

Teamcenter's Service Oriented Architecture

SIEMENS

White Paper

**Integrating Teamcenter-managed product design and
development with your mainstream business processes**

Today's companies are discovering they can more flexibly perform their business processes with less IT intervention if they use a service oriented architecture (SOA) as the technological foundation for their digital product design and development environment. Teamcenter® software's SOA services provide an open, high-performance, coarse-grained interface to the Teamcenter system's Business Logic Server. These services enable you to access Teamcenter-driven capabilities and integrate them with your existing business processes – as well as to create customized, task-specific programs to meet your business needs.

Contents

Executive summary3

 Why a service oriented architecture is important.....3

 What you can do with Teamcenter's SOA services.....4

 Gaining Teamcenter's SOA advantage.....4

Introducing Teamcenter's SOA5

 What is a service oriented architecture?.....5

 A technical look at Teamcenter's SOA6

Using Teamcenter's SOA services9

 Logical deployment architecture9

 Teamcenter's SOA service areas 10

 Teamcenter's SOA service invocation..... 10

 Using Teamcenter's SOA client-side libraries 11

 Using Teamcenter's SOA from WSDL-based applications 11

 Supported configurations 12

A final evaluation13

Executive summary

The biggest challenge for today's corporate IT organizations is to provide improved real-time visibility into their company's business operations. IT leadership is especially concerned with managing the software application lifecycle and controlling its cost.

Research indicates that the vast majority of IT executives and other senior managers are convinced that a service oriented architecture (SOA) is the right technology approach for improving IT agility and lowering that portion of the IT budget (40 percent) associated with application integration. Increasingly companies are discovering that their business becomes more nimble and their processes become more flexible when they implement SOA as the technological foundation for key computing environments (such as the product design and development environment). Equally important, SOAs are able to facilitate these improvements with less IT intervention than was previously possible.

To address these rising IT requirements, Siemens PLM Software has developed an SOA infrastructure for Teamcenter, Siemens PLM Software's flagship software for facilitating product lifecycle management (PLM) and supporting the PLM-driven business processes adopted by its customer base.

Teamcenter is industry's most widely used PLM solution with an integrated suite of applications that leverage technologies such as 3D visualization, community collaboration, supplier management and collaborative product data management (cPDM) to drive business productivity.

Teamcenter's SOA services provide a new coarse-grained interface to Teamcenter's Business Logic Server. Teamcenter's SOA services represent Siemens PLM Software's go-forward strategy for enabling client and external applications to access a Teamcenter-managed environment. These services can be used to:

- Integrate Teamcenter capabilities into your existing business processes, as well as for creating custom, task-specific client programs.

- Embed Teamcenter-managed information and status results into existing clients, portals and executive dashboards.
- Integrate the Teamcenter portfolio and its application solutions with diverse external functionality, including CAD systems, collaborative design management tools (such as Teamcenter's own community collaboration capabilities), manufacturing planning solutions (such as Tecnomatix® software) and highly popular business applications (such as Microsoft Office). As a result, you can establish a single cohesive environment capable of supporting all of your digital lifecycle management initiatives.

Teamcenter's SOA is based on industry standards for Web Services, primarily adhering to WS-1 and other WS-x standards and best practices. SOA services also are accessible through numerous language-specific libraries that make it easy for programmers with C, C++, C# and Java experience to incorporate Teamcenter functionality into new and existing applications.

Why a service oriented architecture is important

Teamcenter's SOA services provide new capabilities that allow you to deploy higher performing and more scalable WAN-friendly and firewall-friendly applications in a Teamcenter environment. Because these capabilities adhere to industry standards and established best practices, Teamcenter services are now accessible to a wider development community.

In turn, wider accessibility increases the likelihood that more robust and comprehensive services will evolve at a faster rate. Similarly, the worldwide development community will be able to access and leverage these services without having to learn how to use a proprietary language or protocol; no special training is required.

What you can do with Teamcenter's SOA services

Teamcenter's SOA is a coarse-grained API that openly exposes Teamcenter's Business Logic Server capabilities to Web Services, as well as to language-specific programs. This provides an ideal solution for enabling a wide range of applications to access your Teamcenter environment's product design and development information.

Teamcenter itself functions as the engine and repository that connects all of your environment's design and product development information in a flexible and loosely coupled manner – while providing your entire global environment with a single point of access to these connected assets.

Because Teamcenter's SOA services are exposed as WS-1 compliant WSDL¹ and language-specific interfaces, project teams are able to leverage an interface that best meets their own needs while assuring that the Teamcenter operations being called are the same – and that their results will be the same – regardless of the language or binding being used. In addition, Teamcenter's SOA easily supports

batch, or non-interactive applications, as well as highly interactive end-use applications on the desktop.

Gaining Teamcenter's SOA advantage

Organizations that adopt Teamcenter's SOA gain the advantages inherent in the use of a consistent, comprehensive set of interfaces that leverage Teamcenter's powerful product design and development services. Regardless of what language you use to call these services (and that choice is entirely up to you), the business logic and results received from Teamcenter's SOA will be the same.

You no longer need to worry about your project teams' skill sets or any language-related dependencies. Teamcenter's SOA provides a single set of entry points for use by all clients and applications – regardless of their heritage.

As new services are exposed via Teamcenter's SOA, they will be immediately available in all supported programming languages, as well as through industry-standard WSDL.

Introducing Teamcenter's SOA

What is a service oriented architecture?

A service oriented architecture is a system design pattern that emphasizes the following attributes – along with other considerations:

- Appropriate functional granularity appropriate to the task at hand
- Separation of the interface definition from its implementation
- Support for the concept of service providers and service consumers who are distinct from each other
- Standards-driven compliance

Service-oriented architectures also reflect a specific:

- Software framework
- Set of policies
- Set of best and accepted practices

Teamcenter's SOA embodies all of these principles and characteristics. It is built upon a framework consisting of the entire client and service infrastructure that contributes to consistent, reliable and high-performance interaction between the application clients, service consumers and Teamcenter Business Logic Server (the service provider in this scenario).

This framework enables application developers to concentrate on creating specific business functionality rather than worrying about the low-level communications and data management activities going on beneath them.

Teamcenter's SOA characteristics reflect two additional factors:

- Messaging via standard protocols
- Message content formats

Like the implementations of most service oriented architectures, Teamcenter's SOA uses standard HTTP/S

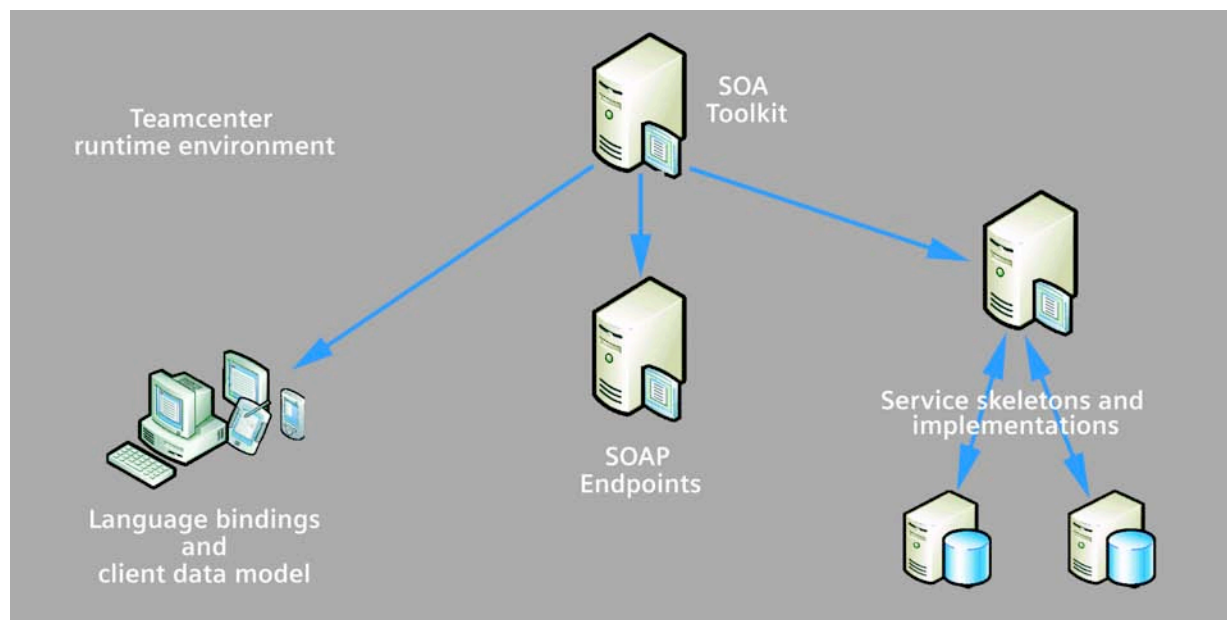


Figure 1: Teamcenter SOA framework components.

communications protocols to send XML documents back and forth between the service provider and the service consumer. Use of these standards contributes to the openness, flexibility and scalability necessary for both the local and global deployment of Teamcenter. It is this open, standards-based messaging environment that allows service providers and service consumers to be implemented in differing technologies while retaining their interoperability.

This messaging environment also allows existing applications in one technology to be quickly adapted so that they can interact with newer and richer technologies and programming languages – which helps your company preserve its investments in existing programs and processes.

As a result of this flexibility, you can incrementally enhance your product design and development environment at a speed that virtually matches your business' real-world evolution. This alignment is largely facilitated by the arms-length, loosely coupled relationship that exists between service providers (like Teamcenter's Business Logic Server) and service consumers (like Teamcenter clients, CAD systems, Microsoft Office and your own portal and dashboard applications).

A technical look at Teamcenter's SOA

Teamcenter's SOA consists of the following client-side and server-side components that follow well-defined patterns using standards-based protocols, languages and data formats.

- SOA Framework provides the communications and infrastructure functionality that enables application developers to concentrate on creating business solutions rather than dealing with underlying communications/messaging issues
- SOA Language Bindings provide language-specific functions (.NET, C++, and Java) used by service consumers to invoke an SOA service and properly handle the ensuing response
- SOA Client Data Model (CDM) and Data Model Manager provide managed type-safe storage on the service consumer side for all data returned by service responses from Teamcenter's Business Logic Server
- SOA Toolkit provides tools that enable Teamcenter developers to auto-generate all of the artifacts necessary for exposing new Teamcenter business logic as SOA services

In addition to these major components, Teamcenter's SOA environment also includes documentation on each of the language bindings, WSDL and various XML Schema Definitions (XSDs) used throughout the system.

The relationships between the environment and its major components are illustrated in Figure 2.

The accompanying table describes the major components in Teamcenter's SOA in more detail.

Teamcenter's SOA's major components

Capability	Function
SOA Framework	Consists of the client-side libraries, as well as the communications infrastructure and server-side libraries that make up a full-featured request/response pipeline. All SOA Framework components are auto-generated by the SOA Toolkit and are completely opaque to both the author of the service and consumer of the service.
SOA Language Bindings	Represents the client-side components required to execute a service request from an application. Because the SOA Toolkit auto-generates these components, the components are functionally equivalent across all supported languages. Initially, Siemens PLM Software provides bindings for C++, .NET and Java.
Teamcenter's Data Model Manager	<p>Populates and updates the client-side data store with objects returned from Teamcenter's Business Logic Server on each SOA service call. Object identity is maintained so that when an existing object is returned, its properties are updated in the data store rather than creating a new/identical item.</p> <p>Data Model Manager is also responsible for maintaining the meta-model of type and property information. Optionally, it can fire events on object creation, deletion or modifications. Client applications can register listeners for these events and react to them as appropriate.</p>
Teamcenter's Client Data Model (CDM)	<p>Generated from Teamcenter's Data Model formally defined within Teamcenter's Business Modeler/IDE, Teamcenter's SOA supports two types of CDM for client application use – a generic model and a type-safe model.</p> <p>The generic model consists of simple objects and their properties. Utility functions are provided to determine an object's Teamcenter type, to query the type hierarchy and to access the characteristics of each property on an object.</p> <p>Derived from the generic model, the type-safe model adds strongly-typed accessors for all objects and their properties. This model's class hierarchy matches Teamcenter's Business Logic data model formally defined in Teamcenter's Business Modeler/IDE. The accessors for each property return the language-specific equivalent of the native Teamcenter type of property.</p>

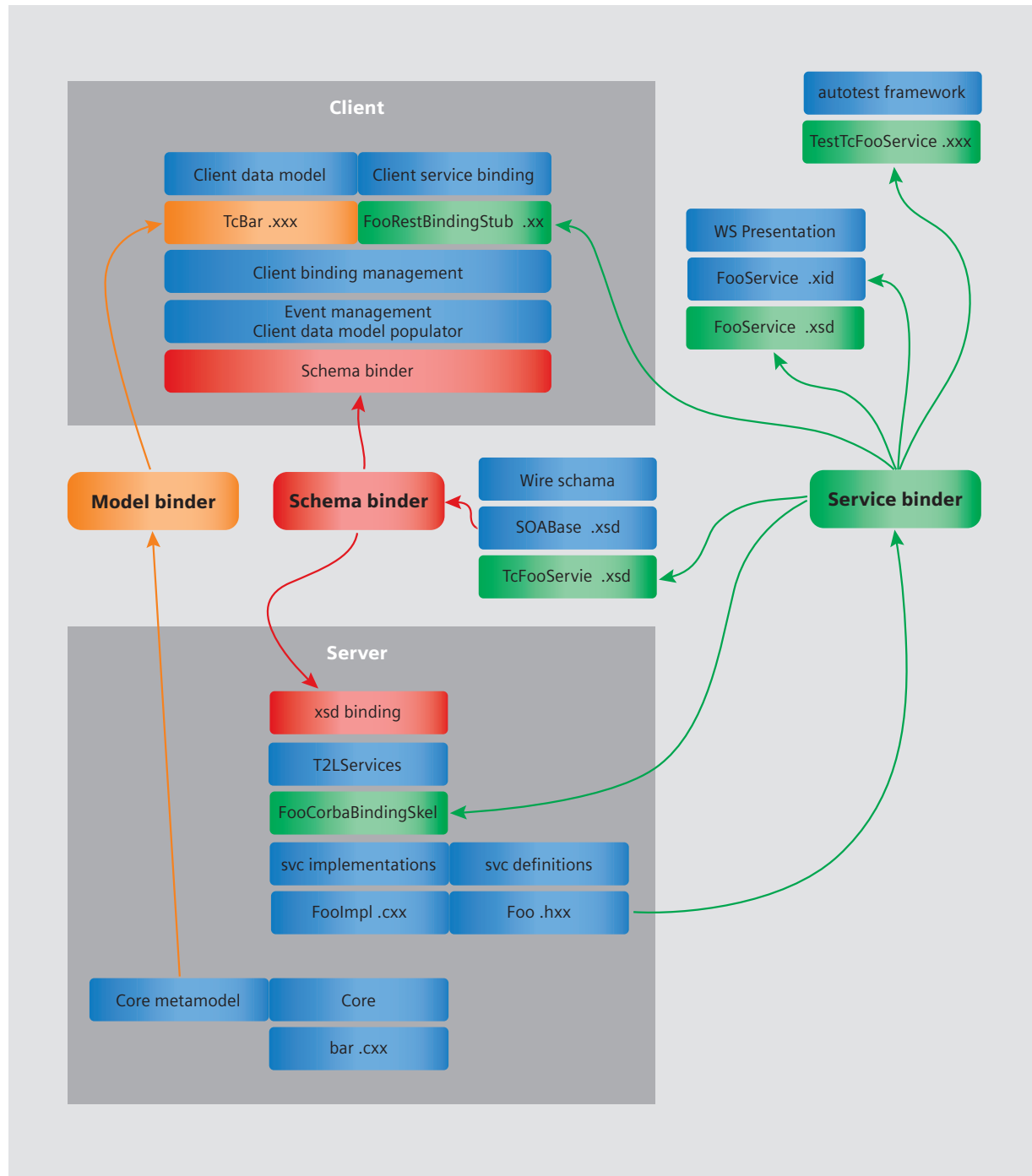


Figure 2: Teamcenter SOA functional components.

Using Teamcenter's SOA services

Logical deployment architecture

Organizations typically deploy the Teamcenter environment using a four-tier logical architecture that consists of:

- Client application
- Web application server
- Enterprise Tier
- Resource Tier

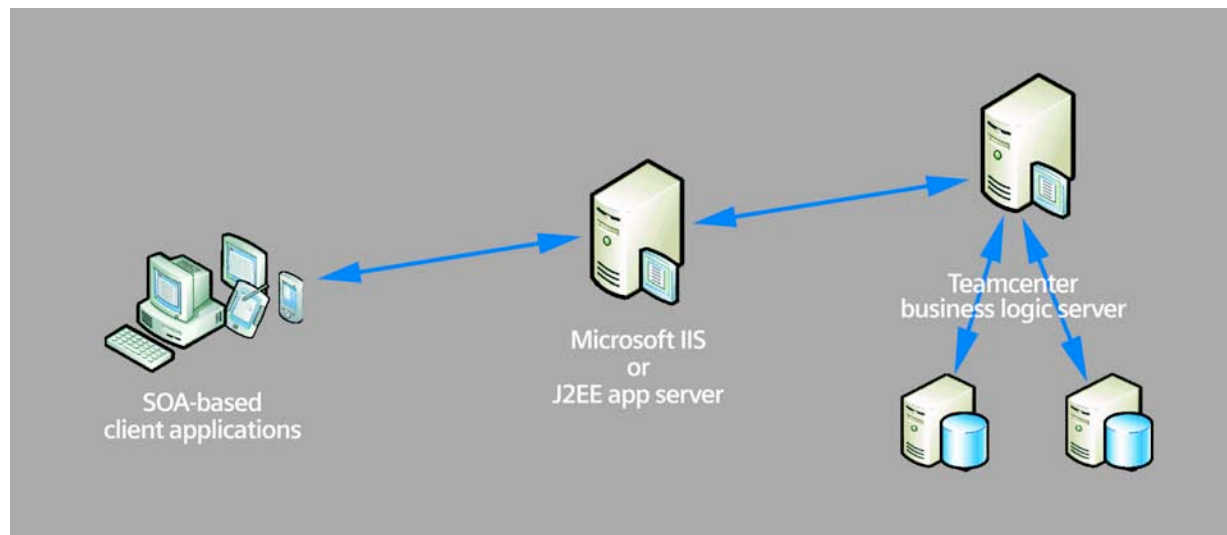


Figure 3: Teamcenter SOA runtime deployment.

Client application Most client applications, including applications developed by Siemens PLM Software and its partners, are language-based, rather than WSDL based. As a result, they typically incorporate one of Teamcenter's SOA client libraries along with the CDM. Custom integrations developed using the WSDL approach usually do not use client libraries or the CDM. Since both types of clients are HTTP/S based, they can easily and effectively access Teamcenter across low or high latency network connections – as well as securely traverse corporate firewalls without needing to open additional application-specific ports in those firewalls.

Web application server Teamcenter deployments use web application servers to expose the SOA service endpoints for all client types. Both REST-style (standard HTTP POST) and SOAP-style requests are supported by Teamcenter components executing on industry-standard application servers, such as Microsoft IIS, as well as servers based on Java and J2EE technology. SOA Framework components running on these application servers are responsible for normalizing the request into a single common format, which is passed directly onto Teamcenter's Business Logic Server that resides in the Enterprise Tier.

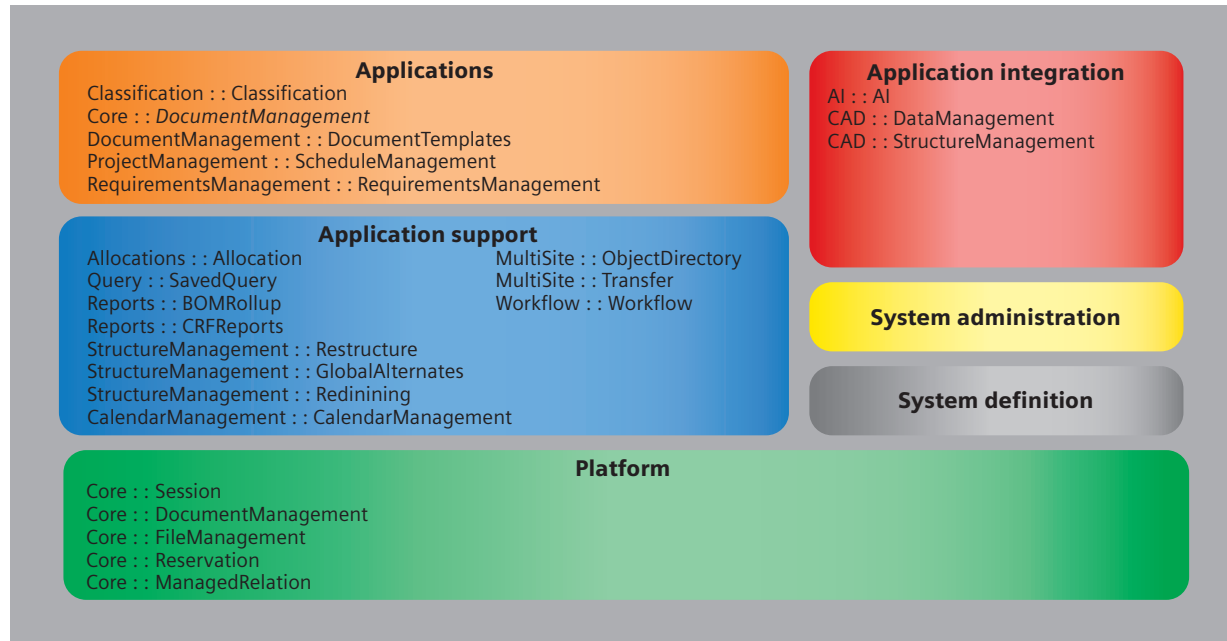


Figure 4: Teamcenter services functional areas.

Enterprise Tier Teamcenter's Business Logic Server – along with Teamcenter's SOA server-side components, resides in the Enterprise Tier. All of the actual SOA services are implemented in C++ as part of Teamcenter's Business Logic Server.

Resource Tier The Resource Tier houses the Teamcenter database and bulk-data file repositories. No SOA components are part of the Resource Tier.

Teamcenter's SOA service areas

Teamcenter's SOA services are organized into the following six functional areas.

- Platform
- Application support
- Applications
- Application integration
- System administration
- System definition

Each of these functional areas is further broken down into one or more libraries that contain the actual services and operations exposed by Teamcenter's Business Object Server. The accompanying diagram indicates how the six functional areas are organized,

as well as the representative library and service names that pertain to key areas.

As the diagram illustrates, services are provided for fundamental needs (such as session and file management), for supporting capabilities (such as queries and reporting), and for specialized services (such as PLM system definition and administration). Over time, Siemens PLM Software will introduce additional libraries and services for each of these areas – especially in the application and application support areas.

Teamcenter's SOA service invocation

Before discussing how Teamcenter's SOA services are invoked, it is helpful to understand some of the basic principles that drive the design of Teamcenter's SOA services.

First, Teamcenter's SOA provides set-based services – meaning that virtually all services accept multiple input objects that can be operated on by the Business Logic in a single call. On both inbound and outbound sides of a request/response cycle, it is advantageous to be able to submit or respond with more than one item at a time. This flexibility is especially important when invoking services over high-latency wide-area network connections.

Adopting a set-based default design for all services also supports a loose-coupling and infrequent communications pattern. At the same time, it reduces the overall size of the service set by eliminating the requirement for both single item and multi-item interfaces for the same operation. A set-based operation can easily handle a set-of-one without requiring a separate interface.

Second, Teamcenter's SOA services are coarse-grained services – meaning that they perform operations that, from a developer and end-user perspective, are relatively complete in and of themselves. The service consumer asks for something to be done and Teamcenter's Business Logic Server does it. Teamcenter's SOA services eliminate intermittent network chatter that plagues many ordinary client/server applications.

Siemens PLM Software designed Teamcenter's SOA services so that the response to a user operation (for example, highlighting a set of objects or hitting the "delete" button) is performed in a single request/response cycle. In other architectures and system designs, an operation like this might take hundreds of request/response cycles to individually submit and acknowledge the deletion of each highlighted or selected object.

Third, Teamcenter's SOA services support the "partial errors" concept – meaning that a set-based operation can be only carried out on a subset of the input items when an error condition arises. Again, drawing on the "delete" button example, if some of the items selected for deletion are locked or have outstanding dependencies or relationships, it often is neither wise nor possible to delete these items as requested. In these cases, it is important to tell the service consumer which items were not operated on and the reasons why. Efficient and consistent partial error handling is a crucial principle that drives Teamcenter's SOA Framework and Business Logic Server – as well as the client libraries and data model.

Using Teamcenter's SOA client-side libraries

Regardless of what specific language binding is used in the application, developers perform the following consistent steps when they leverage Teamcenter's SOA client-side libraries.

Connection The client application first requests a connection object, which contains the parameters that can be used to connect to the server. The SOA

client framework instantiates the connection object and returns it to the application – where subsequently it can be used to invoke appropriate services.

Authentication Since Teamcenter is a secure environment, any service invocation must be preceded by an authentication request. Successful authentication returns a token that is used on all service requests across the established connection. Teamcenter's SOA supports all Teamcenter authentication models, including single sign-on (SSO) and synchronized LDAP.

Service invocation Before invoking a service, the client application must build the necessary input structures required by the service, as well as set parameters on each item being input as part of the set-based call. Once all of the data has been properly structured, it is passed to a service stub for submission to the Business Logic Server.

Over-the-wire The service stub works in conjunction with the SOA Framework to marshal the service request into a REST XML document and sends it over HTTP/S to the Web Tier.

Web Tier SOA Framework forwards the request to an available/assigned instance of the Business Logic Server.

Enterprise Tier SOA Framework forwards the request to the appropriate services skeleton, where the XML document is un-marshaled. Then, the service endpoint or implementation is called and the service's business logic is executed. Once the business logic's execution is complete, the service skeleton marshals the results (including any information on exceptions or partial failures) to a REST XML document and passes it back through the Web Tier.

Web Tier SOA Framework receives the service response and properly formats it for an HTTP/S POST response.

Client framework The service stub un-marshals the response, uses the SOA Framework to process it into the Client Data Model (including any error information), and returns control to the client application.

Using Teamcenter's SOA from WSDL-based applications

With just a few adjustments to the components, you can use the same set of steps described above (in the "Using Teamcenter's SOA client-side libraries" discussion) to invoke an SOA service from a

WSDL-based web services application. These adjustments include the following differences.

First, WSDL-based applications generally do not make use of the Teamcenter client-side libraries. Instead, they use utility functions within each of the major web services toolkits to automatically build the necessary connection objects and input/output structures for a service directly from the Web Services Description Language (WSDL). As you might expect, tools such as Microsoft's Visual Studio can greatly assist the developer by reading the WSDL for a service and auto-generating the necessary artifacts transparently.

Second, the only other difference between the REST and SOAP paths is the entry point into the Web Tier. Unlike the REST entry points (described earlier), the SOAP request is processed by the SOAP engine on the Web Tier. The SOAP engine un-marshals the SOAP XML document and forwards it to the Web Services endpoint that is auto-generated by Teamcenter's SOA Toolkit.

At the service endpoint, the request is marshaled to a REST XML document and routed to Teamcenter's Business Logic Server in the same manner as the REST example described earlier. Information about whether the request came through a SOAP endpoint or a REST endpoint is never passed to the Business Logic Server. This practice helps ensure that service innovations are always processed in the same manner regardless what type of client invokes the request.

With respect to the service response, the Web Service endpoint returns control to the SOAP engine, where it is properly formatted for a SOAP-over-HTTP response.

Once the response is received, utilities in the web services toolkit (i.e. in the toolkit is being used to build the client application) un-marshal the response into structures directly usable by the client application. Since the Client Data Model and the Data Manager Model are part of the SOA Framework, no additional processing is necessary for handling the business objects returned on the response.

Supported configurations

Teamcenter's SOA services are fully supported on all standard four-tier Teamcenter deployments as discussed throughout this white paper. In addition, Teamcenter's SOA services also are supported in Teamcenter's "two-tier" configuration sometimes used by Teamcenter users migrating up from Teamcenter Engineering.

Two-tier deployments combine the Teamcenter client and a Business Logic Server instance on each end-user's computer; the shared database and file stores remain on a centralized resource tier. Typically, this configuration uses CORBA/IIOP communications rather than HTTP/S and eliminates the Teamcenter Web Tier altogether. As a result, client implementations in this type of deployment must use one of the supplied language binds rather than a WSDL-based client implementation.

A final evaluation

By applying an effective SOA to their PLM processes, companies can support more business capabilities, reduce IT complexity and accelerate IT implementation – as well as re-use more applications (via web services) and better align their PLM initiatives with other business activity.

Teamcenter's SOA services provide a robust, flexible and highly-scalable interface into your digital design and development processes and their related information assets. The SOA's secure, WAN-friendly and firewall-friendly attributes are especially appropriate for meeting the most rigorous demands of both large and small companies.

On the technical side, Teamcenter's SOA is a state-of-the-art implementation that is based on – and compliant with – the latest industry standards for web services interoperability. By adopting Teamcenter as the backbone for your digital enterprise, you can immediately reap the benefits of web services technology through an infrastructure that

facilitates lower cost of ownership, as well as greatly improved integration between your business systems and the technical systems that power your product design/development initiatives.

Siemens PLM Software has initially focused Teamcenter's SOA on the robust capabilities and high performance required to deliver a core architecture and service-based design capable of supporting both large and small deployments. The services exposed by Teamcenter's SOA cover a broad range of Teamcenter functionality with particular attention to streamlining your product design and development processes.

Future releases of Teamcenter's SOA will add value to your organization by exposing additional Teamcenter functions, supporting more language bindings, invoking additional tiers in the Teamcenter stack and facilitating interoperation with other external application offerings as these programs converge toward accepted industry standards.

References

1. Web Services Description Language (WSDL).

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with nearly 6.7 million licensed seats and 63,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

Siemens PLM Software

Headquarters

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
972 987 3000
Fax 972 987 3398

Americas

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
800 498 5351
Fax 972 987 3398

Europe

3 Knoll Road
Camberley
Surrey GU15 3SY
United Kingdom
44 (0) 1276 702000
Fax 44 (0) 1276 702130

Asia-Pacific

Suites 6804-8, 68/F
Central Plaza
18 Harbour Road
WanChai
Hong Kong
852 2230 3333
Fax 852 2230 3210

www.siemens.com/plm

© 2010 Siemens Product Lifecycle Management Software Inc. All rights reserved. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Geolus, GO PLM, I-deas, Insight, Jack, JT, NX, Parasolid, Solid Edge, Teamcenter, Tecnomatix and Velocity Series are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks used herein are the property of their respective holders.