

The SOA Magazine

Feature Article



Service-Oriented and Systems of Record: The Northwest Customer Support Example

by Larry Guger

Published: September 22, 2008 (SOA Magazine Issue XXII: September 2008)

[Download this article as a PDF document.](#)

Abstract: Service-orientation makes information systems more flexible as well as more agile. Information technology systems can be integrated better and code that was written once for one system can be leveraged by IT departments for other systems. These are the key goals of using a service -oriented approach to systems design.

This paper is in part a retrospective and in part a direction recommendation of, and for, the service-orientation approaches taken for enterprise systems in general. Lessons learned and new approaches will be discussed as well as the specific needs that arise when dealing with systems of record and data submitted to those systems. Examples using customer data will be used throughout.

Introduction: Northwest Customer Support, an Enterprise Application Facing Service-Oriented

Northwest Customer Support (NCS) is fictional application that forms the basis of the scenarios explored in this article. Northwest Transport, a fictional company that is the transportation division of North American Services, owns a fleet of trucks to transport goods and is heavily reliant on NCS.

NCS was developed and designed in a service-oriented manner to enable flexibility and reuse. At the highest architectural level this was achieved by creating a service tier that was independent of the consumer tiers. Services were developed to enable interaction with the backend system in a stateless and scalable manner. This design permits any consumer application to utilize these services. In the eventuality that service consumers increase in either the number of users or the number of consuming applications, the service tier can be scaled out to support higher demand while maintaining service levels.

This approach works very well given two scenarios; reading data from the system and submitting data to the system. Data that is stored within NCS can be read by any consumer with the correct credentials and utilized in any manner that is required. Likewise, any consumer with the correct credentials can submit data to NCS. The NCS services take full responsibility for validating the data submitted. This is a requirement in a service-oriented design if there is a desire or expectation to expose the services to any consumer that may come along. The service must not assume any data quality other than the fact that the data has not be verified or validated in any form.

This approach met the requirements for the first release of NCS, but the scenario in which it does not necessarily work well is flowing data to other systems as it is entered or updated in NCS. As a standalone application the initial release of NCS had no integration requirements built into it. NCS was also designed as the system of record for all of the data that it stored. "System of record" is defined here as a system that is the primary owner of data. For example, NCS owns all data related to requests to transport goods on Northwest Transport's fleet of trucks. The assumption is that no other system requires this information beyond a cursory or "read only" manner. However, the assumption that NCS was the sole system of record for customers was incorrect and the service enablement approach that was used did not support integration with other systems of record of customer data.

The Customer Creation Process

By analyzing the business process for introducing a new customer into Northwest Transport it becomes obvious that NCS was not necessarily the system of record for customers. As part of introducing a new customer into Northwest Transport the user that enters the new customer into the application must also submit the customer data to another system, Central Identity Control (CIC), to get a Customer Identification Number (CI#) for that customer.

This CI# is used as the unique identifier for the customer in other systems within the North American Services enterprise application ecosystem, in particular the central accounting system. As the accounting system is a single enterprise application within North American Services it could be considered the system of record rather than NCS; however it uses customer data in a different context - that of payment processing.

Analysis of the business process of creating a new customer gets closer to identifying the true systems of record for data, of which there may be more than one. This should reinforce the importance of business process analysis when developing and designing service-oriented systems and the need to have these services be business-driven and not technology-driven. This also identifies the importance of identifying the context to a system of record and recognizing that there may be more than one.

Increasing Integration

As development of the NCS suite proceeds, continued analysis of business processes further identifies the interactions with other external systems. These interactions are becoming both clearer as well as more abundant.

As the teams that are developing NCS work with more business departments within the enterprise, further uses of common data are identified. To stay with the customer example it is discovered that the Credit and Risk department is storing customer information in spreadsheets as well as a credit scoring application. This is in addition to a legacy invoice application, NCS and any systems that reside in the enterprise data center, two of which are currently known: CIC and the master accounting system.

The legacy invoicing application is being replaced, however in its place is a third party application that will continue to contain customer data. This means that new customer data entered into NCS will need to flow to these other systems as customer data is updated. Automated integration between each of these systems is the ultimate IT goal. The challenge is the difference between the business processes within Northern Transport and the business processes at the corporate level, as well as the systems that support them.

NCS was developed to enable the business process required to create a new customer and then support contracts with those customers. To date, the customer information is entered and validated within NCS with no concern for corporate accounting systems. The user is then required to submit the same information for North American Services corporate to assign a CI#. This number is then passed through a couple of hands via email until it is fed into the invoicing system as well as being entered into the corporate accounting system in some manner to support data integration between the shipping division and Northwest Transport corporate systems.

To fully automate the process of entering a new customer once and have a new CI# automatically assigned would require several system upgrades that may or may not happen before the new invoicing system is delivered into production.

We are faced with a situation in which there is one system of record for customers that do business with NCS, a separate system of record for customers that submit payments to North American Services (central accounting) and a third system of record for tracking and assigning CI#s. This means that we have at least three systems of record for the same customer data. This situation is not unique for large enterprises; what's important is recognizing that systems are to be developed to support both the current manual process as well as future automation.

Flowing Data to Match Business Processes

Whether the flow of data is manual or automated, there exists the need to enter customer data first in NCS to support business needs. For example a recently added customer may have had an agreement defined that requires both new customer data as well as new contract data be entered into NCS at the same time so as to initiate a shipment of goods

(which is the business of Northwest Transport). Moving new customer data entry to a central process in head office may not support the timeliness requirements of the business process at the shipping division even though it would make more sense from a data flow (CI# assignment) perspective.

To provide timely business process we need to support the entry of customer data in NCS first. Until the CI# assignment process becomes service enabled there will continue to be the need to re-enter customer data into the CIC system in addition to the NCS system. There is also the need to flow the customer data from NCS to the new third party application to support invoicing functionality as well as to the enterprise accounting application.

To support the flow of customer data from NCS to the invoicing application, or any other system that may need the data now or in the future, a publish/subscribe system is recommended. This same approach will be used wherever data needs to be created in any application as the system of record and then propagated to other systems of interest. (In this example customer and contract data would both have this need.)

After the data is written to the NCS database it can be submitted to a service bus implemented using technologies such as BizTalk Server that can be used to feed that data to other systems of interest. The data must first be written to the NCS database to ensure that it is captured in the system of record as well as to get the primary key for new records before publishing it to other systems.

To support this in the most flexible manner, a factory pattern should be used to load the publisher code. Within the NCS application a call to load the publisher code is made followed by a call to publish the corresponding data. The publisher code will be responsible for submitting the data to the appropriate messaging system. This approach enables changing the publishing code and respective process as well as changing consumers of the data without the need to alter any of the NCS code. New consumers of the data can be added by creating a new subscription in the service bus and having it forward the data to the new consumer's endpoint.

Conclusion

What this example helps demonstrate is that systems of record, of which there may be many for a single data item depending on context, provide an effective means of letting interested parties know of new entries or changes to existing records. By using the service bus publish mechanism and a dynamic publishing module within the application, the publishing mechanism can be decoupled from the system implementation. When applying the publish-and-subscribe model, subscribers can be added and removed at any time without impacting existing systems integration.

By moving to this model we can build systems that more easily support service-orientation at the enterprise level, while also easing future integration and automation and maintaining the integrity of systems of record.

