
Component Development 99 Conference Trip Report

August 2 – 5, 1999
New York City, New York

Attended by Ly Sauer

Both academia and industries have dubbed component-based development as the next major movement in application development. The motion is the potential that components can significantly improve productivity and speed the development process while producing more flexible and quality applications. The Gartner Group asserts that by 2001 at least 60% of all new applications developed will be based on assemblies of components. Component Development provides the information necessary to help developers confidently chart your course for the effective implementation of component development in large-scale systems.

The conference consisted of four separate tracks to cover COM/DCOM, CORBA, Enterprise JavaBeans, and “how to” build and manage components (see the end of this document for the conference syllabus). What I concluded from spending four days following the CORBA and component construction/management tracks are:

- The software community is still trying to define a component. It is clear that components are more than objects and are an independent encapsulation of a collection of “things” that corporations reuse, its interface is context specific, and is composable. Some attributes of components include no inheritance; does not have state; supports security, licensing, versioning, life cycle management, event notification, introspection, concurrent control and locking, configuration and management, persistence, and transaction; plug and play; interoperable; portable; works with existing legacy applications; self managing, and self installable.
- While the software community is diligently working to define a component definition, there is still a major force converging on components as the mechanism to support reuse. From this perspective, a component is defined to be an encapsulation of whatever is reusable for the enterprise.
- For an enterprise to provide software “faster, better, and cheaper” requires taking full advantage of reuse at the enterprise level. This calls for more than just software reuse; it needs a reuse infrastructure, which must include architecture (business, target, and constraint), human factor, and methodology (process, notation, and tools).
- Component technologies are still not mature enough for large scale deployment. A good component architecture should not depend on these technologies.
- There is a company which architected, designed, implemented, and deployed an exact subset of the ASIA specification in less than one year with three people. Their architecture implemented the Component Adapter, Component Constructor, and Component Generator of ASIA. This demonstrated that the ASIA is realizable within a reasonable period of time.

In conclusion, the conference provided me with some insights into where industries are with respect to component-based software development as well as getting a good verification of the realization feasibility of the ASIA specification. More details of the sessions offered at the conference and the ones I attended are described below.

The Component Cookbook (Part 1) (8/2)

The objective of this session is to provide a framework (Cookbook) for methodical stepwise execution of a project from genesis to delivery using object paradigm and components. In particular, the session covered the fundamental concepts and terminology of component and object-oriented software development and the ingredient and mixing instructions of the Cookbook framework.

Using Components, Patterns, and Frameworks to Realize Architecture (8/2)

This session provided a clarification and indicated how components, patterns, and frameworks should be utilized to establish an industrial strength architecture that supports the total needs of the business. The tutorial also examined

how to plan and attain effective reuse by combining class libraries, patterns, frameworks, domain specific pattern languages, and the corporate infrastructure necessary for enabling large-scale reuse.

Successful Components (8/2)

This tutorial explored the various possible characteristics of a component to provide a better understanding of what is a component. The characteristics examined include small vs. large grained, technology-based, services, frameworks, patterns, Jini, and Agents.

OMG: Software Integration Through OT (8/3)

Dr. Jon Siegel provided an overview of CORBA and its elements in preparation for the CORBA 3 session.

What's New in CORBA 3 (8/3)

In this session Dr. Siegel provided a fundamental understanding of the features that will be added to CORBA 3. These features include:

- Improved Java and Internet Integration (Java-to-IDL Mapping, Firewall Specification, and CORBA Object URLs)
- Quality of Service Control (Asynchronous Invocation/Messaging, Invocation QoS Control, and Realtime, Minimum, and Fault Tolerant CORBA)
- CORBA Component Model (Objects Pass-by-Value, Component Container, Distribution Format, and Scripting Language Specification)

The CORBA Component Specification (8/3)

This presentation covered the CORBA components specification and explored its impact on CORBA architectures in the future.

Enterprise Productivity Breakthrough in Components (8/4)

Dr. Vayda discussed how organizations improved ROI through technical foundations which result in large scale reuse. He focused on effective approaches to development process and design, distributed component architecture, component frameworks and useful metrics.

Building a Reuse Infrastructure (8/4)

This session presented the bigger picture of how to implement an infrastructure that facilitates and supports an ongoing reuse effort. The infrastructure consists of an integrated tool set, a change management process, a reuse process, and most importantly, a rapid application development process. All parts of the infrastructure work together to generate and maintain enterprise-worthy components within the context of the organization's culture. The approach presented is discussed in the June 99 issue of Component Strategies (http://archive.componentmag.com/9906/html/from_pages/feature.shtml).

Distributed Object Architectures with XML (8/4)

This session explored ways XML can be used in distributed object architectures.

Java in XML (8/4)

This introductory lecture described the current technologies, toolkits, and API's available for combining XML and Java.

Componentizing Legacy Applications (8/5)

This session discussed a component architecture for integrating legacy applications as well as provided an introduction into the various techniques used, technologies chosen, and the advantages and disadvantages of this architecture.

Testing Component Systems (8/5)

This session presented the necessary steps to ensure a successful project, from planning for testability to testing harnesses to using scripting languages.

Component Development 99 Syllabus

The table below lists the sessions offered at the conference. The bolded sessions are the ones I attended.

The Component Cookbook (Part 1)	Trusted Components	Bean Soup: Using Components with Java	Adopting CORBA for Application Developments-Roadmaps, Strategies and Pitfalls
The Component Cookbook (Part 2)	Using Components, Patterns, & Frameworks to Realize Architecture	Getting Started with JFC	Building Transactional Components
How to Prepare Commercial Quality JavaBeans	Successful Components	Building Flexible Components with Design and Analysis Patterns	
Understanding COM Architecture	OMG: Software Integration through OT	Developing Enterprise Solutions with EJB	Performance Analysis and Engineering of Distributed Component-based Systems
Keynote: Jack Bissell, Chief Scientist, Hitachi Software			
Modeling COM with UML	What's New in CORBA3	Building an EJB Component Architecture	Distributed Component-Based Event Monitoring and Debugging
Working with COM Classes	The CORBA Component Specification	Java, JavaBeans, & EJB: The Return of Transaction Processing	Delivering Large Scale Component-Based Software Solutions
Keynote: Thomas P. Vayda, Vayda Enterprise			
Introduction to ATL	Using CORBA, Messaging & XML for Legacy Integration	Using Java, CORBA, and XML for a Distributed Object Architecture	Enterprise Productivity Breakthroughs in Components
Keynote: Allan Vermeulen, Vice President of Development & CTO, RogueWave Software			
COM and DCOM Programming Using ATL	Web Application Server vs. Object Transaction Monitors	Mastering JFC 'Look and Feel' and UI Delegation	Building a Reuse Infrastructure
Understanding and Using MTS	Distributed Object Architectures with XML	Comparing Java Application Servers	Modeling Business Components
Aardvark-A Lightweight Java-Application Architecture	Java in XML	Understanding Infobus Technology	
Using DCOM for Interprocess Communication	Using Server-Side Components with CORBA	Dragging & Dropping Swing Components	Enterprise Component Modeling for Internet-Enabled, Scalable System Developments
DCOM Security	Distributed Internet Application with CORBA	Developing JavaBeans for Various Integrated Development Environments	Componentizing Legacy Applications
DCOM Apartments, Threading and Concurrency	Requirements Engineering	Peer-to-Peer Shared Components	Testing Component Systems