RECURSION SOFTWARE, Inc.

CASE STUDY

– The Invensys® Group, Foxboro Business Unit –

Fast Facts

Company

The Invensys Group, Foxboro Business Unit

Industry

Energy

Geography

Global Markets

Challenges

- Proprietary hardware and software platforms lacking in scalability
- Mixture of Windows® and Solaris® platforms

Solution

- Recursion Software's Voyager® *Java Development Platform*

Results

- Easily integrated, highly configurable, modular based architecture
- Preservation of legacy systems
- Standards-based, open architecture enables easy integration of new technologies in Control Systems

COMPANY

Operating in more than 60 countries worldwide, Invensys is a global automation, controls and process solutions Group that empowers companies to increase productivity, reduce waste, remove cost from supply chains, boost profitability and demonstrate measurable improvement in overall performance.

The Foxboro business unit of Invensys is a leading supplier of instruments, systems, and services for industrial process automation. These solutions typically include intelligent field instrumentation, automation systems, and advanced software applications, plus comprehensive engineering, maintenance, and training for industries such as: chemicals, oil and gas, pulp and paper, food, pharmaceuticals, mining and metals, electric utilities, and water and waste treatment.

CHALLENGES

Control System vendors developed proprietary hardware and software with restrictions that limited the Control System functionality to specific brands. For years the only option in Control Systems Management was to purchase hardware and software that were vendor specific and sometimes lacking in scalability, with capabilities that could migrate from one release to the next. Foxboro was using a mixture of Windows and Solaris platforms within their Control Systems Management systems architecture.

By adhering to industry standards, Invensys thought it possible to develop an industry-wide, vendor-neutral solution to Control Systems Management, scaleable to the complexity of any given system.

SOLUTION

SCALABLE, PLATFORM OF INDEPENDENT SOFTWARE MODULES

Utilizing the advanced features of Recursion Software's Voyager® and Voyager Messaging, Foxboro implemented the Refinery Off-Sites Solutions Suite, the first of a generation of standards-based, open industrial systems designed to meet the total measurement, control and real-time information requirements of today's plants.

The systems architecture consists of easily integrated, highly configurable modules that can be scaled for use in any size plant. The software is platform independent and can run on virtually any platform

Through continuous innovation, the Intelligent Automation Series system has maintained its technological leadership over time, enabling users to continuously leverage the latest technologies for improved plant performance, while preserving their investments in system hardware, software, and applications.

GOALS AND OBJECTIVES

- Utilize existing industry standards and third party modules where necessary.
- Develop a suite of highly configurable tools that encompass Order Movement Management (OMM), Tank Inventory System (TIS), Batch Tracking (BTM), Movement Automation System (MAS), and Blend Optimization (BOSS).
- Develop a suite that is platform independent and scalable from a simple PLC driven system or a CDS/FF system from any vendor.
- Support existing technology including Java, OPC connectivity, database systems from Oracle, Informix, and SQL server, SAP and some open source products.

THE SOLUTION

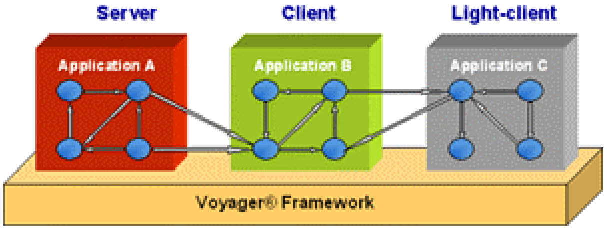
SEAMLESS, EXTENSIBLE, AND DISTRIBUTED COMMUNICATION

Invensys utilized the advanced features of Recursion Software's Voyager® product, a framework for the development of distributed Java™ applications, to create seamless communications between application servers, legacy systems and software clients in a robust, extensible, scalable solution for enterprise distributed development. Based on established ORB technology, Voyager's layered modular architecture transparently supports IIOP, RMI, SOAP, DCOM messaging protocols, RMI Naming, CORBA Naming, JNDI naming protocols, and synchronous, one-way, delayed-synchronous, asynchronous messaging patterns.

OVERVIEW OF VOYAGER®

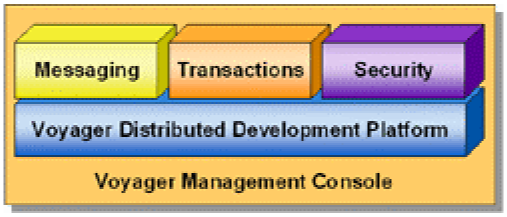
Voyager bridges the communication gap between existing applications built with different object-oriented solutions such as Borland Visibroker, IBM Websphere and Sun/Java. Build new distributed, secure Java applications fast transaction processing and flexible messaging using Voyager, quickly adding new protocols, services and transport layers when needed.

Voyager can be used to develop distributed server, client and light client applications as well as mobile agent based applications.



The Voyager Framework includes:

- Voyager Distributed Development Platform (VDDP)
- Plug-In Modules: Security, Transaction, Messaging
- Voyager Management Console


Messaging
Transactions
Security
Voyager Distributed Development Platform
Voyager Management Console

MESSAGE/EVENT REPLICATION

Voyager uses a different and innovative architecture for message/event replication called Space™ that can scale to global proportions. Clusters of objects in the target group are stored in local groups called subspaces. Subspaces are linked to form a larger logical group called a Space. When a message or event is sent into one of the subspaces, the message or event is cloned to each of the neighboring subspaces before being delivered to every object in the local subspace. This process results in a rapid parallel fanout of the message or event to every object in the Space.
A special mechanism in each subspace ensures that no message or event is accidentally processed more than once, regardless of how the subspaces are linked. Voyager's multicast, distributed events, and publish/subscribe features all use and benefit from the same underlying Space architecture.

REMOTE ENABLING OF CLASSES

Java classes are automatically remote-enabled at runtime. A class does not have to be modified in any way, and no additional files are necessary to remote-enable a class. Thus, there is no difference between a "regular" Java class and a remote-enabled class. Classes may also be explicitly remote-enabled by declaring them to implement either java.rmi.Remote or com.objectspace.voyager.IRemote.

REMOTE OBJECT CONSRUCTION

With a single method call, you can create a remote instance of any class on any Voyager VM and obtain a proxy to the newly created object. The proxy class is generated dynamically if it does not exist, eliminating the need to manually generate this class. Since the proxy implements the same interfaces as the object, by using interface-based programming techniques you do not need to modify any code to work with remote objects.

REMOTE MESSAGING

Method calls to a proxy are transparently forwarded to its object. If the object is in a remote program, the arguments are serialized using the standard Java serialization mechanism and de-serialized at the destination. The morphology of the arguments is maintained. If an object's class implements com.objectspace.voyager.IRemote or java.rmi.Remote, the object is passed by reference. An appropriate proxy class will be generated dynamically if needed. If an object's class implements java.io.Serializable, it will be passed by value. Objects that implement none of these interfaces are passed by reference.

RMI SUPPORT

Voyager provides full RMI support. This means that you can easily use classes in Voyager that were originally designed for use with RMI. Classes designed for RMI may be used in Voyager applications with no modifications.

ADVANCED MESSAGING

You can send one-way, synchronized, and future messages. One-way invocations return to the caller immediately after sending the message; any return value or exception is discarded. Future messages immediately return a placeholder to the result, which may then be polled or read in a blocking fashion.

PUBLISH-SUBSCRIBE

You can publish a Java event on a specified topic to a distributed group of subscribers. The publish-subscribe facility supports server-side filtering and wildcard matching of topics.

RESULTS

The Refinery Off-Sites Solutions Suite consists of modules that are each highly configurable and can be scaled for use in any size plant. The modules integrate seamlessly. The software is platform independent and can be run on virtually any platform.

Voyager is able to automatically reconnect clients after a network failure, without the need of a separate registry process or stub/skeleton generation during development.

“We deploy on almost all Windows platforms, and several versions of Sun Solaris, and Voyager works without noticeable difference on all of them, even in mixed networks. Deployment itself is easy since you only have to package a few libraries,” said Richard Spaninks, Development Manager of Invensys Systems N.V. “Over the years, Voyager is the only software package we have used through several newer versions of our product, multiple updates to the Java language, and several operation systems changes without significant porting or compatibility problems.”

ABOUT RECURSION SOFTWARE, INC.

Recursion Software, Inc. provides software engineers with powerful enterprise development solutions for constructing distributed systems using proven Java™-based methods and emerging Web Services technologies. Recursion Software products allow Java, C# and C++ developers to be productive quickly, by leveraging our expertise in object-oriented techniques and incorporating industry standards. Recursion Software's customer base includes Defense, Financial, Computer Technology, and Telecommunications industries, among others. Recursion Software is headquartered in Frisco, Texas. To learn more about us, visit our website at www.recursionsw.com.

Recursion Software, Inc.
2591 North Dallas Parkway
Suite 200
Frisco, Texas 75034
1.800.727.8674 or 972.731.8800
www.recursionsw.com