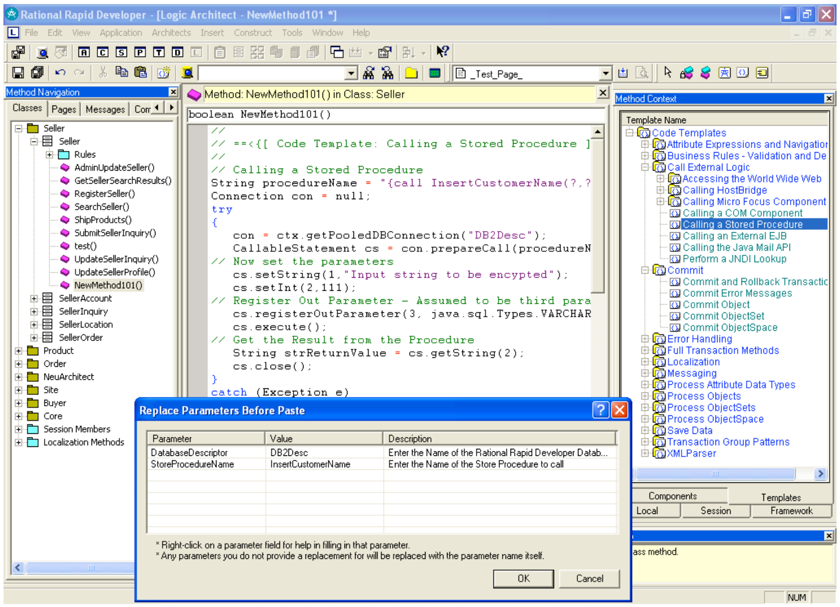
Business<Bus>
EJBMsgRouter <Message rout...
M_OrderStatusResponse <Me...
M_PlaceOrder <Message sessi...

Artifact Name	Type	Functional Group Reference
M_OrderStatusResponse	Message	PartitionFunctionalGroup
M_PlaceOrder	Message	PartitionFunctionalGroup
EJBMsgRouter	Message router	PartitionFunctionalGroup
aboutus	Page	PartitionFunctionalGroup
addbuyeraccount	Page	PartitionFunctionalGroup
addproduct	Page	PartitionFunctionalGroup
addsellerproduct	Page	PartitionFunctionalGroup
adminbuyeredit	Page	PartitionFunctionalGroup
adminbuyerresults	Page	PartitionFunctionalGroup
adminbuyersearch	Page	PartitionFunctionalGroup
admincategory	Page	PartitionFunctionalGroup
admincategorydetail	Page	PartitionFunctionalGroup
adminhome	Page	PartitionFunctionalGroup

Logic Architect

Rational Rapid Developer includes a coding environment called the Logic Architect. This is where the developer actually writes their business logic. In order to streamline this process and ensure code consistency between developers, Logic Architect provides a series of code templates to perform common tasks. The developer picks the code template to use, responds to questions from an automated wizard, and the Logic Architect then populates the editor with the appropriate code.

Figure 13. Logic Architect



Method: NewMethod101() in Class: Seller

```
boolean NewMethod101()
//
// ==<{[ Code Template: Calling a Stored Procedure ]
//
// Calling a Stored Procedure
String procedureName = "{call InsertCustomerName(?,?
Connection con = null;
try
{
   con = ctx.getPooledDBConnection("DB2Desc");
   CallableStatement cs = con.prepareCall(procedureN
// Now set the parameters
   cs.setString(1,"Input string to be encypted");
   cs.setInt(2,111);
// Register Out Parameter - Assumed to be third para
   cs.registerOutParameter(3, java.sql.Types.VARCHAR
   cs.execute();
// Get the Result from the Procedure
   String strReturnValue = cs.getString(2);
   cs.close();
}
catch (Exception e)
```

Replace Parameters Before Paste

Parameter	Value	Description
DatabaseDescriptor	DB2Desc	Enter the Name of the Rational Rapid Developer Datab...
StoreProcedureName	InsertCustomerName	Enter the Name of the Store Procedure to call

* Right-click on a parameter field for help in filling in that parameter.
* Any parameters you do not provide a replacement for will be replaced with the parameter name itself.

OK Cancel

In Figure 13, a template for calling a stored procedure was selected. The developer is prompted to enter the database descriptor name and the name of the stored procedure. Rational Rapid Developer automatically creates the code. This is an open system that allows the architect to modify the existing code templates as well as add new code templates. The bottom line is that the code the developer uses can be controlled to a large extent by the project architect or senior developer.

Conclusion

New technologies are rapidly emerging and new versions of existing technologies will continue to be released. To keep pace, businesses need to quickly adapt their existing applications to new technologies and business requirements without losing their investments in current systems. Rational Rapid Developer allows you to design and maintain applications as agile application models, rather than at the technology level. Your application model can be used to rapidly re-design, re-construct and re-deploy applications for the latest technologies.

IBM software integrated solutions

IBM Rational supports a wealth of other offerings from IBM software. IBM software solutions can give you the power to achieve your priority business and IT goals.

- DB2® software helps you leverage information with solutions for data enablement, data management, and data distribution.
- Lotus® software helps your staff be productive with solutions for authoring, managing, communicating, and sharing knowledge.
- Tivoli® software helps you manage the technology that runs your e-business infrastructure.
- WebSphere® software helps you extend your existing business-critical processes to the Web.
- Rational® software helps you improve your software development capability with tools, services, and best practices.

Rational software from IBM

Rational software from IBM helps organizations create business value by improving their software development capability. The Rational software development platform integrates software engineering best practices, tools, and services. With it, organizations thrive in an on demand world by being more responsive, resilient, and focused. Rational's standards-based, cross-platform solution helps software development teams create and extend business applications, embedded systems and software products. Ninety-eight of the Fortune 100 rely on Rational tools to build better software, faster. Additional information is available at www.rational.com and www.therationaledge.com, the monthly e-zine for the Rational community.

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

Printed in the United States of America
01-03 All Rights Reserved. Made in the U.S.A.

The IBM home page on the Internet can be found at **ibm.com**

TP910