ACTL
Advanced CASE Technology & Language Systems

ACTL Systems Ltd.
217/5 Jaffa St. P.O.B. 8129
91081 Jerusalem, ISRAEL
Tel. 972-2-5376459
Fax. 972-2-5370425

Basic Rhapsody Tool Training

3 Days

1. Introduction

Rhapsody is the industry's leading Model-Driven Development environment based on UML 2.1 and OMG SysML™ 1.0 to address the needs of systems engineers, SW Engineers and developers in one fully integrated modeling environment solution. Rhapsody is a Model-Driven Development (MDD) environment for systems and software development of real-time and embedded applications. With its powerful requirements management and industry standard notations to capture the design supported out of the box make it a first class solution for the systems engineering community.

Model-Driven Development (MDD) technology enables you to achieve unparalleled gains in productivity over traditional document driven approaches by enabling you to specify the system design and architecture graphically, simulate and automatically validate the system as you build it, and ultimately produce a quality systems specification that you can be sure is correct, non-ambiguous and completely satisfies original requirements. Based on this specification, SW developers use Rhapsody to generate their applications that filly conform to the specification.

2. Course Description

This course is a hands-on course in which the participants will do most of the work. The training is based on a number of exercises that build on each other. Through the exercises the participants become familiar with all fundamental capabilities of Rhapsody required to design and build SW systems.

The course starts with a simple example that produces the “Hello World” application. The next example builds a simple counter using a single class, a couple of attributes and operations and a simple state chart. Then next step is to explore the advanced capabilities of animation, testing and complex state machines. The last example is used to combine all modeling, code generation and testing capabilities of Rhapsody by building a complete cash register SW.

3. Goals

The primary goal of this course is to get familiar and gather hands on experience with Rhapsody for designing and building SW systems. Specifically you will:

- Learn and experience the modeling capabilities of Rhapsody.
- Learn and experience the code generation and round tripping capabilities.
- Learn and experience the testing and debugging capabilities.
- Learn how to use Rhapsody in a multi user development environment

- Learn the internal of Rhapsody

4. Participants

The course is meant for software engineers, experienced programmers and system analysts who intend using Rhapsody for developing embedded RT SW systems.

5. Pre-requisites

It is assumed that you have a good understanding of object oriented analysis and design using UML, as well as of C++.

Course Outline

Day 1:

During the first day, the fundamental capabilities of Rhapsody are presented and exercised.

Introduction

- The Rhapsody environment – Diagram pane – Browser.

Hello World example

- The class diagram – Components – Creating instances – Code generation – Project files.

Count down example

- Attributes – Operations - State chart – Animation – Sequence diagrams.

Dishwasher example

- Associations - Composite state charts – Event generation and event parameters

Day 2:

During the second day you will see how all modeling and testing capabilities combined lead to efficient system development.

Cash register application

- Use cases – Requirements – Gateway.
- Class diagram – Interfaces – Components - Ports – Composite structure diagrams.
- Active classes – OMThread – Aggregation – generalization

Day 3:

During the third day the exercises of the previous day are completed and some more advanced subjects of Rhapsody are tackled:

Cash register application

- Mutators – Accessors –
- Animation – Simulated time model – Webify.

Advanced topics

- The Rhapsody Framework
- Configuration Management
- Containers
- Test Conductor
- Reporting

As the training is based on hands-on exercises, each participant must have a work station on which Rhapsody with an active license and a supported compiler (best if Visual C++ V.6.0 or Visual Studio .Net) is installed.

Detailed Course contents

Introduction

- Pre-requisites
- Basic Rhapsody
- Case study
- Advanced Rhapsody
- Specialist Rhapsody

Part One: Rhapsody Basics

The goal of this section is to understand how to install Rhapsody and how it interacts with the visual studio environment.

1. Creating Hello World Project

The goal of this section is to master creating classes using the class diagram, generating and modifying the generated code and creating the initial instances via the configuration and component.

- The browser
- Class diagram
- Adding a modifiable constructor
- Creating a Component
- Initial instances
- Selecting an environment
- Code generation
- Generated files
- Editing the generated code
- Display options on the OMD
- Displaying the make and main files
- Project files

2. Countdown Project

The goal of this section is to add implementation based on the Rhapsody framework and to animate the browser and a statechart.

- Adding attribute and operations
- Attribute visibility
- Operation implementation
- Active code view
- A simple Statechart
- The OXF timer mechanism

- Generated code
- Animation
- Animation tool bar and browser
- Drawing a sequence diagram
- Project properties and customization

3. Dishwasher Project

- Dishwasher Statechart
- Concurrent states
- History and diagram connectors
- Adding the transitions
- Debugging at the design level
- Creating and running a script
- Generating events
- A directed association
- Code for a relation
- Adding and accessing event parameters
- Modifying the main to initialize initial object relations

Part Two: Using Rhapsody for real projects

4. CashRegister Project

The goal of this section is to create a mini project based on the steps of the Rapid Object Oriented Process for Embedded Systems, and to see how the Rhapsody framework implements the various advanced UML artifacts.

- Use Cases and actors
- Descriptions
- Relation between Use Cases
- Analysis package
- Navigating the project
- Adding a public constructor
- Collections for an association with multiplicity
- An ordered and qualified association
- Auto generated operations for collections
- Composite class
- Viewing the objects and relations in the animated browser
- Sequence diagrams for the various scenarios
- Changing a message to an event
- Including iostream.h to the Display class
- Adding actions to the transitions

- Automatically generated sequence diagrams
- Injecting events
- Automating the tests
- Substatecharts
- Active classes
- The execute() operation of OMThread
- Thread properties
- Setting the animation focus on a specific thread
- Adding breakpoints
- Referenced sequence diagram
- Generalization
- Directed aggregation
- Abstract operations
- Optimization: get() and set()
- Generating accessors
- Optimization: avoiding new
- Embedded classes

Part Three: Advanced Topics

The goal of this section is to gain a greater understanding of the underlying philosophy and implementation of the Rhapsody framework so that we can better tailor it to our specific project and company needs.

5. Overview on The Rhapsody Framework

- The Object eXecution Framework
- Event driven framework
- OMReactive, OMThread and container classes
- Operating system adapter layer
- Triggered operations and sending events
- Using the STL

6. Introduction to Configuration management

The goal of this section is to learn how the Rhapsody project division can easily integrate into our general project plan and configuration management scheme.

- CM Units
- Command line interface
- Using the SCC interface
- Admin and archive operations
- User defined CM operations using the site.prp file
- DiffMerge tool and Comparing diagrams

- Working with large projects
- Loading Units
- Useful Tips