

GE Energy

Smallworld Core Spatial Technology™ 4

Interoperability in an open environment



Abstract

From its origins in engineering departments, Geospatial Information Systems (GIS) has emerged to adopt a more important responsibility within the enterprise. This expanded role has resulted in a greater demand for interoperability between existing systems and standards.

Many conventional GIS architectures were designed to support specialized engineering tasks and, as a result have proved to be less flexible to adapting to the new challenges of enterprise-wide applications. Smallworld architects, from GE Energy, have embraced open standards and developed a set of novel base technologies that can form the building blocks of some of the most ambitious integration projects.

Introduction

Integration of many disparate systems is becoming increasingly important in the enterprise. The complexity of many of the systems, the lengthy deployment periods and high replacement costs often mean that systems continue to operate well beyond their intended life span.

However, in reality, many of these systems do not require replacement (the cost of replacement does not justify the benefits). These systems often support business critical operations in the organization and act as a repository for large amounts of valuable data.

From the business point of view, this arrangement is often the result of a realistic cost/benefit analysis, but to the Information Technology team, the integration of these systems often represents a major technical challenge.

As a result, a wide range of technologies have emerged to help with this kind of complex integration. At one level, integration technologies such as CORBA® Object Management Group, Inc. EAI and COM allow legacy systems to interact seamlessly with state of the art systems and also with widely used desktop applications. Standards such ODBC, JDBC, OLE DB and SQL allow data in a wide variety of systems to be easily shared and exploited in new ways. The ubiquitous XML standard allows data to be presented in a format that supports the widest possible distribution.

Using one of these technologies, however, is rarely a complete solution to the integration challenges faced

by today's enterprise. More often than not a number of these technologies are employed to piece together various systems. This is very much a case of choosing the best tools for the job.

Unlike GIS architectures that limit integration to one or two possibilities, the advanced Smallworld architecture supports a wide range of integration strategies giving the customer the ability to choose the most appropriate technology given the inevitable constraints of cost, time and existing technology.

The following sections give a brief overview of these integration technologies and give an indication of the wide range of integration possibilities provided by the Smallworld platform.

CORBA

The Common Object Request Broker Architecture (CORBA) is an architecture and a specification for creating, distributing and managing distributed program objects in a network. It allows programs at different locations and developed by different vendors to communicate in a network through an interface broker.

The essential concept in CORBA is the Object Request Broker (ORB). Support for an ORB in a network of computers means that a client program can request services from a server program without having to understand where the server is or what the interface to the server program looks like.

The Smallworld CORBA API allows Smallworld classes to be published to the CORBA universe and

also to link instances with suitable CORBA objects published by other applications.

A standard Interface Description Language (IDL) is used to define named interfaces within one or more modules. An interface might have attributes and methods, possibly with one or more arguments, and return a result. In addition, an interface may raise an exception with appropriate attributes.

The Smallworld CORBA API provides the behavior necessary to represent standard Smallworld classes and conditions using IDL interfaces and exceptions. Other applications can access the Smallworld classes through these IDL interfaces.

Supported ORBS are: Borland's VisiBroker® and IONA'S Orbix®.

EAI

Enterprise Application Integration (EAI) is a key technology allowing the integration and consolidation of a number of systems within an enterprise allowing the investment in existing technology to be reused and rejuvenated by new applications. EAI achieves this integration using a loosely coupled approach based on messaging and bus technologies.

The Smallworld EAI API provides classes to connect to a suitable messaging system and then to send and to receive messages within that system.

In the Smallworld EAI implementation, a message is a simple structure that has a body, a header and some properties. The body contains the message data and might be XML text or any structure constructed from base types supported by the underlying messaging system such as strings, integers and lists. The header itself contains metadata: for example, the time the message was sent and the details of the sender. Each property can contain a qualifier about the message itself¹.

¹ JMS only.

Examples of messaging systems that are supported by Smallworld EAI include IBM®'s WebSphere MQ®, Sun®'s Java Message Service (JMS) and TIB/Rendezvous® (TIBCO®). Connection classes are also provided for connecting to SonicMQ® from Sonic Software Corporation, Oracle® AQ, Vitria® from Vitria Technology, Inc. and for point to point messaging.

COM

The Component Object Model (COM) is Microsoft®'s framework for developing and supporting program component objects. It is aimed at providing similar capabilities to those defined in the Common Object Request Broker Architecture (CORBA) with the exception that all the objects involved reside on the same physical machine.

Smallworld's COM support provides two integration possibilities:

- It allows Smallworld applications to invoke functionality exposed via COM in an external application (such as Microsoft® Word® or Microsoft® Excel).
- It allows external development languages such as Microsoft® Visual Basic® or Microsoft® Visual C++ to invoke native Smallworld functionality®.

From a Smallworld development perspective, external COM objects manifest themselves within the Smallworld development environment as native classes. Broadly speaking, this means that methods and properties implemented on the external COM object appear as methods and properties on the equivalent Smallworld native class. By attempting to faithfully replicate the published COM API, this allows Smallworld applications to easily integrate with popular desktop applications such as Microsoft® Word® or Microsoft® Excel without the need to learn a new API. For example, a Smallworld application could feed the results of a complex network analysis into the fields of a mail shot program.

From an external development perspective, Smallworld's COM support allows native classes to be published as COM objects that can then be referenced in a language such as Microsoft® Visual Basic®. This allows external applications to exploit Smallworld's rich spatial functionality and specialized queries using a familiar API. For example, a simple Visual Basic® for Applications script could be used to get specific data from a Smallworld application that could then be referenced from a spreadsheet.

OLE DB

OLE DB is part of Microsoft® Data Access Components (MDAC) and provides a set of interfaces that expose data from a variety of relational and non-relational sources by using the Component Object Model (COM). OLE DB interfaces provide applications with uniform access to data stored in diverse information sources.

These interfaces support the amount of DBMS functionality appropriate to the data store, enabling the data store to share its data.

OLE DB components consist of data providers that contain and expose data; data consumers that use data and service components that process and transport data (such as query processors and cursor engines). In addition, OLE DB includes a bridge to ODBC to enable continued support for the broad range of ODBC relational database drivers.

The Smallworld OLE DB provider offers a way to send read only record data from a Smallworld database to an external OLE DB enabled application. Many OLE DB enabled applications exist (for example, Microsoft® Office) and many languages such as Microsoft® Visual Basic® allow OLE DB enabled ActiveX® controls and com classes to be embedded into applications providing, for example, easy to use table browsers.

SQL server and ODBC

SQL is a standard interactive and programming language for getting information from a database and also for updating it. Queries take the form of a command language that makes it possible to select, insert, update data, and so on. There is also a programming interface.

The Smallworld SQL Server provides remote read and write access to the Smallworld database from Windows®-based client applications.

Using the Smallworld SQL Server and a suitable Microsoft® ODBC enabled application, it is possible to quickly and easily interrogate any attribute data within a Smallworld database from a remote Windows®-based PC.

A great many ODBC enabled applications exist. For example, Microsoft® Word®, Microsoft® Excel, Crystal Solutions are all ODBC enabled allowing Smallworld data to be easily exchanged with these applications. ODBC is also a software development kit allowing custom applications to be written in other development languages such as Microsoft® Visual Basic®, Microsoft® Visual C#, and Borland® Delphi® to also access Smallworld® data.

The SQL Server supports a subset of the ANSI® SQL-92 (or SQL2) standard.

JDBC

Java™ Database Connectivity (JDBC®) is the Java™ version of ODBC. The JDBC® interface is a Java™ programming interface that allows a Java™ client application to access a wide range of data sources. The Smallworld JDBC® implementation provides a set of classes that use the JDBC® interface to enable connections from Smallworld clients to a number of external database systems, including Oracle®.

XML

XML (eXtensible Mark-up Language) is a flexible way to create common information formats and share both the format and the data on the world wide web, intranets and between applications.

The Smallworld platform provides an extensive implementation of the XML document object model based on Xerces, a Document Object Model (DOM) implementation provided by the Apache Software Foundation that supports Level 1 of the W3CDOM. The DOM allows random access to parts of the document, provides editing capabilities such as cut, copy and paste as well as the ability to create new XML documents from scratch.

A rich API is provided that allows a Smallworld application both to read XML data (stored for example in file) and to write data in the XML format using the Simple API for XML.

Support is also provided for very fast XML output of Smallworld table data, for example spatial data presented using the Geography Mark-up Language (GML).

The Smallworld XML API includes a Uniform Resource Name (URN) generator that allows URNs to be manufactured for all relevant object types (for example, individual records in the Smallworld database).

TCP/IP

The Smallworld platform supports TCP/IP communication to other systems and from other systems to itself using a sockets API. TCP/IP allows data to be transmitted between Smallworld and an external remote system that might be located in the same room or on another continent.

This API allows developers to build server-type applications that can be connected to by multiple clients. Clients can connect to a server application by specifying the network name (or IP address) of the

physical machine running the Smallworld system and the port number that is used to handle data traffic. Once connected, client applications can then exchange data with the server.

TCP/IP is the basis of many internet-based protocols (for example http). As such, the sockets API provides the low level mechanisms needed by many performance critical applications to communicate with a Smallworld application. However, TCP/IP is itself not a server application per se: it simply provides the base technology required to implement many server-like solutions.

Some common examples of the use of the sockets API include: monitoring of remote sensors and interfacing to proprietary data providers.

ACP

The Smallworld ACP API allows applications to take advantage of functionality in a third party library such as DLLS or Java® JAR libraries. To protect GIS data from rogue code or defects in a third party library, the Smallworld ACP API cleverly manages the third party's library from within a separate process. This approach prevents the third party library from corrupting data which, for most enterprises, is one of their most valuable assets. Parameters to the external API and data returned are streamed between the external process and the host process.

Some common examples of the use of the ACP API include: specialized power industry analysis software and high performance image compression tools.

Data marshalling

Data marshalling is a generic Smallworld API that supports the high speed bulk transfer of regularly shaped data (for example, records). The API allows this technology to be attached to existing streams enabling the very rapid exchange of data when using either the ACP or sockets APIS.

Data marshalling provides a low level template API that allows various data structures to be described. Larger number of instances of these structures can then be streamed between a server and a client in groups of a configurable size (for example, a hundred records). This ability to stream data in chunks supports user interface niceties such as progress bars and allows the operation to be interrupted by the user.

Data marshalling is also the foundation that supports many other Smallworld technologies such as OLE DB and JDBC® providing extremely fast transfer of data.

Conclusion

The Smallworld architecture provides a rich and powerful array of integration technologies that support interoperability with a wide range of disparate systems. Open standards such as com, ODBC, JDBC®, OLEDB and so on allow the Smallworld platform to interact with systems that range from desktop applications to corporate data warehouses. CORBA® and EAI are powerful integration technologies that allow enterprise systems to be interconnected to deliver maximum business benefit to the customer. Extensive XML support allows Smallworld applications to communicate in what is rapidly becoming the lingua franca of the internet.

This broad support for open standards, together with additional technologies such as TCP/IP, ACP and data marshalling that allow highly optimized integration solutions, gives the Smallworld technology the flexibility to meet the demands of today's enterprise integration challenges.

CORBA® is claimed as a registered trademark by Object Management Group, Inc., which is not affiliated with GE.

VisiBroker is claimed as a registered trademark by Visigenic Software, Inc., which is not affiliated with GE.

Orbix is claimed as a registered trademark by Iona Technologies pic, which is not affiliated with GE.

IBM and WebSphere are claimed as registered trademarks by International Business Machines Corporation, which is not affiliated with GE.

Sun and JDBC are claimed as registered trademarks and Java is claimed as a trademark by Sun Microsystems, Inc., which is not affiliated with GE.

Rendezvous and TIBCO are claimed as registered trademarks by TIBCO Software, Inc., which is not affiliated with GE.

SonicMQ is claimed as a registered trademark by Sonic Software Corporation, which is not affiliated with GE.

Oracle is claimed as a registered trademark by Oracle Corporation, which is not affiliated with GE.

Vitria is claimed as a registered trademark by Vitria Technology, Inc, which is not affiliated with GE.

Microsoft, Word, Windows, ActiveX, Visual Basic and Visual C++ are claimed as registered trademarks by Microsoft Corporation, which is not affiliated with GE.

Borland and Delphi are claimed as registered trademarks by BORLAND INTERNATIONAL, INC., which is not affiliated with GE.

ANSI is claimed as registered trademarks by American National Standards Institute Incorporated, which is not affiliated with General Electric.