
OASIS WSIA Technical Committee

**Requirements Document
Use Case Report: Customized Producer**

Version <1.0>

Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

Revision History

Date	Version	Description	Author
26/Feb/2002	1.0	Customized Producer/Consumer without coordination or flow control	Charlie Wiecha

Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

Table of Contents

1. Customized Producer	Error! Bookmark not defined.
1.1 Brief Description	1
2. Flow of Events	1
2.1 Basic Flow	1
2.2 Alternative Flows	2
2.2.1 Static definition of Producer context by the Consumer	Error! Bookmark not defined.
2.2.2 Dynamic definition of Producer context by the Consumer	Error! Bookmark not defined.
3. Diagrams	4
3.1 WSIA Activity Flow for Customized Use Case	8
4. PreConditions	8
4.1 < Precondition One >	8
5. PostConditions	8
5.1 < Postcondition One >	8

Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

Use Case Report: Customized Producer

1. Customized Producer

1.1 Brief Description

The Customized Producer use case captures the scenarios in which the Consumer is able to initialize Producer components with application specific context, under programmatic control, and retrieve application specific results from the Producer components upon completion of some unit of their work.

The Customized use case extends the Aggregated use case by providing for application specific context, and by allowing for that context to be both passed into the component by the Consumer, and returned from the Producer to the Consumer at defined points of the Producer's execution.

The Customized use case is less ambitious than the Integrated use case in that the interface between Consumer and Producer in this use case still does not capture the formal input and output data values passed between them, such as at initialization and termination times. In the Integrated case, these data are specified between Producer and Consumer so that the Consumer is free to call relevant parts of the Producer on behalf of the End-User without their involvement.

Note that in this use case, the Consumer does not republish itself as a web service -- this step is left to the Republished use case. Thus the consumer executes as a platform-specific container such as a portal server or a conventional servlet.

2. Actors

There are three actors in this use case:

- Producer: one or more WSIA web services
- Consumer: a platform-specific container which instantiates and controls interaction with the Producers on behalf of End-Users
- End-User: a person who interacts directly with the output of the Consumer

3. Flow of Events

Two flows are considered in this use case:

- Static definition of a Producer's context by means of a Consumer-provided resource definition, and
- Dynamic definition of a Producer's context, computed by the Consumer using relevant data about the End-User's profile, interaction state, or other information.

3.1 Basic Flow

- The End-User enters a URL pointing to the Consumer into a browser
- The Consumer is instantiated by an application server, which creates a session on it for interaction with the End-User

Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

- The Consumer instantiates WSIA proxies for each of the Producer services it wishes to include in the page to be returned to the End-User, and creates a session on each to represent interaction with the End-User
- The Consumer obtains the definition of the allowed context information supported by each Producer. This information may be obtained statically from a directory service such as UDDI for the given Producer or dynamically by querying the Producer service itself.
- The Consumer requests output from each Producer service, passing it the appropriate context information. Producers respond with their output, along with any context information changes that may occur as a side effect of the assignments made by the Consumer.
- The Consumer may adapt each Producer's returned page using the Producer-supplied Adaptation Description Specification to further customize the output by embedding additional output, removing optional output, or replacing content such as optional images.
- The Consumer assembles the page resulting from output from each Producer, 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 providers . In this case, the adaptation information should have sufficient information 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 the right parts to the right provider.
- 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 Producer, on returning from the action invocation, may pass updated context information back to the Consumer indicating the effect of the user action on the published context maintained by the Producer. In the case where the Producer has completed its interaction with the End-User, this result context is to be interpreted as the output of the Producer and will be used by the Consumer in continuing its dialog with the End-User.

3.2 Alternative Flows

3.2.1 Static definition of Producer context by the Consumer

- In this alternative flow, the Consumer proceeds as above through instantiation of the Producer and creation of a session for interaction with the individual End-User.

Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

- The context information to be passed to the Producer is determined statically in this flow by the Consumer through use of an appropriate Producer-specific persistent store.

3.2.1.1 < An alternative sub-flow >

- In this alternative sub-flow, the static context information determined above by the Consumer is overridden by End-User specific preferences, also obtained statically from a persistent End-User profile store. It is assumed that End-User preferences have precedence over those set on an application-specific basis by the Consumer for all users.

3.2.2 *Dynamic definition of Producer context by the Consumer*

- In this alternative flow, the Consumer derives Producer-specific context information dynamically using available information on the End-User's preferences, on the current or historical interaction state, and any other information available to the Consumer.

3.2.3 *Producer applied adaptation and URL rewriting*

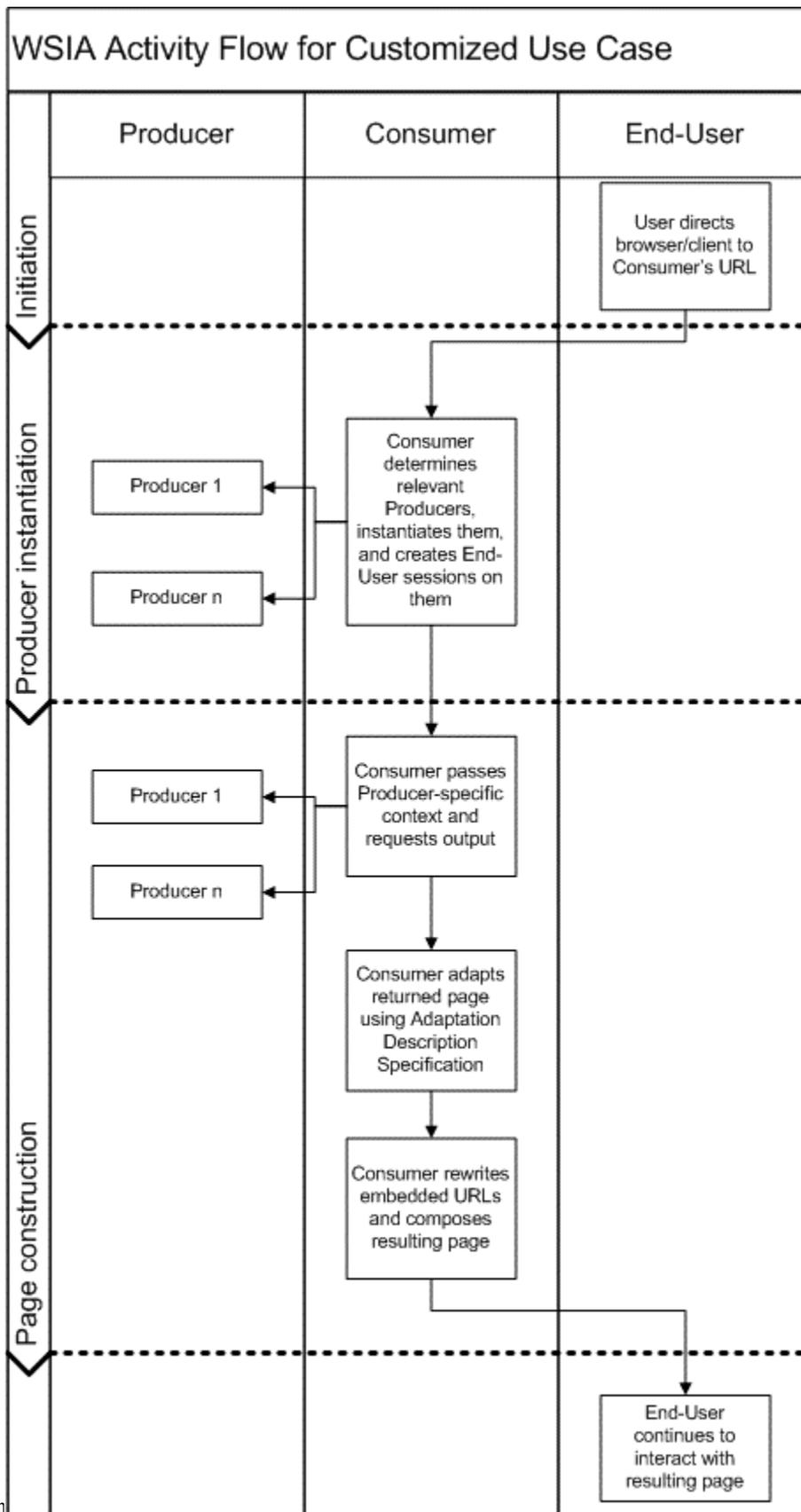
- To further customize the output, Producers adapts output based on passed context information. The adaptation can embed additional output, remove optional output, or replace content such as optional images.
- To enable redirection of all End-User actions back to Consumer's page, Producer rewrites URLs in the page to link back to Consumer URL identified in the passed context information.

Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

4. Diagrams

4.1 Main flow for Customized use case

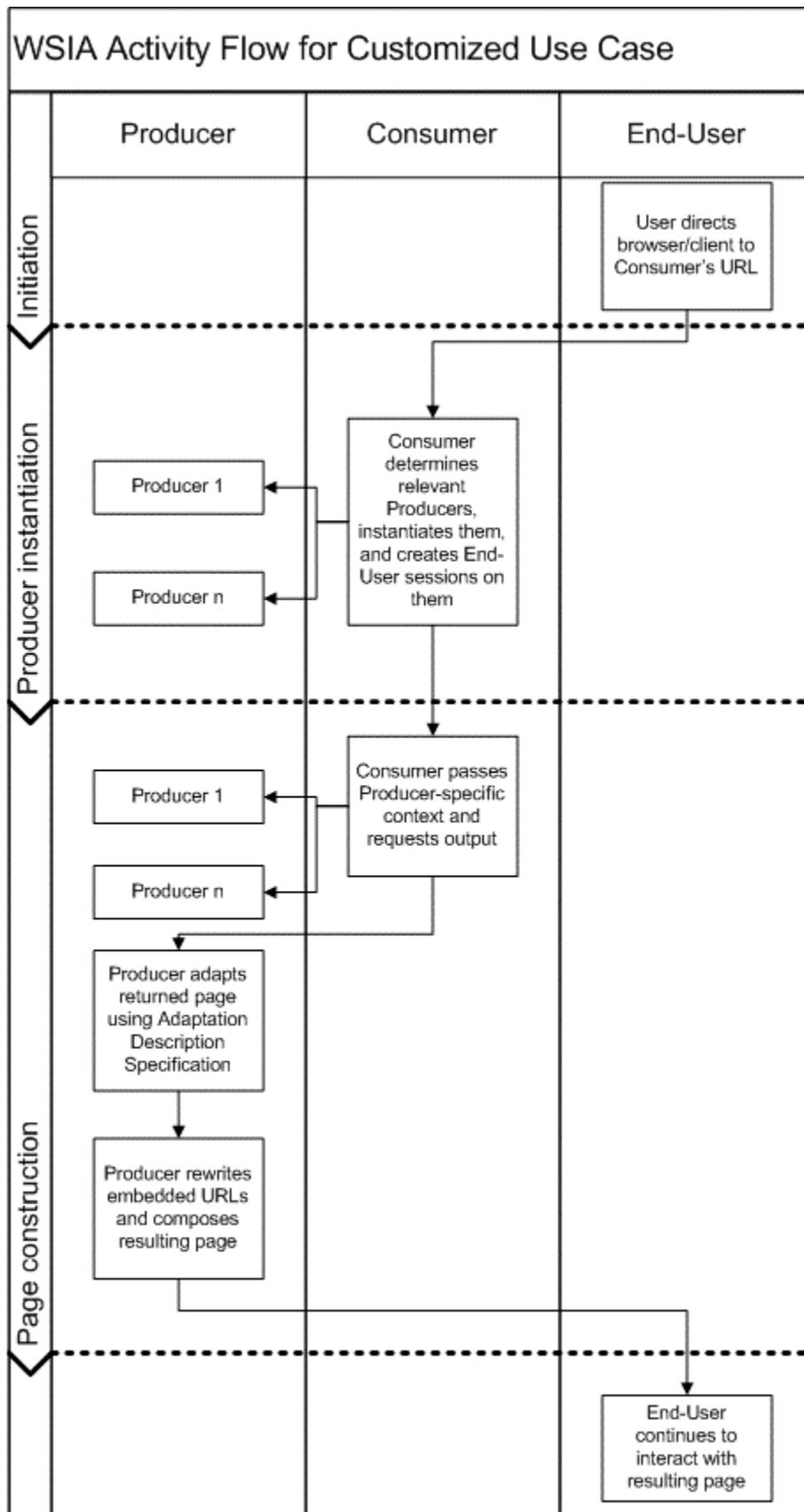
Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	



Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

4.2 Alternative flow diagram for Producer applied adaptation and URL rewriting

Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	



Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

4.3 < 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 satisfied after termination of the use case.]

6.1 < Postcondition One >