OASIS WSIA Technical Committee

Requirements Document
Use Case Report: (Re)Publishing Components

Version <1.1>

Revision History

Date	Version	Description	Author
05/Mar/2002	1.0	Republished Producers	Dan Gisolfi, Graeme Riddell, Alan Kropp, Eilon Reshef, Gil Tayar, Rex Brooks, Ravi Konuru, Keven Brinkley, Aditi Karandikar, Monica Martin, Rich Thompson, Charlie Wiecha
14/Mar/2002	1.1	Add basic, alternate flows	Alan Kropp

Requirements Document	Version: <1.1>
Use Case Report: <Use-Case Name>	Date: 3/15/2002
WSIA Republished Use Case.DOC	

Table of Contents

Use Case Report: (Re)Publishing Components

1. Definition of the Republishing use case

Definition: Use cases in which the Consumer, having added value to producer WSIA services through the application of coordination, orchestration, etc., now wishes to re-publish the component as a new first line component, for other consumers downstream to integrate.

1.1 Brief Description

2. Actors

There are two actors in this use case:

-   Producer: one or more WSIA web services

-   Republishing Consumer (or Republisher): a Consumer which instantiates and controls interaction with the Producers on behalf of downstream Consumers (who are themselves acting on behalf of their End Users), playing the role of intermediary to route consumer requests to producers, and producer responses back to the consumer.

3.   Flow of Events

3.1   Basic Flow

[Traveler's checks] The consumer actor in this scenario (EuroVacation) decides to package the online travelers check ordering service as part of a general purpose vacation planning tool, for licensing to its affiliate travel partners (downstream consumers) around the world. Branding for the online ordering tool is to be passed through to the integrating consumers, but EuroVacation enforces no additional branding or L&F restrictions as part of the license.

[Mortgage center] A mortgage broker has created a one-stop shopping site for real estate financing, from initial offer to escrow. The mortgage calculator and related services feature prominently, but there is no branding requirement from the producer of these services. The broker maintains tight branding and L&F control over this component, hoping to increase awareness of its premium services by offering it at no charge to real estate listing services and agencies.

- This case is a "wrapper" for all of the other use cases that precede it, in that any service, however composed via customization, orchestration, etc., is a candidate for republishing. This case could also be considered, at its most generic, the general case describing what the Provider (including Consumers acting in the Republisher role) does to publish any service.

- Republisher creates a description of allowed context information that each producer supports. This may start out as a superset of the each producer's (if there are more than one) context information, or the republisher may choose to refine it by adding additional fields, or removing (establishing default values) existing ones.

- Republisher creates the Adaptation Description Specification for the new service. It is in this specification that the republisher establishes branding behavior, as well as look-and-feel restrictions to what the consumer is permitted to change.

- The Adaptation Description Specification and context information together comprise the "Contract" for this published service. A Consumer obtains the contract and uses it to integrate the service into an application. When an End User uses this application, the Consumer makes requests on behalf of the End User of, in the basic flow, the Republisher.

- The Republisher requests output from each Producer service, passing it the appropriate context information from the consumer request. Producers respond with their output, along with any context information changes that may occur as a side effect of the assignments made by the Republisher.

- The Republisher passes on the producer response(s) to the consumer. As the intermediary in this flow, the Republisher does not assemble the page, although it may perform normalization or unmarshalling of the response data for the consumer.

- The use case continues with the standard Consumer-Producer interaction, except that the Republisher stands in for the Producer. The Consumer assembles the page resulting from output from the Republisher, along with any additional page content originating within the Consumer itself. Embedded URLs and form actions are rewritten by the Consumer to allow for redirecting End-User actions to the appropriate Producer.

- The consumer may create an integrated form from multiple Producers. In this case, the adaptation information should have sufficient information to remove redundant submit buttons either at the producer or the consumer. Further, an interaction such as submit must be parsed at the consumer to send specific components to the correct Producer.

- End-User interactions with the Consumer's page are directed back to the Consumer. At the Consumer, rewritten URLs are decoded to delegate the End-User action back to the appropriate Producer component. The Consumer invokes the Producer component passing any arguments provided by the page's action invocation as opaque data to the Producer.

- The Republisher, on returning from the action invocation, must pass updated context information that it has received from the Producer(s) back to the Consumer, indicating the effect of the user action on the published context maintained by the Producer.

-

3.2 Alternative Flows

3.2.1 Republisher in non-Itermediary Role

The Republisher does not interpose in Consumer-Producer interactions. The Consumer integrates the republished service(s), and interacts directly with the original Producer(s), as in the other Consumer-Producer use case categories. The Republisher has no other role but to create and maintain the Contract for the service, and make it available to interested parties (Consumers).

4. Diagrams

4.1 Relationship between Producers and Consumers in the Orchestrated Producers Use Case

4.2 < First special requirement >

5. PreConditions

[A precondition (of a use case) is a textual description of any constraints or dependencies that must be satisfied prior to entry of the use case.]

5.1 < Precondition One >

6. PostConditions

[A postcondition (of a use case) is a textual description of any constraints or dependencies that must be

Requirements Document	Version: <1.1>
Use Case Report: <Use-Case Name>	Date: 3/15/2002
WSIA Republished Use Case.DOC	

satisfied after termination of the use case.]

6.1   < Postcondition One >