Information Builders

Worst Practices in SOA Implementation

Why Service-Oriented Architectures Fail

A White Paper

by John Senor

iWay Software

John Senor

John Senor is president and chief operating officer of iWay Software – an Information Builders company and market leader in middleware that accelerates business integration. With facilities to integrate systems in real time, near-real time, or on a scheduled basis, iWay provides a complete reusable infrastructure for EAI, B2B, e-commerce, and mobile business to over thousands of customers worldwide. iWay's experience with complex information systems led to the development of prepackaged intelligent adapters to connect to more than 250 packaged applications, transaction systems, legacy data, relational databases, and e-business formats, without writing custom code. iWay Software integration solutions significantly reduce the time, cost, effort, and risk of integration projects throughout the enterprise.

Senor held the position of group vice president and general manager of Information Builders' Middleware Technology Group, responsible for the overall strategy and development of Information Builders' comprehensive line of middleware software products. He is a frequent speaker at technology trade shows and a highly published author on integration topics.

In 1991, Senor founded the EDA Division and the EDA/SQL middleware product line as a new business venture for Information Builders, and quickly became one of the leading players in the evolution of the middleware software industry.

Senor joined Information Builders in 1988 as director of marketing, before founding the company's middleware business.

During his twenty-eight year career in the industry, he has held senior management positions with the IBM Corporation, Informatics General Corporation, Applied Data Research, and Cullinet Software.

He is a 1970 graduate of the United States Military Academy at West Point, New York and holds a Bachelor of Science degree in Engineering.

Table of Contents

Turning Failure Into Success

Over the last two-plus decades, most organizations have come to rely on information technology (IT) systems to run their business. Now, with a variety of mission-critical applications and tools in place, many organizations see the value in integrating these systems in order to operate more efficiently, provide better customer service, and develop new revenue-generating activities. According to Gartner, integration increases the value of application portfolios and positions IT to use their portfolios to deliver improved business value.[1]

In the computer industry, integration is a general term for any software that serves to join together or mediate between two separate and usually already existing programs, applications, or systems. From a business perspective, integration is about creating automated business processes – known as workflows – to codify routines that were once performed manually.

A new integration standard has evolved in recent years known as service-oriented architecture (SOA). SOA enables different programs, applications, and tools to interact via self-contained services that do not depend on the context or state of the other service. Working within a distributed systems architecture, SOA has gained momentum because it creates reusable integrated business processes.

While tracking mediocre results, and even failure[2], in the implementation of service-oriented architectures, many common threads, or "worst practices," can be found. The top-four worst practices for SOA integration include:

- Overemphasizing low-level code
- Centralizing design and development
- Ripping and replacing legacy software
- Buying software without support

These worst practices set companies on the inauspicious path of SOA failure. They have been repeated by some of the best run and smartest companies in the world. Typically, these worst practices are the result of wanting to ride the latest technology wave without balancing the hype with practical knowledge and experience. For an SOA integration initiative to be successful, organizations must think about the long-term health of their architecture, even while deploying short-term solutions.

Designed to help organizations learn from the mistakes of others, this paper provides insight into the top-four worst practices for SOA integration. It also provides guidance on how to avoid and/or overcome these worst practices in order to realize the true value of an open, reusable integration architecture. By reading this paper, you will have a solid understanding of how to avoid SOA integration failure and achieve success with your initiatives.

[1] Thompson, Jess. "Questions About Application Integration Underscore Its Pervasive Use." Gartner. ID Number: G00145418. June 20, 2007.

[2] Failure is defined as a considerable expense with little or no return on investment.

Worst Practice #1: Overemphasizing Low-Level Code

A doomed – yet typical – SOA integration scenario goes something like this: Company X purchases an integration tool. The company's integration developers immediately begin automating business processes. The integration solution works for a short while, but soon business processes change and developers are forced to modify the code.

These business-process changes can result from enterprise-wide changes, such as merging with or acquiring another company, or something relatively minor, like contracting a new supply chain vendor. As the developers change the SOA integration solution's underlying code, the system becomes clunky, slow, and less able to adapt to additional changes.

This approach, rooted in the way enterprise application integration (EAI) was historically performed, addresses only specific workflow integrations. It does not allow an organization to create an open, reusable architecture.

iWay's Response: Focus on Business-Level Services

SOA implementations are not about connecting specific steps in a workflow to mirror existing business processes. Organizations that have built successful service-oriented architectures build services for specific projects and then incrementally develop new services that are consistent and work in a mutually supportive way. By breaking the entire system into logical building blocks – also known as services – a sustainable solution is created that can grow along with business needs.

These services, which are nothing more than reusable bits of functionality, can be divided into three levels: fine-grain services, coarse-grain services, and global services.

When organizations overemphasize low-level services, the net result is too many services that don't aggregate up into business-level services. Coding this way creates slow and complex process flows that are not easy to maintain or reuse.

Take for example the workflow required to process a purchase order. Typically a credit check is performed and then inventory levels are determined. Are there other business scenarios in which a customer's credit history must be verified? Probably. The same is also true for inventory levels. By creating these processes as reusable services instead of hard coding the order-processing workflow, checking customer credit or inventory levels only needs to be coded once and reused by any other existing or future business routine that requires that information.

Once an organization gets the architecture design right, everything else naturally follows. For an SOA initiative to be successful, organizations must move away from the EAI paradigm. When businesses processes become too complex, the amount of code developers are required to write increases exponentially. iWay Software encourages organizations to identify recognizable integration patterns. Every repeatable integration pattern should be developed as a reusable service.

Worst Practice #2: Centralizing Design and Development

Here's another example of an SOA integration project headed for failure: After purchasing its SOA integration tools, Company X assigns a centralized team to design and develop the application. These developers, who are part of the corporate IT organization, are able to code the services that are part of their day-to-day activities because they know the applications involved very well. But troubles soon emerge.

One problem is that only a few people will know how the SOA application was built. As people leave, so too does that knowledge. But this is not the biggest pitfall of centralizing the design and development of a service-oriented architecture.

Centralized groups create service isolation, which undermines SOA integration initiatives. Big problems arise when the centralized development team starts to code services for applications that they do not work with every day and therefore do not know thoroughly. For example, SAP may be used in Germany, but not in the U.S. where the corporate SOA integration team is located. Coding SAP services requires a familiarity with the application and how it works. By asking developers unfamiliar with SAP to code the requisite services, organizations run the risk of inefficient services that do not conform to standards.

iWay's Response: Decentralizing Service Creation

Services should be decentralized and locally maintained. Only in this model can they ascend to higher-level services without forcing the coarse-grained and global service developers to try and keep up with underlying information assets. Consider that each information asset is constantly changing as a result of new releases, upgrades, fixes, and patches. A centralized design and development team cannot know about every change made to every system in the enterprise.

When the design and development teams are decentralized, local developers and administrators can change the underlying services without affecting the entire system. By developing the service close to the information asset, the people who work with the application every day and know its ins, outs, and quirks are responsible for making any necessary changes to the services.

SAP, for example, is a complex application and developers need to know how to use the tool to develop services. Local SAP developers are much better suited to creating the library of reusable SAP services since they have the requisite expertise in the technology. These services are then put into a registry managed by a centralized developer. The centralized developer is able to create workflows without actually maintaining the services.

When services are created, implemented, and managed by people who have the expertise in manipulating the specific databases or applications, the development of coarse-grained and global services is much faster and more efficient since developers don't become engulfed in the intricacies of each application.

Worst Practice #3: Ripping and Replacing Legacy Software

When it comes to updating IT infrastructures, some organizations believe this is a good time to replace legacy systems with new technologies. This approach stems from the notion that new technology must be better. In terms of SOA integration, organizations view the data in their legacy systems, not the application, as the main information asset. According to this notion, as long as the data is ported to another environment nothing is lost.

This strategy doesn't address the magnitude of data in legacy systems – an estimated 70 percent of the world's data[3] – nor does it recognize the fact that the application itself is truly the asset. When put into action a rip-and-replace approach pans out as follows: Company X has decided to move forward with an SOA initiative, but as long as they are upgrading their IT architecture they also opt to update some of their legacy systems at the same time. The applications that they choose to upgrade include purchasing, manufacturing, finance, and payroll.

While attempting to port the data into its new environment, unexpected problems arise, delaying the move toward a reusable architecture and – more significantly – negatively impacting the day-to-day business operations of their mission-critical applications.

iWay's Response: Reuse Is the Muse

If a legacy application is working, don't try to fix it. Upgrading technology for technology's sake is never a sound business option.

The programs and applications used by mature IT shops have legacy messaging technology in place and use proprietary interfaces or a slightly different interpretation of the JMS standard. Instead of ripping and replacing these messaging systems, iWay Software's SOA integration solutions run on top of commonly used protocols. This gives iWay Software customers messaging interoperability, which allows them to integrate multiple integration environments without changing the underlying application.

Maintaining systems crucial to an organization's business is essential because they house the data and business processes that differentiate a company from its competitors and represent years of valuable intellectual property. Ripping legacy systems out and replacing them with newer systems, when less drastic alternatives still exist, makes little fiscal or strategic sense.

[3] "Gentry, Joe. "Is 'Rip and Replace' the Only Way to Deal With Legacy Systems?" ebizQ. May 2005.

Worst Practice #4: Buying Software Without Support

Purchasing SOA software is only the first step. How the software is used to implement a reusable service-oriented architecture is what really matters. One surefire way to undermine an SOA initiative is to purchase software without support.

Use the following example to visualize this worst practice in action: Company X purchases software that promises to turn their SOA dreams into reality, but in keeping a close eye on their expenditures, they decide to forgo any support from the vendor's professional services or consulting organization. The developers at Company X begin working on the SOA integration project, but soon run into trouble.

Two of the most egregious problems caused by not having the sufficient know-how to operate SOA software include building an overly complex system and not using the right tools for the job.

Overly complex systems are built when organizations do not fully recognize that an SOA implementation is inherently different from EAI architectures. To be successful, developers must first learn to recognize what should be a service and how to aggregate them into higher levels. Yet, most organizations don't have developers on staff with the design expertise to do this. Instead they code EAI workflows, which become overly cumbersome and complex as the system is modified and changed. Organizations then blame their failure on the software, instead of the fact that they didn't know how to properly design, develop, and deploy a service-oriented architecture.

Without sufficient support, XML can also become an organization's Achilles' heel in terms of creating an overly complex system. Many vendors tout XML as the end-all and be-all for SOA. And, while XML is necessary it is not always as simple as it appears. In the e-business world, for example, messages converted into XML become huge. XML transformation engines were not designed to handle such large messages, which can become hundreds of megabytes. When messages are this large, the system slows down. Support services, however, can help organizations split and accumulate these messages into manageable sizes.

A second often-seen effect of not having sufficient support while developing an SOA strategy is organizations not using the right tool for the right job. Because integration projects involve connecting many different applications, data sources, and business processes, no single product or engine can be used to accomplish every task. Each function of a service-oriented architecture requires different tools or features.

iWay's Response: Support Ensures Success

Having an overall SOA architect means that organizations establish set principles in regards to what their services look like. To be successful, organizations need people who can recognize integration and help create reusable services.

A learning curve exists for every integration project. And, even though the iWay SOA Middleware suite of products is easier to use then other tools on the market, we stress that organizations secure support from the very beginning of the project.

In an ideal SOA integration initiative, iWay Software's Professional Services group designs and implements the first project with the customer looking over our shoulders every step of the way. For the second project, the customer designs and implements the solution and we look over their shoulders. By the third project, the customer will be self-sufficient, but support services are available when needed.

How an organization designs message transformation and splitting makes a difference. It is better to take the time to do this up front than to struggle with performance issues once the architecture is up and running. By proceeding with sufficient support, organizations can avoid developing overly complex systems and know which tools are the right tools for which jobs. iWay Software can help developers recognize the different types of messages being processed and help you achieve acceptable performance levels.

You Have the Antidote

While some of what has been mentioned in this paper may seem like common sense, you can bet that someone in your organization will begin efforts that will put into effect at least one of these worst practices. Who can blame them when industry trade journals, the vendors, and technology consultants promote the latest technology and promise all sorts of benefits? It's easy to get caught up in the hype.

The good news is that you are now well armed to identify and combat at least these four worst practices before they take root and grow into a strangling vine. In addition to the solutions presented in this paper, you can also use counter-intuition to the worst practices and provide yourself with a clearer path to success. Consider the following:

- **Emphasizing business-level services** encourages you to think about the long term, even while deploying short-term solutions
- **Decentralizing design and development and centralizing management** speeds development time and improves efficiency, while at the same time ensuring that standards are met
- **Reusing technologies already in place** ensures that your business continues to run and you make use of your intellectual property
- **Using experts for support services** maximizes the value of your investment and helps you to build an open architecture and identify services for reuse

These steps will deliver an end result with a clearly defined ROI. With pervasive reuse as a goal and no limit on what can be service-enabled – you are able to start small, but think big.

While these four steps can be followed regardless of your choice of SOA software, we strongly believe iWay Software will give you the appropriate blend of integration choices and the most flexible tools. All of this will allow you to build a service-oriented architecture that will help your organization achieve its business goals.

Sales and Consulting Offices

North America

United States

- **Atlanta,*** GA (770) 395-9913
- **Baltimore,** MD Consulting: (703) 247-5565
- **Boston,*** MA (781) 224-7660
- **Channels,** (800) 969-4636
- **Charlotte,** NC Consulting: (704) 494-2680
- **Chicago,*** IL (630) 971-6700
- **Cincinnati,*** OH (513) 891-2338
- **Cleveland,** OH (216) 520-1333
- **Dallas,*** TX (972) 490-1300
- **Denver,*** CO (303) 770-4440
- **Detroit,*** MI (248) 641-8820
- **Federal Systems,*** DC (703) 276-9006
- **Hartford,** CT (860) 249-7229
- **Houston,*** TX (713) 952-4800
- **Los Angeles,*** CA (310) 615-0735
- **Mid-Atlantic**
  New Jersey* Sales: (973) 593-0022
  Philadelphia,* PA Sales: (610) 940-0790
  Pittsburgh, PA Sales: (412) 494-9699
- **Minneapolis,*** MN (651) 602-9100
- **New York,*** NY Sales: (212) 736-7928
  Consulting: (212) 736-4433, ext. 4443
- **Orlando,*** FL (407) 804-8000
- **Phoenix,** AZ (480) 346-1095
- **St. Louis,*** MO (636) 519-1411
- **San Jose,*** CA (408) 453-7600
- **Seattle,** WA (206) 624-9055
- **Washington,*** DC Sales: (703) 276-9006
  Consulting: (703) 247-5565

Canada

Information Builders (Canada) Inc.

- **Calgary** (403) 538-5415
- **Ottawa** (613) 233-0865
- **Montreal*** (514) 421-1555
- **Toronto*** (416) 364-2760
- **Vancouver** (604) 688-2499

Mexico

Information Builders Mexico

- **Mexico City** 52-55-5062-0660

Australia

Information Builders Pty. Ltd.

- **Melbourne*** 61-3-9631-7900
- **Sydney*** 61-2-8223-0600

Europe

- **Belgium*** Information Builders Belgium
  Brussels 32-2-7430240
- **France*** Information Builders France S.A.
  Paris 33-14-507-6600
- **Germany** Information Builders (Deutschland)
  Dusseldorf 49-211-523-91-0
  Eschborn* 49-6196-77576-0
  Munich 49-89-35489-0
  Stuttgart 49-711-7287288-0
- **Netherlands*** Information Builders (Netherlands) B.V.
  Amsterdam 31-20-4563333
- **Portugal** Information Builders Portugal
  Lisbon 351-217-217-491
- **Spain** Information Builders Iberica S.A.
  Barcelona 34-93-344-32-70
  Bilbao 34-94-425-72-24
  Madrid* 34-91-710-22-75
- **Switzerland** Information Builders Switzerland AG
  Dietlikon 41-44-839-49-49
- **United Kingdom*** Information Builders (UK) Ltd.
  London 44-845-658-8484

Representatives

- **Austria** Raiffeisen Informatik Consulting GmbH
  Vienna 43-12-1136-3870
- **Brazil** InfoBuild Brazil Ltda.
  São Paulo 55-11-3285-1050
- **China**
  InfoBuild China, Inc.
  Shanghai 86-21-5080-5432
  Rongji Software Technology Co., Ltd.
  Beijing 86-10-5873-2031
- **Ethiopia** MKTY IT Services Plc
  Addis Ababa 251-11-5501933
- **Finland** InfoBuild Oy
  Espoo 358-207-580-843
- **Greece** Applied Science
  Athens 30-210-699-8225
- **Guatemala** IDS de Centroamerica
  Guatemala City 502-2361-0506
- **Gulf States** Nesma Advanced Technologies
  ■ Bahrain ■ Kuwait ■ Oman
  ■ Qatar ■ Yemen ■ United Arab Emirates
  Riyadh 96-1-465-6767
- **India*** InfoBuild India
  Chennai 91-44-42177082
- **Israel** NESS A.T. Ltd.
  Tel Aviv 972-3-5483638
- **Italy** Selesta G C Applications S.P.A.
  Genova 39-010-64201-224
  Milan 39-02-2515181
  Torino 39-011-5513-211
- **Japan** K.K. Ashisuto
  Osaka 81-6-6373-7113
  Tokyo 81-3-5276-5863
- **Malaysia** Elite Software Technology Sdn Bhd
  Kuala Lumpur 60-3-21165682
- **Norway** InfoBuild Norway
  Oslo 47-23-10-02-80
- **Philippines** Beacon Frontline Solutions, Inc.
  63-2-750-1972
- **Russian Federation** FOBOS Plus Co., Ltd.
  Moscow 7-495-124-0810
- **Saudi Arabia** Nesma Advanced Technology Co.
  Riyadh 996-1-4656767
- **Singapore**
  Automatic Identification Technology Ltd.
  65-6286-2922
- **South Africa** Fujitsu Services (Pty.) Ltd.
  Johannesburg 27-11-2335911
- **South Korea** Unitech Infocom Co. Ltd.
  Seoul 82-2-2026-3100
- **Sweden**
  InfoBuild AB
  Kista 46-735-24-34-97
  Cybernetics Business Solutions AB
  Solna 46-7539900
- **Taiwan** Galaxy Software Services
  Taipei 886-2-2586-7890
- **Thailand** Datapro Computer Systems Co. Ltd.
  Bangkok 662-679-1927, ext. 200
- **Venezuela** InfoServices Consulting
  Caracas 58-212-763-1653

Toll-Free Number

- **Sales, ISV, VAR, and SI Partner Information**
  (800) 969-4636

*Training facilities are located at these branches.
**Authorized to sell iWay Software only.

Corporate Headquarters Two Penn Plaza, New York, NY 10121-2898 (866)297-4929
www.iwaysoftware.com info@iwaysoftware.com

For International Inquiries +1(212) 736-4433

DN7505726.1007

Printed in the U.S.A.
on recycled paper