Service Component Architecture in CICS

March 2010

Service Component Architecture (SCA)

Document version 1.0

Fraser Bohm
Senior Software Engineer, CICS Development

Exec Summary

This paper sets out the advantages to be gained from adopting a SOA approach to CICS applications through the use of the Service Component Architecture (SCA) technology included in CICS Transaction Server v4.1 and provides an introduction to the concepts involved.

SCA offers a framework that provides the flexibility to build and extend modern applications in a Service-Orientated Architecture (SOA). SCA supports rapid deployment of new applications to meet changing business requirements by promoting the reuse of existing application assets in a component model. CICS supports the Open SOA Collaboration (osoa.org) SCA 1.0 programming model to support service composition and deployment in a SOA environment.

The value proposition of SCA, therefore, is to offer the flexibility for true composite applications. These applications flexibly incorporate reusable components in an SOA programming style. The overhead of business logic programmer concerns regarding platforms, infrastructure, plumbing, policies and protocols are removed, enabling a high degree of programmer productivity.

Some of the key features provided by SCA are:

- Decoupling of application business logic from the details of its invoked service calls and it's method of invocation
- Language neutral – callers can be isolated from the language details of the services they invoke

- Declarative Meta-data for consistent application description which allows power modern tooling support. Tooling in support of CICS & SCA available in RDz 7.6

- Bindings (the way in which components communicate) can be optimised where deployments allow or use Web Services for inter-operation with remote services
- The ability to declare (outside of business logic) the Quality of Service requirements, such as Security, Transactions and the use of Reliable Messaging

SCA provides a powerful means of reducing complexity for the programmer creating an applications business logic whilst allowing a great deal of flexibility and richness in the customization of the applications behaviour when deployed or changed.

Introduction

CICS support for SCA in CICS Transaction Server v4.1 allows exiting application logic to be expressed as SCA Components and new application logic to be built using SCA components. SCA supports rapid deployment of new applications to meet changing business requirements by promoting the reuse of existing application assets in a component model.

CICS supports the Open SOA Collaboration (osoa.org) SCA 1.0 programming model to support service composition and deployment in a SOA environment. This allows CICS to support SCA components and composites. Service components can be assembled to form *composites*. A composite is the unit of deployment in SCA and is described in an XML language called SCDL. Composites can contain components, services, references, property declarations, and the wiring that describes the connections between these elements. Composites can also be used as components within other composites, allowing for a hierarchical construction of composite applications, where high-level services are implemented internally by sets of lower-level services and additional business logic.

SCA in CICS allows an existing CICS application to be described as one or more SCA components which provide Services. The bindings used by these components can then be chosen to allow for a wide range of connectivity patterns including direct binary access or full Web Services. These components can then be reused in new applications, still allowing for different bindings, without change to the underlying application code.

SCA in CICS

SCA allows an application to be described as a collection of components which offer services and make use of other services. The need for an application to make use of another service is described in the component as a reference. Components can then be assembled in Composites which can

either be deployed or reused.

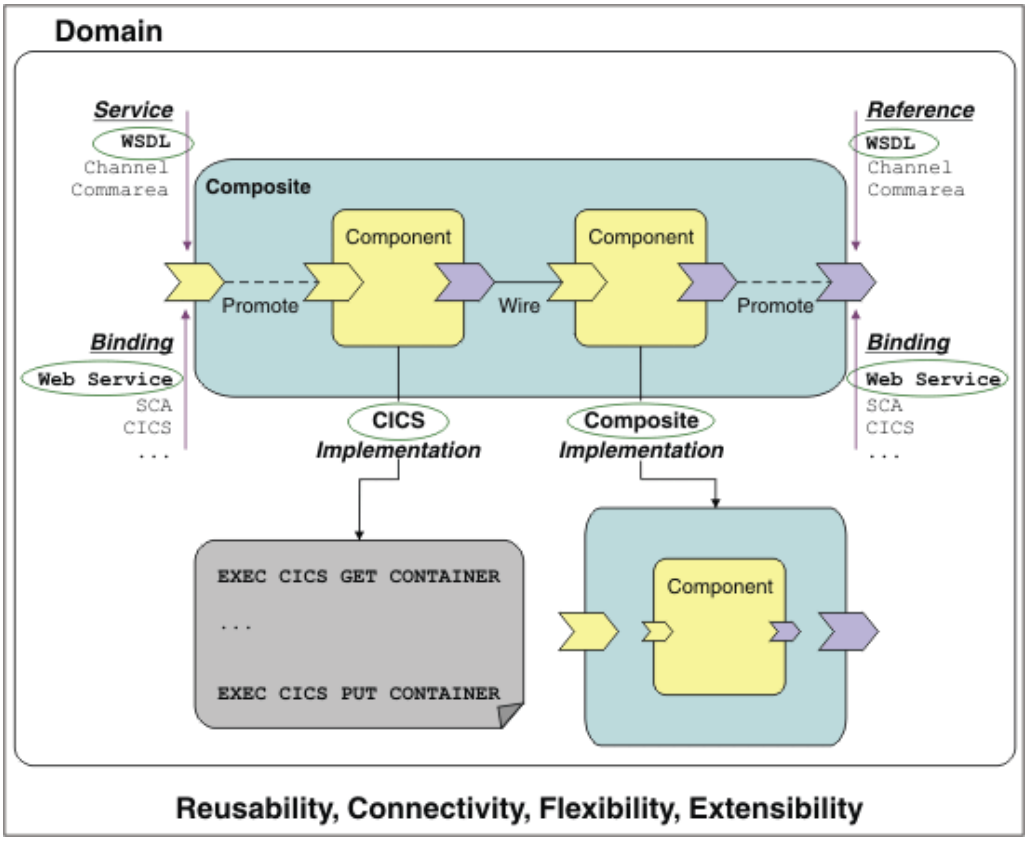

Fig 1 : Overview of SCA artifacts

CICS has a strong heritage allowing application programmers to build reusable business components through technologies such as Dynamic Program Link, Web Services and Service Flow Feature. This means that many of the concepts of SCA has existing natural analogies in the CICS world.

In terms of providing the business logic that implements a component CICS has the PROGRAM resource and artefact. A component in CICS can be implemented by a root program or by another composite which models a collection of programs that communicate via SCA bindings.

For Services provided by components in CICS we draw upon the support for Web Services introduced in CICS Transaction Server v3 to allow CICS Program interfaces to be described with WSDL and XML Schema. Any program that can be exposed as a Web Service can be described as an SCA Service.

CICS applications that make use of Services via the Invoke Web/Service command can have a Reference added to the component that flags this usage and allows for SCA bindings to be used to control how that request is satisfied. This allows options such as protocol, transport, message mapping and custom processing to all be configured without change the application.

The tooling support for SCA in CICS is provided in Rational Developer for System z 7.6(RDz).

IBM support for SCA

SCA support is present in a pre OSOA form in the WebSphere family of products, most notably WebSphere Process Server, tooled through WebSphere Integration Developer (WID) and the IBM WebSphere Application Server v7 Feature Pack for Service Component Architecture (SCA). The Feature Pack provides support for the OSOA SCA 1.0 programming model and tooling support in Rational Application Developer (RAD) 7.5.2.

Although these offering are aimed at different types of component implementations the SCA programming model provides a consistent set of concepts that allow a common understanding to be shared between developers working in both WebSphere and CICS Transaction Server.
The tooling provided for both CICS and WebSphere has a consistent interface customized for the two environments.

SOA on in the inside – Modern & Flexible

Web Services and ESB technology have provided a proven method of embracing SOA at the edges of the application domain. They deliver key benefits in terms of interoperability, governance and productivity. SCA provides a means to realise these benefits on the inside of an application domain without introducing the over heads of a full Web Services stack for each interaction. SCA Bindings are a mechanism by which applications can be

isolated from the specifics of how they are invoked and how they communicate with other application components.

In CICS we support the two standard bindings defined in SCA, the Web Services binding and the SCA binding. We also support a CICS specific binding to allow CICS to CICS communication. In addition to allowing the communication mechanism to be selected appropriately for the deployment these bindings allow the CICS Pipeline support to insert infrastructure components, such as audit or security, between application component calls. This provides a way to clearly demarcate and enforce a separation of concerns and responsibilities between application and infrastructure which further simplifies the application developers tasks and allow greater flexibility in infrastructure changes without materially impacting the applications.

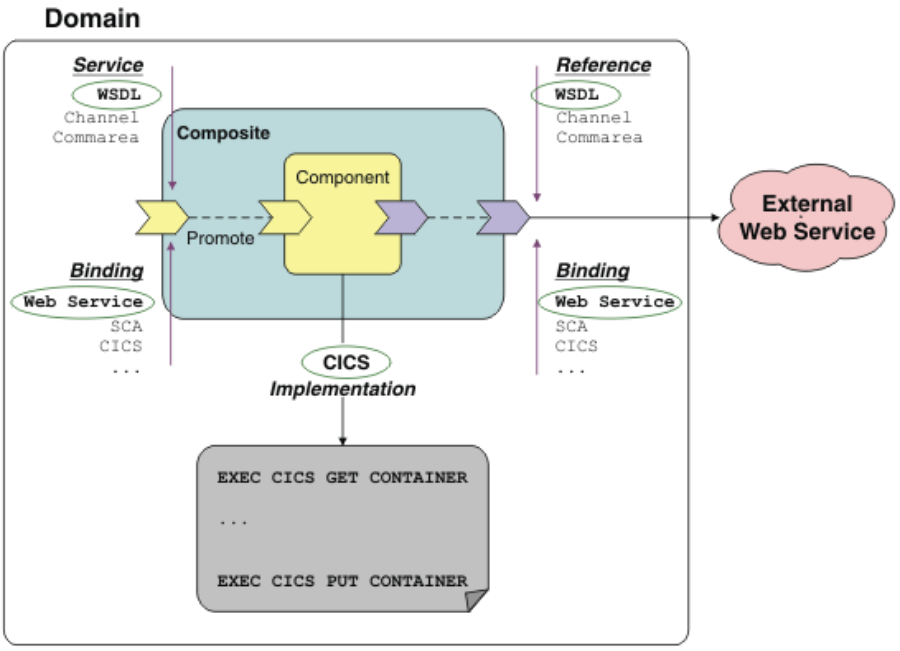

Fig 2: a simple CICS SCA component which makes use of an external Web Service using the web services binding.

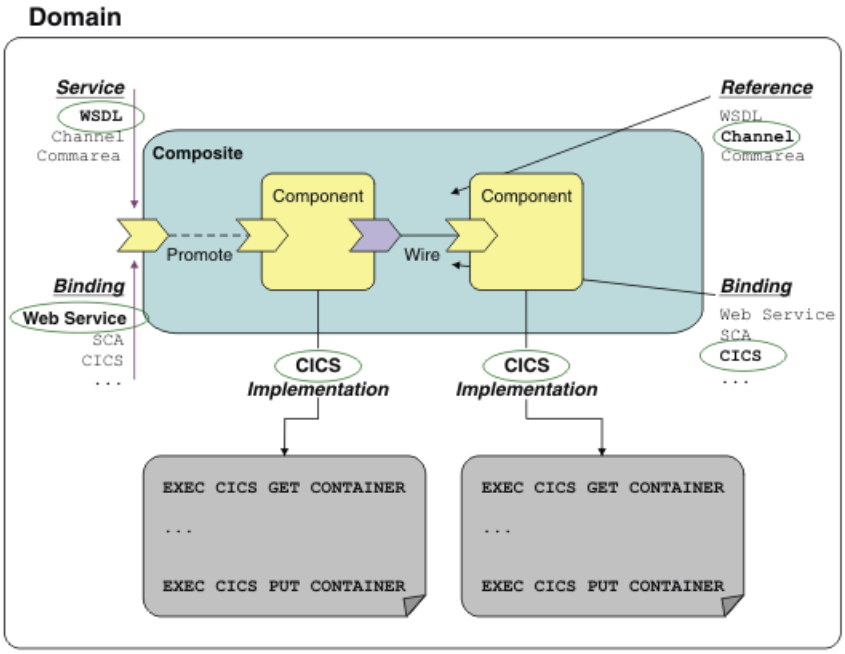

Fig 3: the same component as in Fig 2 but with the external service now implemented in CICS as a channel linkable program and bound to the original component with the CICS binding.
consideration is given to reuse of components or applications. An external Wev Service can be seamlessly replaced by an internal CICS implementation without change to the invoking application component. The transport can be optimized to use CICS internal services and data mapping can be avoided whilst still passing the CICS Channel to hold application and context data.

The support for SCA in CICS provides a broad set of foundational capabilities that can be built on to provide a modern, flexible, application platform that encourages reuse of existing assets.

SCA in general provides for even more flexibility through such features as additional bindings, implementation types and policy support. CICS Transaction Server v4.1 provides the core foundational aspects of the SCA 1.0 programming model and the roadmap aims to broaden these capabilities in line with the maturity of the technology and the needs of the enterprise.

Reducing Required Skills

The focus provided by SCA on a consistent language-neutral application model, declarative interfaces and standard bindings can help reduce the skills need to

create new and reuse existing applications and increase programmer productivity. This provides a benefit by elevating the tasks an application programmer has to do from low-level coding to application assembly and composition, ably supported by rich tooling.

The commonality of the tools and concepts between the IBM SCA offerings and indeed other SCA offerings, allows for the development of common understanding whilst still exploiting the specific environment and language skills available.

Declarative Applications

The SCA support in CICS exploits another new capability of CICS Transaction Server v4.1 for SCA application deployment and management. The intoduction of the Bundle resource and associated tooling allow a SCA application to declaratively express its resource dependencies and to be layered so that it can be deployed in multiple environments or versions.

The Bundle support allows all artefacts created when a given set of application components is deployed to be queried and managed as a single aggregate entity. The ability for these composite applications to express their dependencies, such as the need to access a particular file or web service, allows more robust checking both at deployment time and while managing the operations of the system. This capability can be extended by customers to support deployment specific resource types as well which can then be managed seamlessly alongside the associated CICS resources.

Practical Application

A detailed example of how the concepts and technology described here can be exploited in CICS can be found here:
https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/topic/com.ibm.cics.ts.scenarios.doc/topics/sca_scenarios.htm
This example details how the CICS Catalogue Example Application can be exposed and extended through SCA.

Summary - Conclusion

We have discussed the advantages to be gained from adopting a SOA approach to CICS applications through the use of the Service Component Architecture (SCA) technology included in CICS Transaction Server v4.1 and introduced some of the relevant concepts of SCA.

We have shown that SCA offers a framework that provides the flexibility to build and extend modern applications in a Service-Orientated Architecture (SOA) and how it supports rapid deployment of new applications to meet changing business requirements by promoting the reuse of existing application assets in a component model.

Some of the key features and value provided by the SCA technology in CICS are:

- Decoupling of application business logic from the details of its invoked service calls and it's method of invocation through the use of EXEC CICS INVOKE Service as the standard method of component to component communication.
- Language neutral – callers can be isolated from the language details of the services they invoke and the CICS bindings can be used to handle incompatibilities between the programmatic binary interfaces through the uses of XML Transformation.
- Declarative Meta-data for consistent application description supported by the use of the new Bundle resource and manifest file allowing application dependencies to be managed and multiple deployments of common application components.
- Support for both the standard SCA native and web services bindings and an optimised CICS binding for CICS to CICS component communication.

The SCA capability in CICS Transaction Server v4.1 provides a powerful means of reducing complexity for the programmer creating an applications business logic. The support offers the systems programmer a great deal of flexibility and richness in the customization of the applications behaviour when deployed or changed.

SCA can deliver real value by offering the flexibility for true composite applications.

IBM United States of America

Produced in the United States of America