

Impress and Service Oriented Architectures

Document for:

Internal and external distribution

Version 1.9

October 30, 2006

Created by:

Impress Software
460 Totten Pond Rd. Suite 200
Waltham, MA 02451

1	Introduction	3
2	The Basics of SOA.....	3
2.1	Key Elements of a SOA	4
2.2	SOA Benefits.....	4
	Reduced Development Costs through Software Re-use.....	4
	Increased Developer Productivity	4
	Increased Agility.....	4
	Improved Alignment between IT and Business	4
3	Impress Solutions	4
3.1	Packaged Integration Applications.....	5
4	Impress and SOA.....	5
4.1	Impress as a Service Provider	6
4.1.1	Service Provider Example	6
4.2	Impress as a Service Consumer	7
4.3	The Road to SOA Compliance.....	7
5	Conclusions	8
6	About Impress Software	8

1 Introduction

There's a major push underway in the IT industry towards Service Oriented Architectures (SOAs). This shift is for good reason - SOAs promise to unleash the many benefits offered by a service oriented development methodology, namely faster development cycles and adaptability. For years, Impress has realized the benefits of a service-oriented development approach to deliver integration solutions that address complex integration scenarios between ERP systems and Enterprise Project Management or Geographical Information Systems. With the emergence of important service standards (e.g. web services) and industry adoption of SOAs, Impress has evolved its service-based architecture to specifically support SOA standards.

This whitepaper discusses Impress's support for SOAs. The paper begins by providing a basic overview of SOA and Impress's packaged integration applications. The whitepaper then describes Impress's role as a Service Provider and a Service Consumer, two important elements of a SOA.

2 The Basics of SOA

Let's start by understanding the basics of a SOA. First, SOA is a strategy; a strategy that espouses the creation of all software assets in a company using a service-oriented programming methodology. The most fundamental building blocks of SOA are services.

Services are software chunks, or components, constructed so that they can be easily asked to perform specific functions – usually by other services – and reused in many different areas across a company. Within a SOA, these services offer their functionality as individual services in a standardized way. One such service could be “Create a sales order” or “Display a project plan”. Once applications provide their functionality as services, the idea is that these services can be combined across systems to form new “composite” applications.

Services can be connected in many different ways, such as custom programming links or integration software from vendors. Since 2001, however, a set of software communication mechanisms known as web services, which are based on W3C standards, have become the de facto approach for linking software components together. The concept of web services is often implied when SOA is referred. But web service is a linking and communications methodology, while SOA is an overall IT strategy.

The basics components of a SOA are depicted in Figure 1 below. The Service Directory contains a library of existing web services and instructions on how to invoke each one. The Service Provider is a web service which conforms to a specific set of interface standards (e.g. SOAP and WSDL) and registers with the Service Directory so that other web services and applications can invoke and use it. The Service Consumer finds Service Providers via the Service Directory and invokes them using instructions provided by the Service Directory. The actual location of the Service Provider is transparent to the Service Consumer.

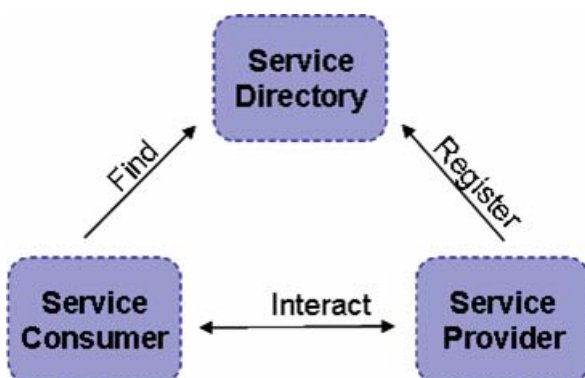


Figure 1 – Service Oriented Architecture Overview

Because services can be accessed in a standard way they provide unprecedented flexibility. Business processes can be added or altered quickly. Software applications can be integrated easily.

2.1 Key Elements of a SOA

- Centralized development and project methodology
- Software applications that provide their discrete functions as web services
- Tools, such as a business process modeler, to design processes and assemble web services
- A centralized repository that functions as a web service library
- A broker that executes services and routes data between services
- A front-end that enables collaboration between people and processes

2.2 SOA Benefits

Reduced Development Costs through Software Re-use

Web services, if designed to the right “size and scope” can be shared among many applications, reducing development, testing and integration costs.

Increased Developer Productivity

With developers able to re-use web services, software projects go faster and the same resources can work on more projects.

Increased Agility

Systems that are made up of a series of isolated components become easier to modify. Specific components can be modified without the need to tear apart and rebuild the entire system.

Improved Alignment between IT and Business

SOA calls for IT to take a business-oriented approach to building applications. The approach is less about building complex applications and more about understanding business processes and assembling pre-existing web services together. In fact, the long-term vision of SOA is to enable non-technical resources that understand business processes to modify and assemble different services together.

3 Impress Solutions

Impress Software develops and markets packaged integration applications – Impress for EPM and Impress for GIS - that are focused on streamlining common business processes that span ERP and specific best-of-breed applications.

Impress for EPM is optimized for bridging SAP Plant Maintenance (PM) and Project System (PS) with project management systems from Microsoft and Primavera. By synchronizing schedule, material, resource, and time and confirmation data, it enables project managers and maintenance schedulers to more effectively utilize the planning and analysis features of their project management systems, while SAP analysts manage the financial accounting for these projects.

Impress for GIS bridges asset data and maintenance information between SAP and Geographic Information System (GIS) solutions from ESRI and Miner & Miner. It enables workforces to access and perform key asset and work management processes from the familiar confines of GIS, while automatically synchronizing data to ensure both SAP and GIS remain up-to-date and accurate.

3.1 Packaged Integration Applications

Technically, a packaged integration application is comprised of the following five elements:

Application UI	Windows-based clients, application plug-ins
Application Logic	Domain specific business processes
Framework	General Integration Tools
Access Components	Domain specific technical system knowledge
Integration Server	Core technology

Application UI: Windows-based clients that provide access to the administration, configuration, and operation GUIs for Impress solutions. Much of this functionality can be plugged directly into portal interfaces. In addition, Impress provides application plug-ins that enable seamless access to specific ERP functions directly from external applications such as ESRI's ArcGIS.

Application Logic: Consists of out-of-the-box integration processes that are supported and maintained by Impress Software. These processes are based on integration best practices and are highly flexible due to the underlying Framework.

Framework: The integration toolset to develop, configure, administer, and operate complex integration scenarios. The Framework includes tools and technical processes to perform consistency management between systems, maintain cross-system relationships, detect specific field updates, and perform exception handling and notification.

Access Components: Provides the application-specific know-how and processes to optimally complete specific business processes. This includes optimized system access, data validation and error recovery.

Integration Server: Provides the high performance connectivity into participating systems and the Impress repository. The server also facilitates the execution of integration application processes, including real-time data validation and exception handling.

One basic function of a packaged integration application is to move data from one application to another, similar to the central function of a message bus. But packaged integration applications do much more than simply calling back-end system services and directing data flows. For example, Impress solutions can apply a pre-determined set of best practice integration processes for a given business solution (e.g. capital project, turnarounds, or spatial asset management). In a capital project scenario, for instance, Impress processes are much more complex and robust than "load project from Microsoft Project and save it in SAP". Instead, they are more like "load project from Microsoft Project, aggregate numbers (e.g. costs, planned hours) into controlling structures, calculate milestones for cash flow planning and recreate the multiple currencies and time planning units that exist within SAP following the pattern of last week's SAP data conditions."

4 Impress and SOA

Impress supports SOAs as both a Service Provider and a Service Consumer. As a Service Provider, Impress exposes many of its business processes – such as synchronizing data between SAP and Microsoft – as web services. Impress also consumes web services provided by specific back-end systems to interact with these systems. It's important to point out that the back-end systems Impress ties together are in various stages of web service adoption and some still require interaction via a proprietary API. Thus, Impress is designed to interact with systems leveraging both web services and APIs where appropriate. Impress will continue to keep pace with partners as they further incorporate SOA principles, namely web services, into their solutions.

4.1 Impress as a Service Provider

As a Service Provider, Impress exposes many of its business processes as web services. For example, the following web services are included with the Impress for EPM product:

- Synchronize a project
- Synchronize a group of projects
- Retrieve synchronization configuration information
- Retrieve synchronization history

Figure 2 below depicts Impress within a SOA along with other systems within which Impress interacts. Once Impress web services are registered in the service directory (e.g. via UDDI), they can be used by a W3C-compliant platform, such as SAP XI, to easily access Impress functions when building composite applications. To ensure high performance and reliability as a Service Provider, Impress interacts with the back-end systems (SAP, EPM, and GIS) via API or direct web service calls.

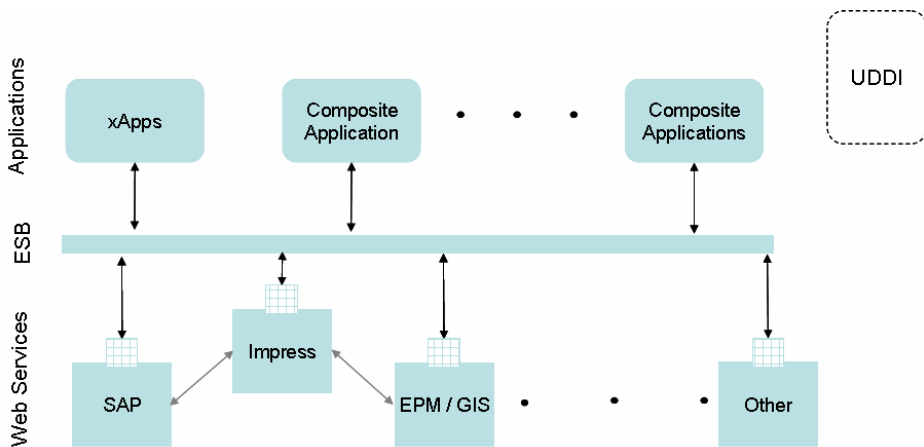
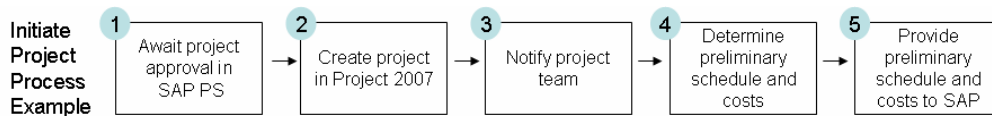


Figure 2 - Impress as a Service Provider

In addition to providing out-of-the-box web services, Impress also enables the creation of new web services for any of the Impress processes. A wizard-based tool is provided to facilitate this creation process.

4.1.1 Service Provider Example

Acme Corporation has created a composite application involving SAP, Microsoft, and Impress that automates its internal business process for initiating new projects. The fictitious business flow might look like the following:



The project definition begins in SAP, but once approved needs to be transferred to Microsoft Project where planning, scheduling, and execution take place. So the flow is performed in the following way:

Step 1: A project coordinator uses the SAP PS user interface to set the project status to “approved”. Once approved, the process flow moves on to the next step.

Step 2: This step calls for creating the SAP project’s counterpart in Microsoft Project. Here the composite application calls the Impress synchronize project web service to synchronize Microsoft Project with SAP for this specific project.

Once invoked, the Impress web service will not only map the SAP project data to Microsoft format, but also apply specific business processes and perform both validation and error handling to support Acme's business requirements.

Step 3: The project team within Acme that is responsible for the project is notified via email or via some sort of project dashboard that a new project exists and needs to be scheduled.

Step 4: Using Microsoft Project, the team determines resource requirements and schedule. The Project Status is set to schedule, which triggers the last step of the process.

Step 5: The composite application calls the Impress web service to send relevant project updates performed in Microsoft Project over to its counterpart in SAP. That way every project stakeholder is on the same page. In this case, because the project is defined in much more granularity in MS Project, Impress first aggregates information – such as costs and dates – according to Acme's business requirements before mapping and transferring the data to SAP PS.

4.2 Impress as a Service Consumer

From a historic perspective, Impress packaged integration applications have “consumed” application services for years via back-end system APIs. These interfaces are now evolving to support web service standards and Impress is ready to take advantage of these services as they become available. Case in point--Impress for EPM 3.1 will interact with MS Project 2007 via its web services interface (PSI). In Figure 3 below, Impress for EPM consumes both web services and API-based services.

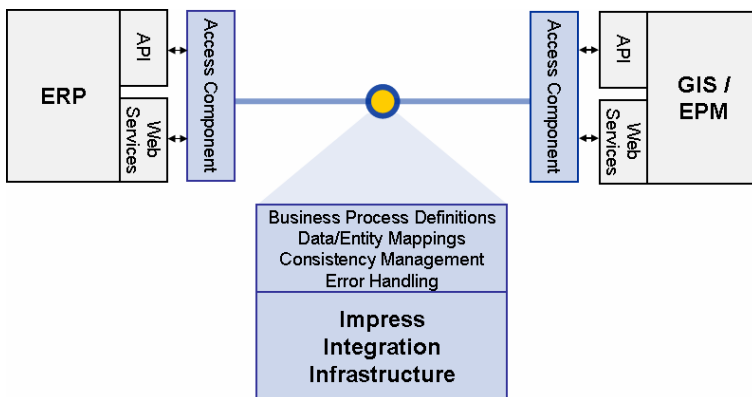


Figure 3 - Impress as a Service Consumer

As a Service Consumer, Impress interacts directly with back-end systems. This allows Impress to optimize performance while maintaining the tight end-to-end control over the integration process needed to perform functions such as data validation and error recovery that help ensure transactions are completed reliably.

4.3 The Road to SOA Compliance

Impress solutions comply with Service Oriented Architectures as both a Service Consumer and a Service Provider. More specifically, Impress includes the following SOA-specific features:

- Provides out-of-the-box web services for business processes such as invoking a data synchronization between systems
- Provides wizard-based tools to create new web services based on any Impress process
- Leverages the web service interface of external applications (e.g. Microsoft Project 2007 web service interface) as they become available and mature

Impress Software's intent is to use its partners' SOA-compliant services instead of proprietary APIs when these services become available with sufficient performance and stability to support its customers' performance and reliability requirements.

5 Conclusions

With the emergence of important technology standards, IT organizations are now closing in on the benefits promised by a Service Oriented Architecture. At the heart of SOA is the web service standard, which defines the standard communication mechanisms for linking software components together. To fully support a SOA, a commercial software application, such as Impress for GIS or Impress for EPM, must not only be designed to support web services as both a Service Consumer and a Service Provider, but also must provide IT with the ability to convert any business process, even highly configured or customized processes, into web services. Falling short of these capabilities - web service consumer, web service provider, and web service creation - an application will hamstring a company's ability to fully realize the benefits of a SOA.

Impress supports SOAs as both a consumer and provider of services. As a Service Consumer, Impress accesses back-end systems through both web services and traditional APIs. Impress's intent is to interact with back-end systems via web services when such services are made available by vendors and are robust enough to support Impress's commitment to performance and reliability. As a Service Provider, Impress provides several common business processes, such as "synchronize a project between SAP and Microsoft Project", as ready-to-use web services. And, perhaps more importantly than providing out-of-the-box web services, Impress also provides IT with the ability to convert any Impress business process into a web service for use by other composite applications.

Impress fully supports SOAs today and will continue to leverage SOA-related technology standards as they become available and are robust enough to support Impress's commitment to provide solutions that can handle complex integration scenarios and meet the performance and reliability demands of its customers.

6 About Impress Software

Impress Software is the leading provider of packaged integration applications that enable quick and cost-effective integration of project management and geographical information systems with SAP, resulting in streamlined business processes for asset-intensive organizations. Impress solutions allow integration to be completed at a fraction of the time and cost compared to custom integration solutions. Impress Software customers include global leaders such as Bayer, BP, DuPont, Haliburton, Valero, and AOL