

**ENSEMBLE WHITE PAPER**

# **THE TWO KEY TECHNOLOGIES FOR SOA SUCCESS**

Mike LaRocca, Product Manager  
InterSystems Corporation

# THE TWO KEY TECHNOLOGIES FOR SOA SUCCESS

## Introduction

Software services, in a service-oriented architecture (SOA), can be used again and again in many different business processes, making for a very flexible, efficient, and vibrant business and information technology infrastructure. Providing, of course, that IT has the right platform in place for modeling business processes and implementing the SOA. Using the InterSystems Ensemble product as an example, this paper examines two key technologies that are absolutely essential for successful SOA implementation, but missing from most SOA development platforms.

## Abstraction and Data Persistence

InterSystems Ensemble provides a single, comprehensive, and architecturally consistent solution for developing, integrating, orchestrating, deploying, and managing services in a service oriented architecture. Layered on top of the infrastructure you currently have in place, Ensemble provides all the functionality found in enterprise service bus (ESB) products commonly used for SOA implementation. But Ensemble goes further by providing two innovative and essential technologies that should be part of any effective SOA platform – *a comprehensive application and data abstraction facility, and an integrated object and meta-data repository.*

Abstraction technology hides the nature, variety, and complexity of the underlying interfaces and data that are part of the SOA implementation. The integrated repository, and the data persistence it provides, is the key to efficient reuse of services and SOA management. With this technology in the SOA platform, your IT organization can be assured of having the flexibility to adapt rapidly to changes in business requirements and processes, and investment protection through insulation from changes to the underlying technologies and standards.

## Concrete benefits from greater abstraction

Included among basic features, such as data transformation, asynchronous and synchronous processing, content-based routing, and distributed operation and management, most SOA platforms, including Ensemble, support Web services standards such as XML, WSDL, HTTP, and SOAP. And most, but not all, support the business process execution language (BPEL) standard for orchestrating services into business processes. These standards, and others, enable software developers to express service functionality, methods to access those functions over a network, and business process workflow, in an abstract format usable by any client wedded to the same standards. This level of abstraction is fine for IT environments implementing an SOA with new services written from the start using Web services standards. But for most IT shops, with a rich legacy of critical business applications and data that are not Web services enabled, it does not go far enough. It leaves IT without an easy way to bring the vast majority of its assets into an SOA and the business unable to reap the benefits. And standards compliance alone does not isolate the SOA platform or the SOA itself from the effect of changing technology and standards.

What is needed is the ability to abstract and represent the entire range of application functionality and data stores found in most IT environments into common forms that can then be shared out in whatever format is best suited to the task at hand. Abstraction reduces the complexity typically associated with integration projects (integration of services, in this case), enables incorporation of services and data as needed into different business processes, and drastically reduces the time to deployment.

Ensemble uses the object oriented programming paradigm and abstracts functionality and data as a consistent set of object classes with properties (data elements that objects of a given class will hold) and methods (functions that objects of a class can perform).<sup>1</sup> Such abstraction provides a consistent, unified view of underlying systems and applications – no matter what platforms, languages, data models, storage architectures, network protocols, or other technologies those underlying systems use.

<sup>1</sup> See <<http://www.answers.com/object-oriented-programming>> for a brief description of object orientation, classes, objects, and methods.

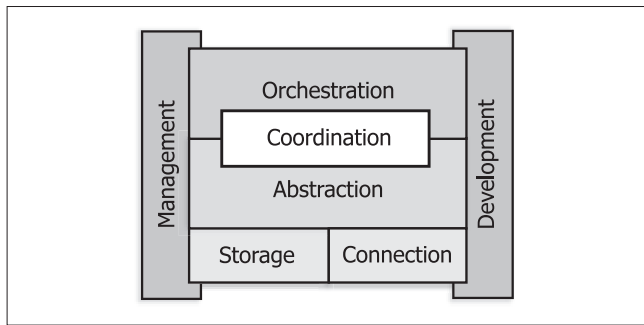


Figure 1. Ensemble functional architecture.

In Ensemble, the product’s abstraction layer rests on top of the connectivity and storage layers, hiding the nature, variety, and complexity of the underlying interfaces and data. Ensemble’s development environment provides simple ways for a developer to identify those capabilities and data that are of interest for a given business process, and then transform those capabilities and data into the Ensemble objects that are visible to all components of the solution. As an example, consider a project that connects an SAP application suite, a legacy mainframe application with an IMS database, a new Windows application that exposes functionality as Web services, a Java application built using J2EE technologies, and an application based on a relational database with extensive stored procedures. Ensemble provides abstraction in two ways:

- Within the Ensemble abstraction layer, the functionality and data in these applications are seen as a set of object classes with properties and methods. Ensemble shields business analysts and developers from the complexity of the multiple languages, object models, databases, platforms, and other technologies used to build and deploy the original five applications. This is “inward” abstraction – Ensemble makes resources from the outside world available to itself.
- Once these external application resources are available as Ensemble classes, Ensemble can incorporate them into the business process, which can be accessed by an external client technology such as COM, .NET, ODBC, Java, JDBC, EJB, XML, and Web services. This is projecting the Ensemble abstraction outward – making internal resources available to the outside world.

Abstraction in Ensemble provides a consistent and efficient object representation of different programming models and data formats and a “universal” or “global” view of all data and requests traveling through the system. As a result of its abstraction capability, Ensemble:

- Facilitates rapid development of composite applications in business processes through powerful abstractions of both logic and data.
- Enables the latest development tools and technologies to access legacy data and functionality as reusable .NET or J2EE components, Web services, or XML.
- Provides equal support for various technology frameworks, including J2EE and .NET, while being easily extensible for future object models and frameworks.

Whereas Java-based systems have strong logic abstraction but weak data abstraction, and XML-based systems have strong data abstraction but weak logic abstraction, Ensemble abstraction provides the missing dimensions required to consistently support sophisticated business processes, which may include elements from both types of system. SOA platforms that are J2EE-specific or .NET-specific risk the need to “lock in” a specific vendor or technology, but Ensemble avoids this risk by providing equal support for various technology frameworks.

## Persistence pays in the SOA world

In the world of SOA platforms amnesia is rampant. With technology that has grown out of the messaging world, many SOA platforms have no integrated and persistent data store. Messages are created, pass through the system, and are gone. Data important to the entire SOA solution is held at the outer edges of the environment, in the various systems hosting services, and not readily available to the orchestrated business process of which the service is but a part.

Some SOA platforms do provide for links to an optional, separately licensed third-party relational database. But, besides the additional expense, using an external database for SOA storage incurs significant processing overhead for calls out to the database, and for the mapping of multidimensional XML

and other data structures into and out of two-dimensional relational tables. Using an external database puts performance of the entire solution at risk, increases the complexity of the solution, and raises management costs.

What's needed is an integrated, centralized, high-performance persistent data store to hold distributed SOA data and metadata. An SOA without one is brain dead – it functions, but only at the most basic level, with zero intelligence. What is lost is a wealth of data and metadata that could be used to make development of the overall solution easier, and for business activity monitoring, problem tracing and resolution, reporting, auditing, and other management functions.

## **Operational efficiency and awareness – data and metadata storage**

SOA storage requirements range from cross-application indices that enable multiple services to work together, to high-demand information for which relational database applications cannot provide adequate performance. Ensemble addresses these requirements with an embedded, highly reliable, high-speed, persistent, object database. It supports message persistence, transaction integrity, event logging and tracing, and high-performance transactional bitmap-indexed data retrieval.

In Ensemble, all components of a given business process, from the process itself to executable code and messages, are represented in the database as a consistent set of classes, objects, and methods. Because Ensemble's run-time execution engine and the database engine are the same, all components of the solution can execute in the same in-memory process for exceptional speed.

The database embedded in Ensemble provides SOA solutions with:

- Highly-scalable storage of solution-specific data or indices, with built-in persistent storage capabilities
- Scalability to tens of thousands of concurrent users and terabytes of data
- Transactional capabilities, including automatic recovery from system failures

- Read-write access via objects, SQL, and direct mode
- Live-system full, cumulative, and incremental data backup
- Technologies for distributed data storage and access with full transparency

Metadata is the key element enabling use of services within an SOA. We need to know what services exist before we can even think of using them. We need to know where they are, and we need to know how to use them. We need to know how to string them together to model the business process. And finally we need a place (besides our leaky brains) for persistent storage of all the data flowing through the system.

Ensemble uses its embedded database for storage of metadata as well as transactional data, messages, and other objects. This shared metadata repository contains information about all the services and other components in the environment. It provides the foundation for a common and consistent framework for integrating services and data sources, orchestrating business processes, and developing composite applications. With the metadata readily available, it's possible to create a consistent set of interfaces to, and information about, each component in the SOA, enabling faster integration, rapid development, easier management, and greater extensibility.

Several benefits accrue when the SOA platform contains its own integrated data store. In Ensemble these benefits include:

- **Efficient investigative and analytical capabilities** through read-only access to a warehouse of messages of arbitrary complexity, with full transactional state.
- **Highly reliable long-running business processes.** Ensemble automatically stores the state of all business processes executing. This convention provides a strong degree of recoverability in the event of unplanned system outages or other incidents.
- **Access to configuration data and other metadata** with the same reliability, scalability, and performance advantages as the enterprise data within the solution. Metadata can be accessed programmatically via SQL and object views, or visually using the Ensemble Configuration portal.

- **Data synchronization** using simple formulas or lookups in data tables (internal or external), extensible to any degree of complexity by adding customized functions. Coupled with Ensemble abstraction and other functions, Ensemble storage enables you to universally synchronize data across the solution.

- **Performance enhancement**, as Ensemble can cache copies of frequently accessed data on behalf of any part of the solution. This persistent cache capability avoids the typical bottleneck in integration scenarios – the need to access data inside overworked corporate databases.

- **Real-time data analysis.** Ensemble’s transaction-oriented bitmap indexing capability permits the rapid retrieval operations that are essential to support business intelligence and real-time data analysis.

## Management

Service oriented architectures consist by design of loosely coupled services. Such systems tend to be difficult to manage. Asynchronous business processes and message-oriented services make it difficult to trace threads of execution, and therefore to debug them. Persistent storage, especially on the scale and scope possible with Ensemble, addresses this difficulty. Ensemble provides a wealth of diagnostic data and the tools to analyze it in real time. Management features include:

- The ability to analyze stored messages (and other information) and trace message paths
- Message warehouse maintenance
- Configuration control
- Queue and process monitoring
- Detailed event logs
- Usage histograms

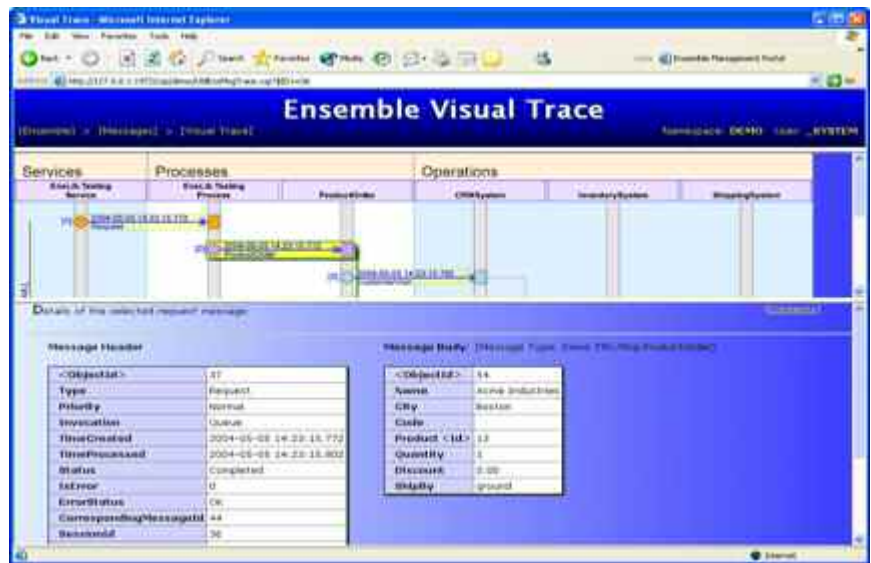


Figure 2. Storage of every message passing through the system, and associated metadata, provides users with powerful business process tracking, debugging, and reporting mechanisms.

## Conclusion

Some SOA platforms are really just a collection of independent tools packaged together and lacking the integration necessary to make them easy to learn and efficient to use. Others are integrated toolsets but are hobbled by the lack of abstraction technology and an integrated data store. They are inefficient development environments with limited support for the wide range of application interfaces, data formats, and technologies typically found in all but the smallest IT environments.

At the end of the day innovative abstraction and data persistence technology in Ensemble makes it:

- Easier to focus on improving business processes instead of which technology to use
- Easier and faster to develop and deploy IT assets as services in a flexible service-oriented architecture
- Easier to bring legacy application functionality into the service oriented architecture
- Easier to align IT assets with frequently changing business requirements and processes
- Easier to work within your existing frameworks and methodologies
- More cost-effective and less risky to move to a service oriented architecture

InterSystems Corporation  
World Headquarters  
One Memorial Drive  
Cambridge, MA 02142-1356  
Tel: +1.617.621.0600  
Fax: +1.617.494.1631  
[InterSystems.com](http://InterSystems.com)

The logo for InterSystems, featuring the word "INTERSYSTEMS" in a stylized, blue, serif font. The letters are closely spaced and have a slightly irregular, hand-drawn appearance.