Arcati Research Note

Making CICS a peer player in web services

by Mark Lillycrop, Arcati Research

Recent developments in CICS – and specifically the announcement of Transaction Server 3.1 – have redefined the role of the mainframe as a genuine peer player in web services, an essential change for the growing number of customers who view the zSeries as a strategic platform for new applications.

Despite the great enthusiasm for web services and so-called service-oriented architectures (SOAs), it's still true that most large organizations won't get far without the mainframe data and processes on which their business depends for survival. Consequently, the use of standardized middleware to allow mainframe systems to interact with web-facing applications is absolutely essential for the future development of many *Fortune 500* companies.

Those who have predicted the demise of the mainframe – many times over the last fifteen years – have clearly been proved wrong. Rather than sinking irrevocably into the corporate background, and waiting to be relieved of its workload by newer and more nimble technologies, the IBM mainframe has proved that it is irreplaceable, at least for the foreseeable future. Its capability as a highly scalable, highly available transaction processing engine remains to be challenged, and its role as the central repository of key corporate data is equally secure – with anything up to 80% of business-critical data still residing under MVS control.

One of the problems the industry has faced with the mainframe is that it has remained resolutely distinct from every platform that has come along since. Mainframes were always intended to run a complex mix of workloads, many of them batch-oriented, with very high utilization rates and particularly good performance for transaction-oriented environments. Distributed servers tend to offer much better performance for numerically intensive work and single applications, as well as dramatically lower utilization rates. As such, their value proposition is so fundamentally different that industry pundits (and indeed IT vendors and users) have spent the last decade arguing about the relative cost of ownership of chalk and cheese.

For some years now, the mainframe has been a very open platform and an early adopter of open standards. But with its distinct skill-set and complex architecture, many users have tended to position the platform as a separate technical entity, a back-end server in a world increasingly governed by web services and peer connectivity.

Having said that, it is definitely time to draw a distinction between legacy mainframe users (those who continue to rely on the system, either internally or via an external service company, for back-end data and processing, but who see no prospect of new applications for the large systems platform) and strategic mainframe users (those who wish to leverage the mainframe's strengths to build new web-facing business applications). In both cases, it's essential for the mainframe platform to offer some level of support for key web service middleware, but the level of interoperability required and the ease of web development will vary considerably between the two groups.

For the 'strategic' group in particular, there is a strong requirement to make the mainframe a peer player in web services, so that new applications can be created fast and positioned according to a rapidly changing set of business criteria. This means that the mainframe must be capable of requesting web services as well as serving data and transactional function. Without this two-way flow, users who see real benefit in leveraging the strengths of the mainframe for new systems may face real technical obstacles in making it happen.

CICS and web services

The success of the mainframe in the past, and the key to its future role in the large enterprise, comes down in no small part to CICS – the ubiquitous TP monitor. CICS reportedly processes some 30 billion transactions a day worldwide, not bad for a piece of software that has just celebrated its 35th birthday. Much of IBM's focus recently (and that of the ISVs) has been on making CICS ready for web services by developing its interoperability with the appropriate middleware components, particularly within the XML area.

The recent announcement of CICS Transaction Server for z/OS V3.1 attempts to position CICS much more favourably within the Web Services environment. The announcement focuses on three key areas – ease of integration; enhanced application transformation; and performance and system management. The third of these areas includes the usual fine tuning to please the systems programmers and some improvements to the CICSPlex SM Web User Interface. The interesting stuff comes under the first two headings, Integration and Application Transformation.

Integration

As IBM starts to put more meat on the bones of its on-demand computing strategy, its prime integration objective is to allow the re-use of CICS logic and processes within web services applications, by providing full access to standard APIs and communications protocols. This brings it one step closer to participating fully in those elusive service-oriented architectures (SOAs).

From IBM's perspective, delivering web services is all about support for XML and its associated middleware standards, WSDL and SOAP. WSDL (Web Services Description Language) is an XML-formatted language used to describe a Web

service's capabilities – basically a definition of the service. SOAP (Simple Open Access Protocol) is a lightweight protocol for exchange of information. The two standards have grown out of a joint development between Microsoft and IBM, and are widely deployed together. SOAP has been available as an add-on function to CICS for 18 months now, attracting some 800 users who are keen to use SOAP to help simplify the management of transactions and calls between CICS and the distributed world. In V3.1 SOAP becomes a standard feature, alongside the more heavyweight transport mechanisms such as WebSphere MQ.

The strongest message to come out of V3.1 is that CICS can act not just as a provider of services but as a requester too. This latter function is far less mature, as most of the industry's attention so far has been on unlocking the data and functionality of the CICS sub-system to allow distributed applications to tap into it. The requester capability opens up a new role for CICS, potentially allowing it to deploy web services in ways that will be of particular interest to the more *strategic* mainframe users. Among these users is Charles Schwab, long considered to be one of IBM's most influential mainframe customers, which was quoted at the announcement as saying, "We need IBM to enable CICS as a service provider and eventually as a consumer, and look forward to the day when CICS is fully Web Services enabled." This bidirectional requirement is directly addressed with the server/requester capability of CICS TS 3.1.

To achieve this, TS V3.1 includes a number of features for distributed transaction coordination, using WSDL to define the services. There is also a new tool, CICS Web Services Assistant, a build-time capability provided to create a WSDL document from a simple language structure or a language structure from an existing

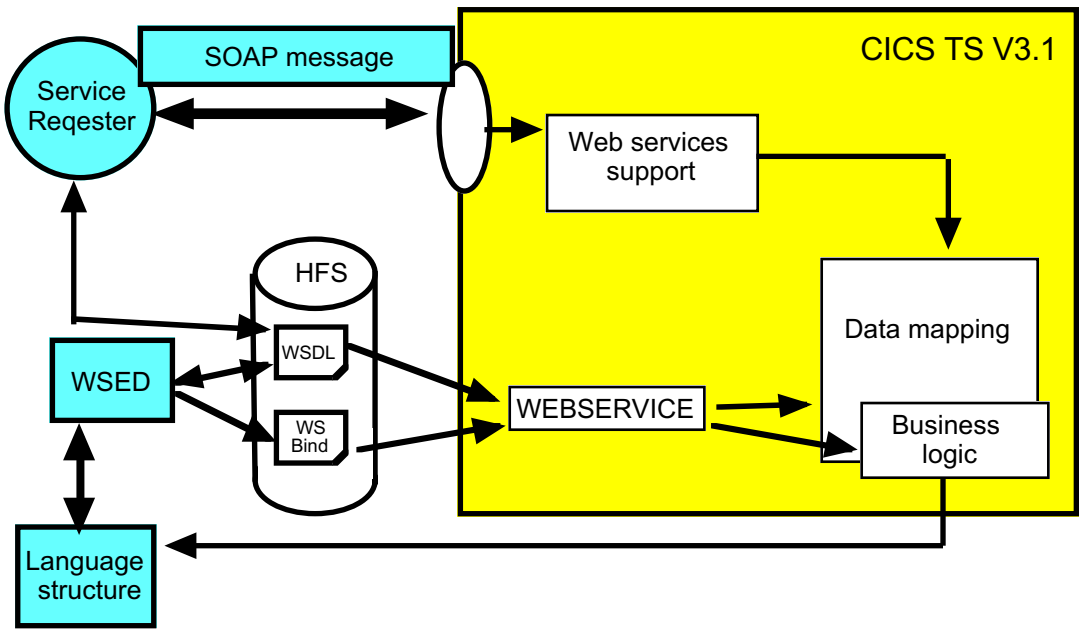

Figure 1: CICS TS 3.1 architecture. Source: IBM

WSDL document (with support for COBOL, C/C++ and PL/I).

Another enhancement to integration in V3.1 is improved HTTP support. Significantly, the new CICS version is compliant with HTTP 1.1, with Outbound support added – once again confirming the role of CICS as both server and requester. Further new web service functions are persistent sessions (as a default for interactions between CICS and a remote partner) plus support for pipelining and chunking of messages.

The third key component of web service integration is security. As well as support for Secure Sockets Layer (SSL) 3.0 over IP, the new technology offers a range of key functions that will allow closer control of security within a web services environment. These include AES cipher suites with 128- and 256-bit encryption; certificate revocation lists; closer control of session IDs across a mixed sysplex; and support for mixed-case passwords. There is a strong sense here that IBM is trying to address many of the security management issues that are likely to cause an obstacle when building bridges between CICS and distributed web services.

Application transformation

The other key set of announcements for CICS TS V3.1 focuses on development tools and seeks to remove some of the programming limitations that hamper the integration of CICS into the web services environment. For example, the enhancements to C/C++ support bring their performance up to the same level as applications written in COBOL, PL/I, and Assembler. WebSphere Enterprise Studio Developer has gained higher prominence as a universal development solution, while the Open Transaction Environment has been extended with support for COBOL, PL/I, Assembler and non-XPLink C/C++ OPENAPI application programs.

But the main development thrust in V3.1 is tackling the verboseness of XML, which is becoming a real problem for CICS applications attempting to process parameter data in XML and other memory-hungry formats. While the optimized data formats traditionally handled by CICS could easily cope with the 32k COMMAREA limit, XML carries far more baggage. IBM has consequently introduced the concept of containers and channels, which can handle inter-program data transfer much more flexibly than through a COMMAREA. CICS now provides EXEC API verbs to create and access these new transfer mechanisms.

All singing, all dancing?

All this boils down to a major shake-up of CICS' peripheral functionality, and more choices as to the way that host/web solutions are implemented.

When it comes to opening up CICS applications to make their processes and data available to other platforms, there are already many options available to users. IBM often leaves gaps in its function-set which are rapidly filled by

independent software vendors – and CICS is no exception. For some years companies such as Seagull, Attachmate, NetManage, NEON Systems, and Clientsoft (the latter two having just merged) have offered services and products to customers attempting to integrate CICS applications with web services. For some of these, V3.1 may be viewed as a significant threat. However, as web integration analyst Anura Guruge points out in his recent paper on the IT InDepth.com website, there is still a question of granularity here, and the all-singing all-dancing solution proposed by IBM might not be ideal in all cases:

> *"Until now, when it has come to CICS and Web services much of the interest has been about representing specific transactions, or even sub-transactions, as separate Web services. This is what Web-to-host host integration has been all about and from what I can see will continue to be all about. V3.1 does not, in my opinion, reduce the need for traditional host integration solutions such as NetManage's OnWeb 7.1 or even IBM's HATS v5 (though HATS still tends to lean more towards host publishing). Conventional host integration which relies on specific transaction capture, using the input/output fields displayed by a CICS application, provides the granularity to isolate and pick specific transactions from within a CICS application and then convert that to a Web service."*

In other words, V3.1 offers a whole range of new possibilities for customers, and for the first time (within IBM's own product portfolio) really allows a CICS application to be represented as a web service. However, in many cases, a more granular, transaction-focused approach to integration may be required, and here there are a number of third-party solutions that can be employed.

Moreover, IBM itself points out that customers must make important choices between traditional host CICS connectivity and the kind of distributed web functionality supported by V3.1. For developers in established MVS (z/OS) shops, a direct connection to CICS still offers superior quality of service and software maturity, as well as fewer APIs and system changes. The loosely-connected web route into CICS is a better option for those looking to stay as open as possible to emerging web standards, and who envisage frequent re-use of logic.

CICS as a peer player

What we see, then, is IBM laying the groundwork for CICS to act as a peer player in web-oriented applications. With CICS allowed to function as server or requester, and full support provided for integration and security services within the web environment, users will be able to take a considerably more flexible approach in re-purposing and extending CICS functions.

As IBM has said, the choice of web-oriented or direct CICS connectivity depends on a number of factors, and these factors may change as:

1. Web standards mature and mainframe support for them improves or
2. The availability, performance and cost characteristics and requirements for a given process or application change over time.

The choice of a centralized host solution for new business logic may become more or less attractive depending on changes within the organization – mergers and acquisitions, changes in policy, or cost and technical constraints. Certainly it is in IBM's interests to make sure that CICS (as well as other key mainframe sub-systems) remain as open and receptive as possible to new development, but this will be achieved not by placing the mainframe back in the center of the IT architecture but by making it as flexible and interoperative as other web service peers.

Of course, as mentioned earlier, there are a significant number of legacy mainframe users around, whose main priority is to migrate specific applications to different platforms with as little negative impact on the business as possible. They may face numerous problems in maintaining inter-program connectivity during the migration process, particularly if the move is incremental and modules that make calls on one another need to keep track of location changes. There are various issues in this scenario, including changes in data format, backup and recovery, and security. Service-oriented architectures offer a number of ways to support incremental change, by providing a flexible mechanism for inter-program communication. But for this to work effectively within a large enterprise, we need to be thinking about a much more sophisticated solution than those offered by vanilla WSDL implementations.

Strategic mainframe users want the ability to use CICS logic more flexibly within the corporate network, and that's what IBM is beginning to provide. Of course, this is only the first step, and the earlier quote from Charles Schwab indicates that it is expecting more in future releases. In particular, IBM is likely to build on the WSDL support provided so far, by adding support for more protocols as well as focusing on switching and routing and location transparency.

Opening up mainframe software

The signposts are now in place for making CICS a peer player in web services, although it's a fast-moving field and there will certainly be more announcements to come. But wherever users sit on the spectrum between strategic and legacy mainframe deployment, there is no doubt that recent CICS developments will remove many of the existing limitations to web service implementation. IBM, of course, hopes that they will also generate new mainframe business among customers that are attracted by the declining cost of mainframe technology and the potential scalability and availability characteristics.

The CICS announcements are part of a much broader strategy to open up the mainframe software product set for widespread deployment in web services. Version 8 of DB2, which was delivered in 2004, brings a whole range of new integration and portability features to IBM's enterprise DBMS. Furthermore, its new XML Publishing function allows applications to generate XML data from relational data while maintaining high performance, a major step towards storing searchable XML documents within the DBMS itself. This gives DB2 real competitive strength within the web services environment.

Arcati Research Note

Even IMS, which performs a critical role within the data center despite its maturity and low public profile, has received a number of significant enhancements to prepare it for participation in open web-oriented applications. For example, customers can transform existing IMS transactions into web services by using WebShere Studio tools, then deploy these services to WebSphere Application Server.

These major changes to the zSeries software portfolio allow users to preserve long-term investments while reacting more quickly and flexibly to changing business demands. One thing is for sure. The chasm that once separated the MVS architecture from that of other platforms is closing fast.

The author, Mark Lillycrop, is Chief Analyst of Arcati Research Ltd and an Associate of Valley View Ventures, Inc. He can be contacted at mark@arcati.com.