White Paper

Rose PowerBuilder Link

Contents

Overview

Audience

This white paper is intended for Software Developers, Project Managers, Engineers and Analysts looking for proven ways to build better software faster and to fully understand the value of using the Unified Modeling Language (UML) to drive the software development process with PowerBuilder.

The Software Development Landscape

The Nature of Software Development

One of the major issues facing corporations today is the implementation of new software in new projects or updating and maintaining existing projects. Even though the latest development technology has been hailed as the solution to current corporate problems with rapid development, the majority of software projects do not come in on time, on budget, and/or do not fully meet user requirements. Failure is not just an inoperative application or abandoned project, but is also a project or application that falls far below expectations or that is not able to completely provide a business solution for the originally identified problem.

Better Software Development Methods

The latest software development methodologies are fundamentally changing the way companies do business in order to minimize the risk of failure and to deliver successful solutions faster. The combination of larger and more complex software applications, more competition, and shorter development cycles, has created the need for your development team to create higher quality software in less time.

Successful Software Development With UML

For successful software development, all members of a development team must communicate with a common language. Using UML enables all team members to use one tool and one language to communicate effectively. Whether your software team consists of 5 individuals or 500 individuals across 50 different groups, your

team can develop their components individually, communicate collaboratively and deliver better software in less time.

There are common issues in all successes and failures. Successful projects put more emphasis on reengineering, human capital investment, and change management; the projects that failed all showed that emphasis on these elements was lacking or non-existent. The challenge now is to take advantage of the latest technology and have active control of how that technology is used to improve software development quality and delivery time.

Sybase PowerBuilder

4th Generation Languages (4GL) greatly reduce software development time and provide an opportunity for non-computer specialists to produce applications that meet a professional standard. PowerBuilder from Sybase stands out as one of the most influential 4GL products ever engineered. Its success is a result of both the utility it brought to developers and the fact that it has been available for many years.

PowerBuilder offers the highest productivity available for enterprise-class development. PowerBuilder is an object-oriented, application development tool that allows you to build powerful, multi-tier applications to run on multiple platforms and to interact with various databases.

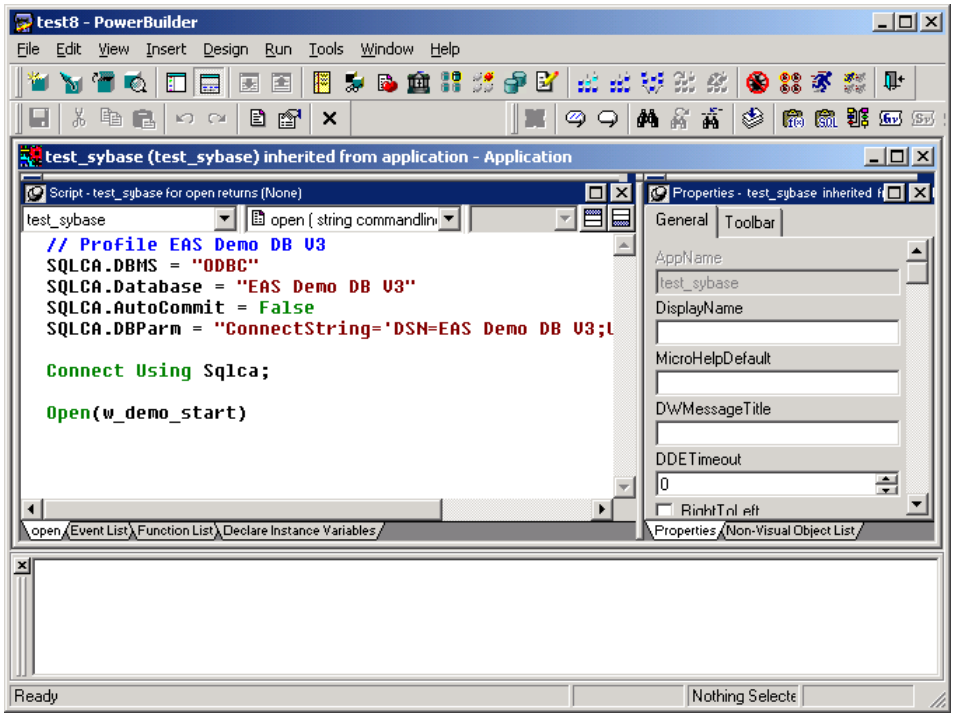

Figure 2: The PowerBuilder 8.0 Environment Where is Figure 1?

PowerBuilder is one of a group of Sybase products that together provide the tools to develop client/server, distributed, and Internet applications. PowerBuilder delivers automatic, industry-standard, component generation, limitless extension possibilities through HTML and Java™ Web-client support, and an unmatched freedom to work with a variety of technologies, all wrapped in a market-leading 4GL environment. Modeling the PowerBuilder design will bring even more efficiency to your software development by abstracting the design or your enterprise solution into a visual representation.

Rational Rose

Rational Rose is an ideal tool for object-oriented analysis, application modeling, design and construction. Industry experts, journal editors, and consumers have all recognized Rational Rose as a significant tool in the software development landscape.

Rational Rose allows you to move easily from analysis to design to implementation to testing, while supporting all team members throughout the project's lifecycle. It supports this dynamic change management process with forward engineering, reverse engineering, and model-updating features, allowing you to alter your implementation, assess your changes, and automatically incorporate them into your design.

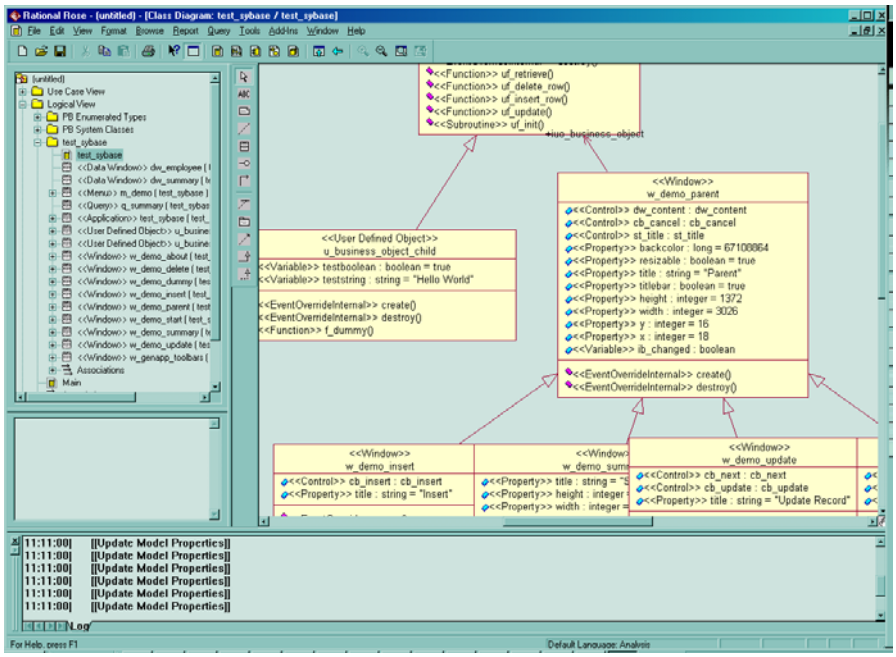

Figure 3: The Rational Rose Environment

Rational Software has over 20 years of industry leading experience and is acknowledged as the technology leader for its role in the development of the UML, largely due to the efforts of Grady Booch, Ivar Jacobson, and Jim Rumbaugh — the three chief UML authors. Partnered with the major players in the software industry, Rational Software will continue to play a significant role in the model-driven development market. Although Rational Rose supports a number of languages with its Round-Trip Engineering mechanisms, it does not support PowerBuilder.

Round-Trip Engineering Principles

Round-Trip Engineering (RTE) allows you to move from a design model of an application under development, convert it to application language source code, analyze it, and then refine it based on your increasing understanding of the problem. This process of modeling, code generation, analysis, and refinement is reiterated until you are satisfied with the completeness and correctness of your system. This controlled, iterative, style of development lets projects begin with a set of known design requirements, and evolve as project parameters change or new requirements are added. aThe value of this approach is evident when you consider how RTE fits into the software development life cycle as illustrated in Figure 4.

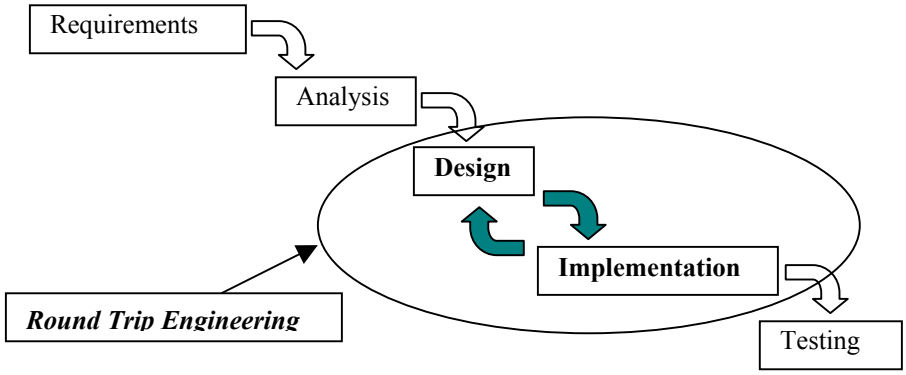

Figure 4: How RTE Fits Into Software Development Life Cycle Iteration

Synchronization of the design with the implementation ensures that the iterative development cycle is controlled and predictable, avoiding the costly and time-consuming changes that afflict most software projects today. The principles of RTE can be applied to any programming language and Rational Rose supports many of the common application-building languages used today. Metex has extended the application development benefits of RTE in Rational Rose to include the PowerBuilder language.

Rose PowerBuilder Link

What is Rose PowerBuilder Link?

Rose PowerBuilder Link™ is a utility developed by Metex to integrate Rational Rose with Sybase's PowerBuilder into an object-oriented design and implementation environment. Specifically, Rose PowerBuilder Link extends the Round-Trip Engineering ability in Rational Rose for applications developed in PowerBuilder.

The integration of Rational Rose with PowerBuilder allows developers to express business needs graphically using Rational Rose's visual modeling capabilities and to convert their models automatically into a functional implementation using PowerBuilder. With Rational Rose, developers create a logical model of a system, in which each component is defined independently of execution platform and implementation language. Rose PowerBuilder Link translates this model into PowerBuilder PowerScript, propagating design information into the application implementation.

Reverse Engineering

Rose PowerBuilder Link can reverse engineer existing PowerBuilder applications, permitting developers to visualize and modify PowerBuilder objects and their relationships.

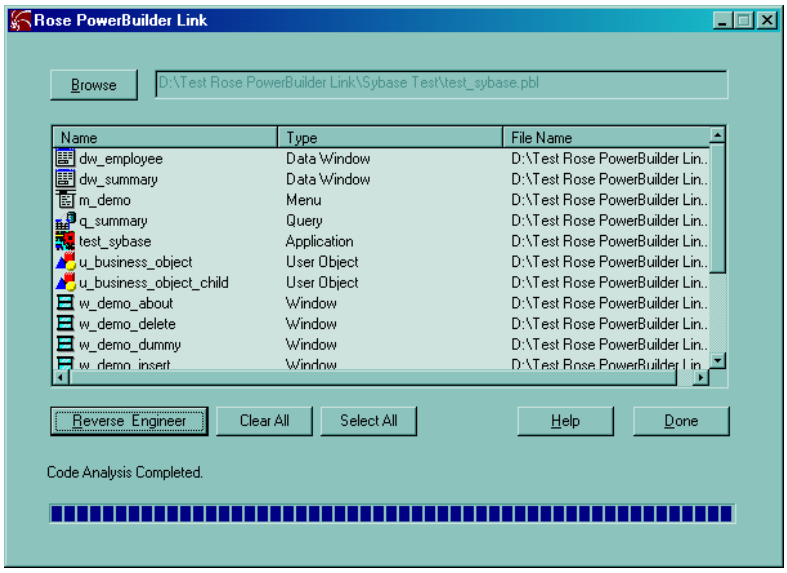

Figure 5: The Rose PowerBuilder Link Reverse Engineering Window

Once the Rose PowerBuilder Link finishes analysing the PowerBuilder language file, it closes the dialog box, and examines the class diagram in the Rational Rose environment. Our Rose PowerBuilder Link is able to capture inheritance, aggregations, and instantiation relationships automatically. I got a little lost here so I am not sure if I got the sequence of events described correctly.

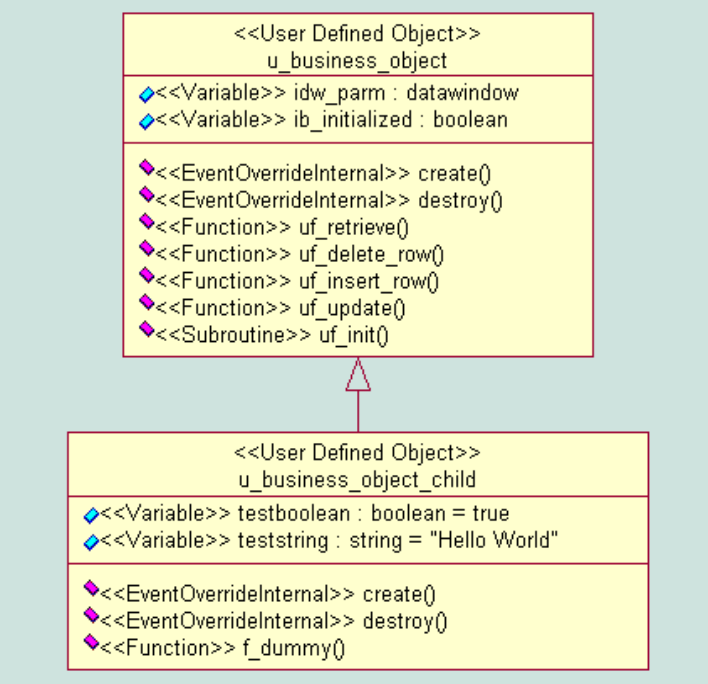

Figure 6: Example of an Inheritance Relationship Automatically Captured

Forward Engineering

Conversely, the design that is created in the Rational Rose class diagram can easily translate into a PowerBuilder implementation with Rose PowerBuilder Link.

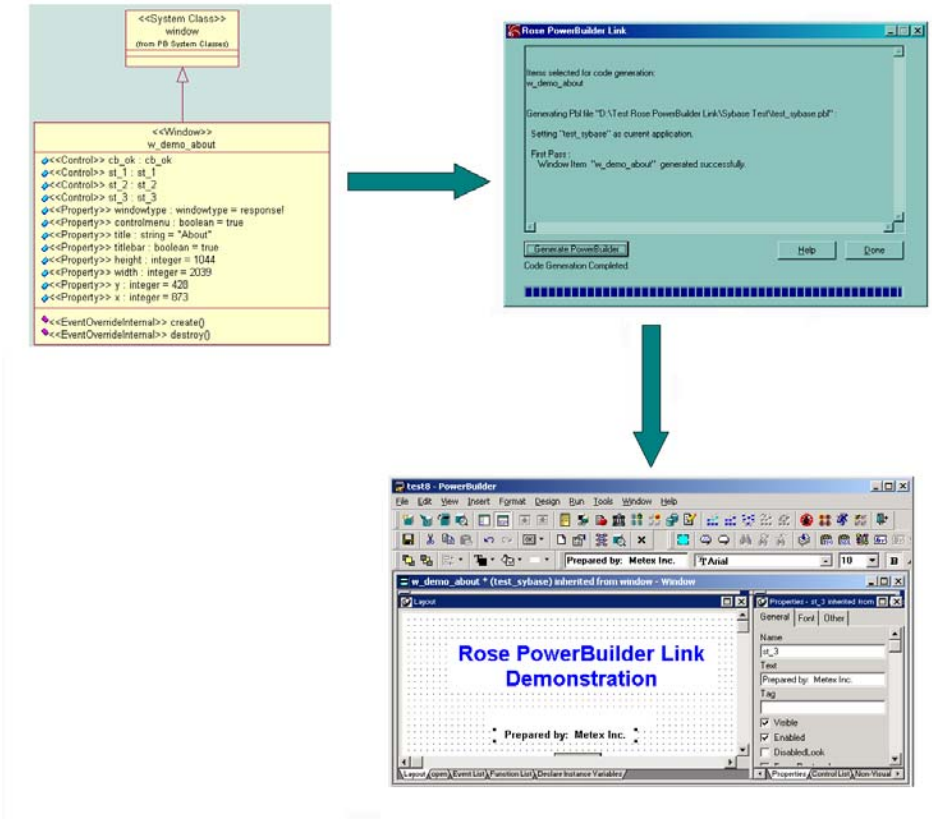

Figure 7: Using Rose PowerBuilder Link to Generate the PB object from the class definition

This functionality allows you to create the framework for your development team instantly and to ensure that your system is built the way it was design to be built.

Round-Trip Engineering

These forward engineering and reverse engineering mechanisms in Rose PowerBuilder Link can be used to continuously maintain consistency between an application's model and implementation, facilitating iterative development.

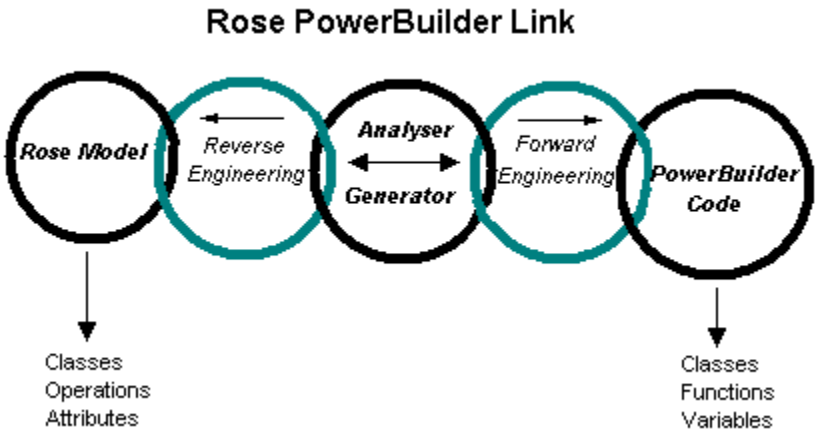

Figure 8: Using Forward and Reverse Engineering Mechanisms for RTE

Our implementation of the Rose PowerBuilder Link synchronizes the implementation with the design allowing developers to develop code according to how designer envision their systems and allowing designers to see developers are building their systems.

Considerations

Knowledge is ultimately the dictating factor of business success. Strategic, tactical, and operational knowledge can be lost when key developers move on, and organizations typically spend much time and money recovering what they already knew, but never took the time to capture.

The Rose PowerBuilder Link is designed to take advantage of the modeling and documentation power of Rational Rose. The Rose PowerBuilder Link is not designed to be a replacement for the PowerBuilder environment and it is not recommended that source code be edited within the Rational Rose environment.

Since Rational Rose is not language specific, there are many implementation specific features that are not captured in the Rational Rose model using our automated tool. For example, the *datawindow* and all its *datawindow controls* are not represented directly in the model after reverse engineering. The *datawindow* is represented with a simple placeholder class to illustrate that it exists within the system.

The UML process consists of many views. The Rose PowerBuilder Link is designed to provide the mechanisms to build and maintain the 'Logical View'.

Summary

Rose PowerBuider Link Provides Many Benefits

Generate PowerBuilder Code From Your Design. Your design will automatically generate the framework for your PowerBuilder application. Minimize application development time by having the framework ready in a top-down approach to software development.

Make Changes to Your Design Easily. Extending Rational Rose functionality to PowerBuilder using Logical Views to understand or design your PowerBuilder systems quickly, and maintain consistency between design and implementation.

Change Control. Use the Round-Trip Engineering capability provided by Rose PowerBuilder Link to have a controlled, iterative style of development that lets projects evolve as project parameters change or new requirements are added.

Reduce Overall Costs. Take advantage of the UML process that has been accepted by the Object Management Group (OMG) as the software engineering standard since November of 1997. Ensure that documentation of your system is thorough and minimize the risk during changes to your system and/or your development team. Having a process that is controlled and predictable will help avoid the costly and time-consuming changes that typically afflict most software projects.

Better Code and Better Value. Minimize risk, maximize control, and develop better code that meets your all your design requirements. Proper management is key to your software success and, in turn, successful software will drive your business and keep your clients satisfied.

References

Web Sites

Web site references:

http://www.rational.com/

http://www.omg.org

http://www.sparxsystems.com.au/

http://www.justpbinfo.com/

Metex Inc.

About Metex Inc.

Founded in 1989, Metex Inc. is a high-performance IT shop of accomplished senior programmers, developers, and project managers. Metex has an enviable record in delivering enterprise solutions and the associated training, service, and support on time, on budget and on specification.

Metex's approach represents the best of breed in project management and software tools development with a focus on object-oriented programming and enterprise solutions. Our expertise includes Java, XML, .NET, UNIX, Centura, Oracle and Rational Rose, plus distributed systems, embedded systems, and real-time applications.

Metex is on the leading edge of creating new approaches through our on-going research and development activities. This leadership has resulted in several award-winning and world-leading applications including the Metex Migrator technology and several leading telecom solutions. Get the Metex Advantage on your next IT development.

Contacting Metex Inc.

Metex Inc. can provide you with further information through:

Electronic Mail: info@metex.com

Phone: 416-203-8388

Fax: 416-203-3095

Phone Toll Free within North America: 1-866-817-813

Author

Reginald Chin, Product Integrations Manager

Electronic Mail: reginaldc@metex.com

Phone: 416-203-8388 ext. 229

Glossary of Terms

Round-Trip Engineering

Round-Trip Engineering (RTE) allows you to make a quick model of an application under development, convert it to application language source code, analyze it, and then refine it based on your increasing understanding of the problem. This process of modeling, code generation, analysis, and refinement is reiterated until you are satisfied with the completeness and correctness of your system.

4GL

4th Generation Languages/Rapid Application Development tools provide greatly reduced software development times, and the opportunity for the non-computer specialist to produce applications to a professional standard.

OMG

The Object Management Group (OMG) is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. Our membership includes virtually every large company in the computer industry, and hundreds of smaller ones. Most of the companies that shape enterprise and Internet computing today are represented in our Board of Directors.

UML

The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components.

Index