

DEMONSTRATIONS

DEMONSTRATIONS

Chair: Peter Kriens,
aQute B.V.

Demonstrations provide an opportunity to describe works-in-progress, to display applications of object-oriented technology, and to share unique and interesting technical aspects of object-oriented products, tools, or systems. Demonstrations are intended to be two-way interactions between the authors presenting state-of-the-art technology, and the audience, which has the opportunity to share ideas, interact with the authors in a small scale venue, and learn techniques used in developing innovative and high quality software. This year we have a selection of 20 demonstrations that cover a broad range from XML to scrolling, from nuclear claims systems to personal information agents.

Demonstration sessions last for 45 minutes. Attendance at demonstrations is on a first come, space available basis until the session capacity is reached. In perusing the schedule, please note that all demonstrations are offered both on Tuesday the 16th and Wednesday the 17th; this should make your conference scheduling plans easier.

Demonstrations take place in rooms 18, 19, 20, and 21, in the Tampa Convention Center.

Tuesday and Wednesday – 10:30 am - 11:15 am

Convention Ctr — Room 18

1 Developing and Evolving Java Applications using Coordination Contracts

João Gouveia, *Oblog Software SA*
 Georgios Koutsoukos, *Oblog Software SA*
 Michel Wermelinger, *Universidade Nova de Lisboa*
 José Luiz Fiadeiro, *Universidade de Lisboa*
 Luís Andrade, *ATX Software SA*

Coordination contracts are a modeling primitive to facilitate the evolution of software systems. Coordination contracts encourage the separation of computation from coordination aspects. They encapsulate the way components interact, and capture the business rules or other volatile rules that govern the application. System evolution consists of adding and removing contracts (i.e., changing the rules) between components. The interactions specified by the contract are superposed on the functionality provided by the components, without the programmer having to modify them and without components being aware of contracts. New contracts may be added anytime (even runtime).

This demonstration presents our Coordination Development Environment (CDE), written in Java, that supports the development of Java applications with coordination contracts. It allows to write contracts, and to register Java classes (components) for coordination. The tool is then used to generate the Java code for adapting those components and for implementing the contract semantics based on a micro-architecture. The CDE also includes an animation tool, in which the runtime behavior of contracts and their participants can be observed using sequence diagrams, thus allowing testing of the deployed application.

Convention Ctr — Room 19

2 Information Integration and Visualization (IIV) with Java and XML

Kurt Derr, *Idaho National Engineering and Environmental Laboratory*
 Kenneth Rohde, *Idaho National Engineering and Environmental Laboratory*

Information systems are typically tailored to specific domains, such as management, financial, geographic, project management, inventory control, military planning, etc. This makes it difficult for a single application to view related information from disparate sources. For example, information from different applications can only be viewed with that application. IIV solves the problem of viewing related information from disparate sources. It stitches together islands of information at the level of the desktop, with no user programming required. An extensive repertoire of visualizations and data manipulation tools is available. IVBQ is a client-server application written in Java.

This demonstration will show the integration and visualization of related information from multiple sources - XML, live network connections, relational databases, files, and GIS sources.

DEMONSTRATIONS

Tuesday and Wednesday – 10:30 am - 11:15 am (cont.)

Convention Ctr — Room 20

3 Round-trip Objects in Smalltalk

Daniel Antion, *American Nuclear Insurers*

The demonstrated system was developed by American Nuclear Insurers to deploy an emergency claims system in response to emergency situations at commercial nuclear power plants. A single host is responsible for claim validation and persistence. Remote nodes can be individual users or host computers. The system communicates by sending Smalltalk objects over TCP/IP in a host/remote mode, mixed mode (some remote hosts and some remote users) or in a multi-tier configuration. Smalltalk's support for creating and manipulating object based transactions and the availability of a socket framework was a key reason for selecting VisualAge. Also important was the high degree of support for object reuse and the ease of testing during development.

The demo includes a discussion of the transaction processing, the underlying communication framework (built on top of TotallyObject's - Socket Set), the reasons for building the system in Smalltalk, the testing methods and the test results.

Convention Ctr — Room 21

4 Asset Locator - an Enterprise Development Resource Management System

Avi Yaeli, *IBM Haifa Research Staff Member*

Gabi Zodik, *Manager, Programming Languages and Environments*

Dept., *IBM Haifa Research Lab*

Iftach Ragoler, *IBM Haifa Research Staff Member*

Alex Akilov, *IBM Haifa Research Staff Member*

Enterprise Development Resource Management (EDRM) is a new paradigm for the management of development resources (e.g. code, documentation, design documents, test suites) in the enterprise. EDRM leverages the valuable information that exists in development resources and systems to provide developers with search, reuse, collaboration and impact analysis capabilities. Asset Locator - a low cost/maintenance scalable solution is a first step towards achieving EDRM. Asset Locator uses a set of autonomous crawlers to crawl into the enterprise repositories/systems and discover new development resources. A set of domain specific analyzers (e.g., Java, C++, HTML, etc.) analyze the discovered resources and then populate a central repository. This repository serves as the base for global analysis (e.g. impact analysis) as well as the rest of the EDRM capabilities that Asset Locator provides.

This demonstration will show how to exploit the textual information and semantic features that reside in OO resources (e.g. inheritance relationship in Java) to support automatic categorization of OO resources into predefined domain taxonomies, semantic navigation of the enterprise repository and improved component/code reuse via a semantic search server.

Tuesday and Wednesday – 11:15 am - 12:00 pm

Convention Ctr — Room 18

5 ComposeJ: Aspect-Oriented Programming with Composition Filters and Java

Lodewijk Bergmans, *University of Twente, The Netherlands*

Mehmet Aksit, *University of Twente, The Netherlands*

Richard van de Stadt, *University of Twente, The Netherlands*

Hans Wichman, *TRIMM bv, The Netherlands*

Patricio Salinas, *University of Twente, The Netherlands*

Luigi Savarese, *University of Twente, The Netherlands*

Composition Filters address a number of modeling obstacles that conventional object-oriented techniques cannot address or can only solve with poorly maintainable designs. Aspect-oriented programming (AOP) deals with features crosscutting multiple methods and/or classes. These features typically results in tangled code that mixes and/or duplicates code that deals with different issues. The Composition Filters approach to AOP integrates aspects and classes, retains strong encapsulation, and supports composability. ComposeJ is a tool that takes Java classes and (separate) composition filters specifications, and transforms the Java classes by inlining so that they implement the behavior as specified in the composition filters specification.

The demonstration will introduce a modeling problem involving crosscutting behavior, and explain why it cannot be implemented in Java effectively. A solution will be shown by extending Java classes with a composition filters specification. The GUI of ComposeJ is used to generate Java files that implement the composition filters specification. The generated code will show that this code is a tangled implementation of the crosscutting behavior.

Convention Ctr — Room 19

6 AspectJ™: the Language and Support Tools

Erik Hilsdale, *Xerox PARC*

Gregor Kiczales, *University of British Columbia*

Aspect-oriented programming (AOP) is a technique for improving separation of concerns in software design and implementation. AOP works by providing explicit mechanisms for capturing the structure of crosscutting concerns. When implemented in a non-aspect-oriented fashion, the code for these concerns typically becomes spread out across entire programs. AspectJ controls such code-tangling and makes the underlying concerns more apparent, making programs easier to develop and maintain. AspectJ is a seamless aspect-oriented extension to Java™. It can be used to cleanly modularize the crosscutting structure of concerns such as exception handling, multi-object protocols, synchronization, performance optimizations, and resource sharing.

This demonstration illustrates what the AspectJ language can do and it shows the tools that support developing programs with this language. AspectJ is freely available at <http://www.aspectj.org>

DEMONSTRATIONS

Tuesday and Wednesday – 11:15 am - 12:00 pm (cont.)

Convention Ctr — Room 20

7 Comparing Two Approaches for Mapping Objects to Relational Databases

Arash Farzin, *Nebras Informatics Co.*

Navid Khosravi, *Nebras Informatics Co.*

Sasan Dashtinezhad, *Nebras Informatics Co.*

Nebras Persistence Service is a product that can be used to add persistency to object-oriented systems. It provides a tool that automatically adds the needed persistency methods and attributes to persistent classes in Rational Rose models. A run-time service facilitates the persistency between the Application Layer and Storage Layer. The persistency is achieved by mapping attributes to relational database tables. Different strategies can be followed with this mapping.

The demonstration will show the two mapping approaches and they will be compared for their benefits and drawbacks in different circumstances.

Convention Ctr — Room 21

8 GME: A Reflective Environment for Domain-Specific Modeling

Akos Ledeczki, *Institute for Software Integrated Systems, Vanderbilt University*

The Generic Modeling Environment (GME 2000) is a metaprogrammable graphical editor supporting the design, analysis and synthesis of complex, software-intensive systems in diverse engineering fields. GME 2000 has a component based architecture using MS COM technology and is implemented in C++. The Core component exposes the domain-specific language specifications, provides access to the models and publishes its modification events. All the other components (GUI, browser, OCL constraint manager, software generators, etc.) are built around the Core. Model persistence is supported via standard database technology and XML import/export functionality.

The demonstration will focus on an example domain, an integrated simulation framework for embedded systems. The UML and OCL based metamodels specifying the domain-specific visual modeling language will be shown. From these metamodels, the domain-specific environment is generated automatically. An example application will be demonstrated; an Automatic Target Recognition system, including its complex models and the automatically generated Matlab simulation and C implementation.

Lunch break

Tuesday and Wednesday – 1:30 pm - 2:15 pm

Convention Ctr — Room 18

9 NetBeans Metrics Module

Thomas Ball, *SunLabs, Sun Microsystems, Inc.*

Software metrics are often used by project managers to help understand a project's design, complexity, and robustness. Unfortunately, much of this information is fed back to the project's engineers when it is too late in a release cycle to be of much use.

This demonstration shows a NetBeans (Forte for Java) module that adds software metrics properties to NetBeans' Explorer. This allows metrics to be easily monitored while classes are designed and developed. Sun Labs is investigating whether this early feedback will improve Java applications and reduce their development and maintenance costs.

Convention Ctr — Room 19

10 Tengger Support of Design-through-Code [UML through Java] for Feature-Oriented Separation of Concerns

William Harrison, *IBM T.J. Watson Research*

Charles Barton, *IBM T.J. Watson Research*

Mukund Raghavachari, *IBM T.J. Watson Research*

Tools for composition-based software building, whether based on Aspect-Oriented Programming or more general approaches to Advanced Separation of Concerns, are generally focused on the weaving or composition of code in OO languages. Tengger is a tool that supports the expression of the design as separated concerns, generating Java code from sets of UML class diagrams. Each concern is expressed by its own UML class diagrams and by Java implementations of the UML-declared methods. The UML design is at a high-level, not specifically tailored to Java's language constructs. Tengger composes the UML designs and produces Java code implementing the result. Hyper/J composes the generated Java code with the developer's code to produce a complete application.

This demonstration will present the concepts behind feature-based design and development as supported by Tengger. A small application is carried through the design using Rational Rose, the code-generation using Tengger, and the code composition with Hyper/J. The focus will be on the design-through-code process, on the Java coding style used in the developer-written code, and on the ways in which Tengger and Hyper/J fit into the process to yield the final application.

DEMONSTRATIONS

Tuesday and Wednesday – 1:30 pm - 2:15 pm (cont.)

Convention Ctr — Room 20

11 Explicit Programming: Improving the Design Vocabulary of your Program

Avi Bryant, *University of British Columbia*

Andrew Catton, *University of British Columbia*

Kris De Volder, *University of British Columbia*

Gail Murphy, *University of British Columbia*

Object-oriented systems are frequently built around conventions and design patterns that can only be indirectly captured in source code. Explicit programming provides a solution to extend the vocabulary of existing, general purpose languages, allowing developers to encode design decisions in a concrete, encapsulated, and reusable way.

ELIDE supports explicit programming in Java and allows developers to introduce new, parameterized modifiers into the Java language, at the class, field, method, and block levels. Each of these modifiers triggers a series of transformations on the program source code. The transformations for a modifier are defined by the developer in a Java class, which operates on a simple representation of the original source code.

The demonstration will show how to use the tool to greatly simplify the usage of existing frameworks such as JUnit and standards such as JavaBeans, by introducing test-specific and bean-specific modifiers.; ELIDE will be used to refactor repetitive code and design structures into custom transformations through lightweight, iterative application of the tool.

Convention Ctr — Room 21

12 Demonstration of a UML-Based Architecture Description Language

David Boyle, *ALLTEL Communications Inc.*

Sandeep Gupta, *ALLTEL Information Services*

Floyd Berus, *ALLTEL Information Services*

Charles Letner, *ALLTEL Information Services*

The notations and tools in widespread use for the purpose of designing and documenting enterprise IT architecture focus primarily on application architecture. As a result they do not adequately provide for the technical management of an enterprise comprised of many systems. An Architecture Description Language (ADL) provides a mechanism for modeling enterprises from a logical level, through system function, integration, hosting, and networking. The goal is to provide an integrated set of models that take the viewer from a business process down to the specific devices on which the software resides that addresses the business concern. This ADL notation is separated into Views and Layers. These Views and Layers represent various abstractions of both the business and the technical domains.

This demonstration will show ADL and how to extend two of the popular UML-based modeling tools and create a CASE environment that supports enterprise architecture design.

Tuesday and Wednesday – 2:15 - 3:00 pm

Convention Ctr — Room 18

13 Integrating Refactoring Support into a Java IDE

Dirk Baeumer, *Object Technology International*

Erich Gamma, *Object Technology International*

Adam Kiezun, *Object Technology International*

Most existing refactoring tools are add-ons to existing Java development environments. The Eclipse Java tooling is a new open source development environment that comes with integrated refactoring support for Java. Refactoring support is built on top of an infrastructure that was designed with a focus on refactoring. A database containing structural information about Java projects provides means for efficient searching and reference updates. Refactoring is seamlessly integrated into the UI and available from all views. And last, the environment is open for developers to extend the set of available refactorings.

The demonstration shows various refactorings and their integration into the Eclipse Java Tooling. An in-depth overview of the technical aspects of the refactoring tool is given by presenting the implementation of a particular refactoring.

Convention Ctr — Room 19

14 Web Testing Using an Object-Relational Database System

Robert Cook, *Yamacraw Professor, Dept. Math and CS, Georgia Southern University*

The demonstration illustrates how the XDB object-relational database system implements a web student testing server. An example web site is <http://bcook.cs.gasou.edu/eclass>. A unique aspect of the testing server is its extensibility and the test format, which is executable. As a result, question parameters and answers are computed on-the-fly. The Answer-Vector (AV) database object supports the flexible grading of tests, which might have a wide variety of question types. The AV encodes each question's format (multiple choice, numeric answer, etc.) and a possible list of answers in format in which answers are graded by performing a case-insensitive comparison to a list containing the correct answer and some common misspellings of the correct answer.

The demonstration will include the dynamic generation of computed tests as well as a review of different question and answer specifications. The goal is to foster a discussion of the applicability of computer-based testing to a problem domain that ranges from true/false questions to essays.

DEMONSTRATIONS

Tuesday and Wednesday – 2:15 - 3:00 pm (cont.)

Convention Ctr — Room 20

15 Traversing the `SemiPositionableStream`: Millions of Objects in a Scrolling List

James Foster, *Systems Architect, Banner Health System*

A scrolling list is generally impractical for displaying more than a few hundred items, particularly if the items are in a multi-user database. While databases typically provide an interface to access sorted records as a stream (e.g., with a cursor), they generally do not provide: (1) the information needed to display the scroll bar (or thumb) that identifies the relative position in the list; (2) an interface to jump to a relative position in the list; or (3) an interface to move backwards. This demonstration will discuss `SemiPositionableStream`, an extension to Brokat's GemStone/S Object Server that identifies an approximate position in a list, allows approximate positioning, and moves both forward and backward. By filling the visible portion of the scrolling list and adjusting the scroll bar to reflect the position of those items in the underlying list, the scrolling list can appear to present any number of items. The database techniques described will be applicable to non-user-interface situations and to other database implementations as well.

Convention Ctr — Room 21

16 An OO Client Server System for Concept Mapping and Lexical Navigation

James Cooper, *IBM T J Watson Research Center, jwcnmr@watson.ibm.com*

The purpose of the system is the discovery of related concepts that might never have been obvious from reading any single group of articles. The system can be used for query refinement and focus of overly general queries. The system works as follows. A server manages a search engine and index, document text mining and a database which holds document metadata, including linguistic relations between discovered multiword terms. A client program displays "concept maps," or lexical relations between terms using a graphical layout system. This program is based on JSP and Java and communicates to the server with SOAP over HTTP.

The system will be demonstrated on a set of prescription drug data, showing what can be learned by mining these documents.

Tuesday and Wednesday – 3:30 pm - 4:15 pm

Convention Ctr — Room 18

17 ArchJava: Connecting Software Architecture to Implementation

Jonathan Aldrich, *University of Washington*

Craig Chambers, *University of Washington*

David Notkin, *University of Washington*

ArchJava is a small extension of Java that integrates software architecture smoothly into Java code. Software architecture describes the structure of a system, and is useful for design, program understanding, and formal analysis. However, existing systems' implementations are developed separately from their architecture and therefore may not conform to that architecture, causing confusion, violating architectural properties, and inhibiting software evolution. ArchJava is unique in that it keeps code and architecture coherent without restricting common programming idioms.

This demonstration will show how to apply ArchJava to create both static and dynamically changing software architectures. It will demonstrate the downloadable ArchJava compiler and tools, and explain how ArchJava's type system keeps implementation code and architecture coherent. ArchJava will be applied to two moderate-size (12-15,000 line) applications. The demo will include a brief comparison to related technologies and will actively search feedback from the audience.

Convention Ctr — Room 19

18 CommonRules : A Framework For Rapid Development of Object Oriented Applications Using Business Logic Modules.

Benjamin Grosf, *Sloan School of Management, MIT*

Jeff Kephart, *IBM T.J. Watson Research Center*

Even with modern OO languages like Java and C++, it still requires skilled programmers to create an application. Java beans technology provides a way to connect objects together but is limited in expressing the logical relationships. Modifications and maintenance will also be a time consuming and difficult, especially when original programmers are not around. CommonRules introduces a platform and methodology allowing rapid application building based on small business logic modules expressed as sets of logic based, human language like rules. Larger applications can be constructed by merging small modules of business logic dynamically. The framework completely hides the programming details and exposes only the business logic as human language like syntax.

The demonstration will show how to build an application using the CommonRules framework and modify the application subsequently to illustrate the various features mentioned above. It will also show how to construct an application by merging small pieces of business logic modules dynamically and resolve conflicts among the small business logic modules.

DEMONSTRATIONS

Tuesday and Wednesday – 3:30 pm - 4:15 pm (cont.)

Convention Ctr — Room 20

19 Supporting Distributed Extreme Programming

Frank Maurer, *Department of Computer Science, University of Calgary*
Sebastien Martel, *Department of Computer Science, University of Calgary*

Extreme programming (XP) is arguably improving the productivity of small, co-located software development teams. However, there is the XP constraint of co-location. This can be overcome by introducing a process-support environment (called MILOS) that helps software development teams to maintain XP practices in a distributed setting. MILOS supports project coordination, process support, object versioning, skill model, information routing, team communication, pair programming, resource pools, long term organizational learning and more. To achieve this, MILOS was build using EJB and various other OO components.

The demonstration will consist of an example scenario of simplified extreme programming development cycle using the MILOS system.

Convention Ctr — Room 21

20 Personal Information Agent

Dominik Kuroпка, *University of Muenster (Germany)*

Information overflow is one of the greatest challenges for information focused professions today. Personal Information Agent (PI-Agent) is an adaptive agent-based information filtering system which automatically scans news-sources for presumably important messages. Different types of information sources are possible (e.g. web-pages, e-mail, newsgroups). PI-Agent works for one user and has ideas about what is important for its user. The user past evaluations are used for learning. The PI-Agent prototype is implemented in Java. Persistency of agents knowledge base is assured by a relational database system (PostgreSQL). For communication with the user a web-based user interface is implemented using an Apache webserver and Java-Servlets.

The audience will experience how the agent works in cooperation with its user. Furthermore the system architecture, the functionality and the implementation details are presented.

NOTES: