UML 2.0 Update with Rational Software Architect

The Unified Modeling Language has become the de facto international standard for modeling and specifying software systems. The recent upgrade of the standard from version 1.5 to version 2.0 adds several new diagram types: composite structure diagrams, timing diagrams, and interaction overview diagrams. UML 2.0 also significantly enhances class diagrams, sequence diagrams, state machine diagrams, and activity diagrams. With these enhancements UML robustly supports component-based development, architectural specifications, and advanced behavioral modeling techniques essential to today's complex, distributed systems. This course updates the experienced modeler's knowledge of UML syntax and semantics by highlighting the key UML 2.0 enhancements. The features of the Rational Software Architect (RSA) tool used to model the UML 2.0 enhancements are demonstrated. RSA is used in hands-on labs so the student is immediately able to turn their new knowledge into practical skills.

Objectives:

- Learn the significant UML 2.0 enhancements to use case, activity, class, sequence, and state machine diagrams
- Learn the syntax and semantics of new UML 2.0 diagram types: composite structure, interaction, and timing diagrams
- Demonstrate the use of the RSA tool on traditional and new diagram types
- Gain hands-on experience in key UML 2.0 enhancements using the RSA tool

Audience:

This course is designed for the systems analysts, architects, designers, developers, and testers who are directly responsible for developing object-oriented systems. This course is also of benefit to technical leads and software quality assurance personnel who oversee development of object-oriented systems and require an understanding of the artifacts being produced.

Prerequisites:

Prior UML modeling and software development experience.

Duration:

3 days
Outline:

1. **Introduction**
   - Overview of UML 1.5 diagrams
   - UML 2.0 goals
   - Overview of UML 2.0 enhancements

2. **Use Case diagram enhancements**
   - Review of UML 1.5 syntax
   - UML 2.0 enhancements
     - Association multiplicities
     - Extension conditions

3. **Activity diagram enhancements**
   - Review of UML 1.5 syntax
   - UML 2.0 enhancements
     - Accept event
     - Accept time event
     - Send signal
     - Activity edges
     - Activity internals: input & output pins, parameters, pre & post conditions
     - Multi-dimensional & hierarchical Partitions
     - Expansion regions
     - Exception handlers
   - Demo: Instructor shows RSA features
   - Lab: Develop an enhanced activity diagram

4. **Class & Component diagram enhancements**
   - Review of UML 1.5 syntax
   - UML 2.0 enhancements
     - Parts
     - Ports
     - Connectors
     - Navigable ends
     - Components redefined
     - Provided interfaces
     - Required interfaces
   - Demo: Instructor shows RSA features
   - Lab: Develop an enhanced state machine diagram

5. **Sequence diagram enhancements**
   - Review of UML 1.5 syntax
   - UML 2.0 enhancements
     - Loop fragments
     - Alternative (alt), option (opt), break & parallel (par) fragments and continuations
     - Interaction use (ref) fragments, decomposed lifelines & gates
     - Critical region fragments
     - Negative (neg), assertion (assert), strict, ignore, and consider interaction operators
     - Part decomposition
   - Demo: Instructor shows RSA features
   - Lab: Develop an enhanced sequence diagram

6. **State Machine diagram enhancements**
   - Review of UML 1.5 syntax
   - UML 2.0 enhancements
     - Submachine states
     - Submachine & composite state entry & exit points
     - Terminate pseudo state
     - State machine generalization & specialization through extension
     - Protocol state machines
   - Demo: Instructor shows RSA features
   - Lab: Develop an enhanced state machine diagram

7. **Intro to Composite Structure diagrams**
   - Diagram intent
   - Syntax and semantics:
     - Parts, ports, & connectors
     - Collaborations
   - Demo: Instructor shows RSA features
8. **Intro to Interaction Overview diagrams**
   - Diagram intent
   - Syntax and semantics:
     - Reuse of Activity diagram syntax
     - Inline interactions
     - Interaction use references
   - Demo: Instructor shows RSA features

9. **Intro to Timing diagrams**
   - Diagram intent
   - Syntax and semantics:
     - State or condition timelines
     - General value lifelines
     - States, conditions, messages, & events
     - Timing rulers, time observations & constraints, duration constraints & observations
   - Demo: Instructor shows RSA features
Why IconATG?

- Thirteen years of experience mentoring, consulting, and developing training programs for large IT organizations
- Full software lifecycle curriculum of cost-effective, tailored courses
- Seasoned instructors qualified through hands-on experience
- Experienced mentors and consultants with demonstrated project success
- Proven experience tailoring and extending the Unified Process and Agile/Scrum Processes

IconATG is a thought-leader in information technology training, mentoring and consulting. Our training staff has successfully developed cost-effective, customized IT training programs and we have taught thousands of students through our formal courseware and hands-on workshops. We offer introductory to advanced courses in focused disciplines of the full software lifecycle including the Unified Process (RUP), Agile, UML, requirements and use cases, usability, project management and architecture (SOA/MDA). Our instructors' real-world expertise is incorporated in each of our classes, giving your team the practical skills to be more productive when developing today's most demanding applications.

Our mentors and consultants have worked with project teams to apply new technologies and processes in their organizations to ensure project success. Full lifecycle experience allows IconATG consultants to deliver expert knowledge in specific disciplines while providing an understanding of the workflow throughout the lifecycle. Mentors and consultants actively work with the project team helping them develop skills and address problems through facilitation, demonstration, co-development, review, observation and advice. Mentoring solidifies knowledge gained through training by applying the concepts learned in class. Icon's extensive project experience has shown that teams better understand new processes and techniques by applying them with a seasoned mentor. IconATG is that critical resource; we can help ensure your success.