

---

**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	Dan Gisolfi, Graeme Riddell, Alan Kropp, Eilon Reshef, Gil Tayar, Rex Bourne, Ravi Konuru, Keven Brinkley, Aditi Karandikar, Monica Martin, Rich Thompson, Charlie Wiecha
05/Mar/2002	1.1	Updated with alternative flows	Charlie Wiecha

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

## Table of Contents

1.	Definition of the Customized Producer use case	1
1.1	Brief Description	1
2.	Actors	2
3.	Flow of Events	2
3.1	Basic Flow	2
3.2	Alternative Flows	3
3.2.1	Static definition of Producer context by the Consumer	3
3.2.2	Dynamic definition of Producer context by the Consumer	3
3.2.3	Producer applied adaptation and URL rewriting	4
3.2.4	Look and feel adaptation permitted at the Consumer	4
3.2.5	Look and feel adaptation prohibited at the Consumer	4
3.2.6	Excluding elements by configuring the Producer	4
3.2.7	Excluding elements by editing output at the Consumer	4
3.2.8	Adding elements by configuring the Producer	4
3.2.9	Adding elements by editing output at the Consumer	4
3.2.10	Replacing element values by configuring the Producer	4
3.2.11	Replacing element values by editing output at the Consumer	4
3.2.12	Editing element values by configuring the Producer	5
3.2.13	Editing element values at the Consumer	5
3.2.14	Adding elements conditionally	5
3.2.15	Adding new options to existing questions	5
3.2.16	Setting preferences for interaction technique selection and formatting at the Producer	5
3.2.17	Removing the need for user interaction by Consumer-computed values	5
3.2.18	Redefining user actions by configuring the Producer	5
3.2.19	Redefining user actions by interception at the Consumer	5
4.	Diagrams	5
4.1	Relationship between Producers and Consumers in the Customized Use Case	5
4.2	Main flow for Customized use case	7
4.3	Alternative flow diagram for Producer applied adaptation and URL rewriting	9
4.4	< First special requirement >	11
5.	PreConditions	11
5.1	< Precondition One >	11
6.	PostConditions	11
6.1	< Postcondition One >	11

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

## Use Case Report: Customized Producer

### 1. Definition of the Customized Producer use case

In this case case, the Producer's interface includes specific information (properties, operations) for that Producer enabling programmatic access by the Consumer to the Producer for customization, rather than end-user or admin customization through the visual service itself (e.g. edit page in portlet).

Depending on the capabilities of the Producer, its interface may describe one or more of the structure of (1) its input/output data, (2) properties of its output (preferences for interaction techniques or formatting (e.g. calendar mode), and (3) the operations it needs to call to perform various actions such as filtering queries and validating data.

Unlike the Integrated use case, the Producer may be Customized only at the beginning of its execution. It does not maintain updated values for the three categories of values listed above. [CFW: still want to separate Customized and Integrated?]

The Consumer may wish to adapt, or customize, the settings of any of these aspects of the Producer's interface and/or of the output it returns.

#### 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 during the Producer's execution.

[CFW: Do we still want to separate the Customized and Integrated cases? Once we can define Producer-specific properties and operations, it seems arbitrary to restrict their use to certain points in the lifecycle. Original text follows...]

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.

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

## 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 in Figure 4.1 is instantiated by an application server, which creates a session on it for interaction with the End-User
- The Consumer instantiates WSIA proxies as shown in Figure 4.1 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. Note that literal instantiation of a proxy may not be required if the Consumer chooses to interact with Producers via a framework such as WSIF -- the Web Services Invocation Framework -- which allows direct access to services without use of proxies.
- 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.
  - Separate cases for various forms of customizing data, presentation preferences, and generated output are given below in the subflows sections.
- 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.

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

- 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.

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.

[Multimedia sports portal] Maintain a user profile that tracks user interest and provides relevant information based on user profile.

[Traveler's checks] Customization can either be static (offline configuration, a typical portal scenario) or dynamic (a per-user or per-session configuration, a.k.a., personalization).

#### 3.2.1.1 End-user preferences override static Consumer definitions

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.

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

[Financial charting] FAME Consumer application determines that stock plot is required for a given end-user, selects it from the UDDI directory, and sets iChart-specific preferences on it given parameters from the end-user's profile. No admin or end-user edit mode screens are required as these configuration parameters are exposed in the iChart service description directly. End-user interactions may override initial choices picked by the Consumer. Example Consumer specified preferences may include stock symbol, start/end dates, averaging options, etc. For calendar displays, Consumer may select options such as day/week/month view, US vs. European format (Sunday vs. Monday starts a week).

### 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.

### 3.2.4 *Look and feel adaptation permitted at the Consumer*

[Universal bank, Mortgage center, Product configurator] Consumer is able to make modifications to look and feel (skin) of the Producer to achieve its own branding goals such as a unified look and feel across Producers.

### 3.2.5 *Look and feel adaptation prohibited at the Consumer*

[Beauty boutique, Insurance Enrollment, SmartBuyer, Multimedia sports portal] Consumer is precluded from making modifications to look and feel (skin) of the Producer to preserve branding requirements of the Producer.

### 3.2.6 *Excluding elements by configuring the Producer*

### 3.2.7 *Excluding elements by editing output at the Consumer*

### 3.2.8 *Adding elements by configuring the Producer*

[Mortgage center, Product configurator, Traveler's checks] Consumer requests elements to be added by the Producer either before generating output

### 3.2.9 *Adding elements by editing output at the Consumer*

[Mortgage center, Product configurator, Traveler's checks] Consumer inserts elements into the output generated by the Producer as it passes through the Consumer.

### 3.2.10 *Replacing element values by configuring the Producer*

### 3.2.11 *Replacing element values by editing output at the Consumer*

[Product configurator] Consumer replaces logos or other branding elements in the Producer's display either before generating output (subflow 1) or as the output passes through the Consumer (subflow 2).

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

### 3.2.12 *Editing element values by configuring the Producer*

[Product configurator] Consumer adds an additional column to a table generated by the Producer by inspecting the values of a key field, such as part number, and adding a column for corresponding values for a new field, such as stock status. Consumer passes stock status back to the Producer and requests the output for the customized table.

### 3.2.13 *Editing element values at the Consumer*

Consumer is given information by the Producer about the table structure to enable it to make the modifications locally.

### 3.2.14 *Adding elements conditionally*

[Mortgage center] Add new questions asked, depending on answers given to the original (Mortgage) questions -- in this use case on the same page.

### 3.2.15 *Adding new options to existing questions*

[Mortgage center] calculation for US locations, but Canadian lending institution. The business calculation has to be adapted too. Tweak how the calculation of the home price affects the final number (in scope?)

### 3.2.16 *Setting preferences for interaction technique selection and formatting at the Producer*

Configure a calendar by selecting the preferred type and controlling display properties on the interaction technique such as day/week/month views or starting day of the week.

### 3.2.17 *Removing the need for user interaction by Consumer-computed values*

[Mortgage center] Remove a question about property tax value and provide a Consumer-specified value. Tweak the generic property tax value to be the exact value for the given city.

### 3.2.18 *Redefining user actions by configuring the Producer*

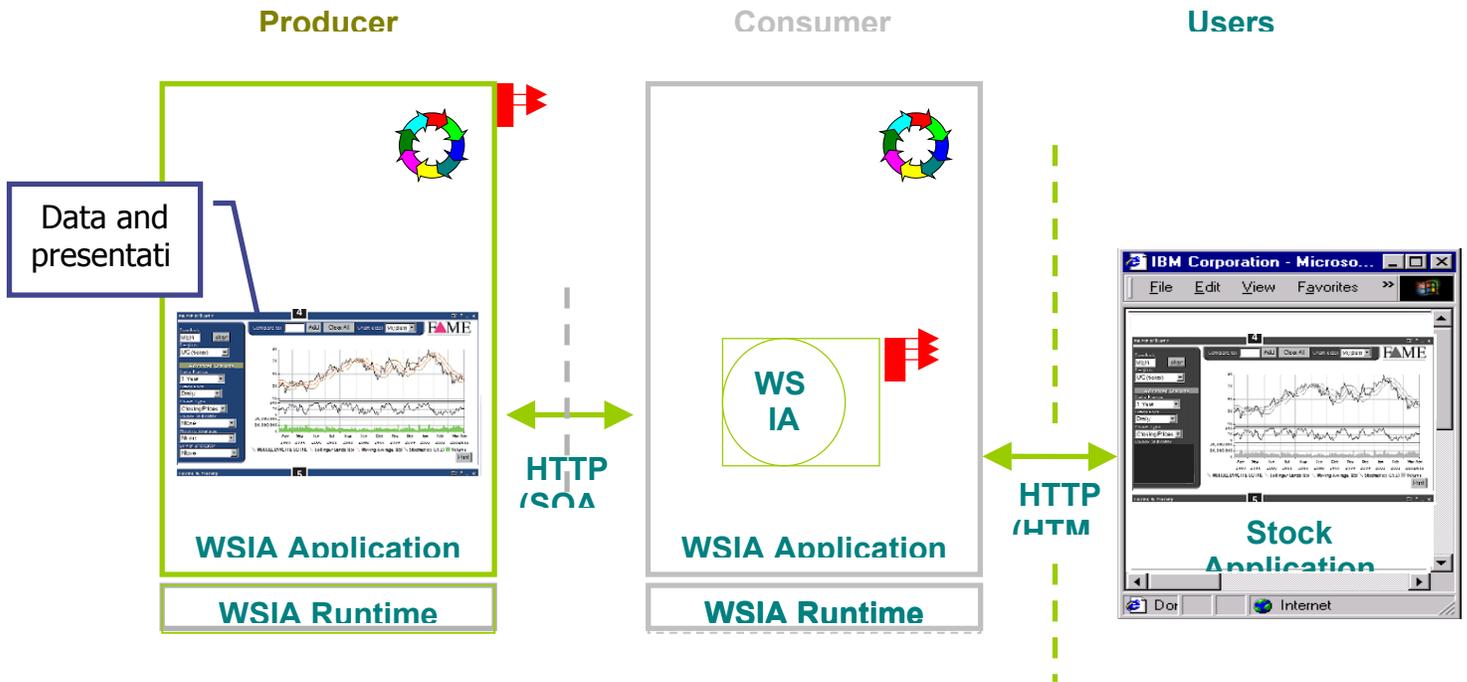
[Product configurator, Buy.com] Altering the function of the Add to Shopping Cart button, subflow 1: intercepted by the Consumer, subflow 2: caught by the Producer by action redirected back to the Consumer.

### 3.2.19 *Redefining user actions by interception at the Consumer*

## 4. **Diagrams**

### 4.1 **Relationship between Producers and Consumers in the Customized Use Case**

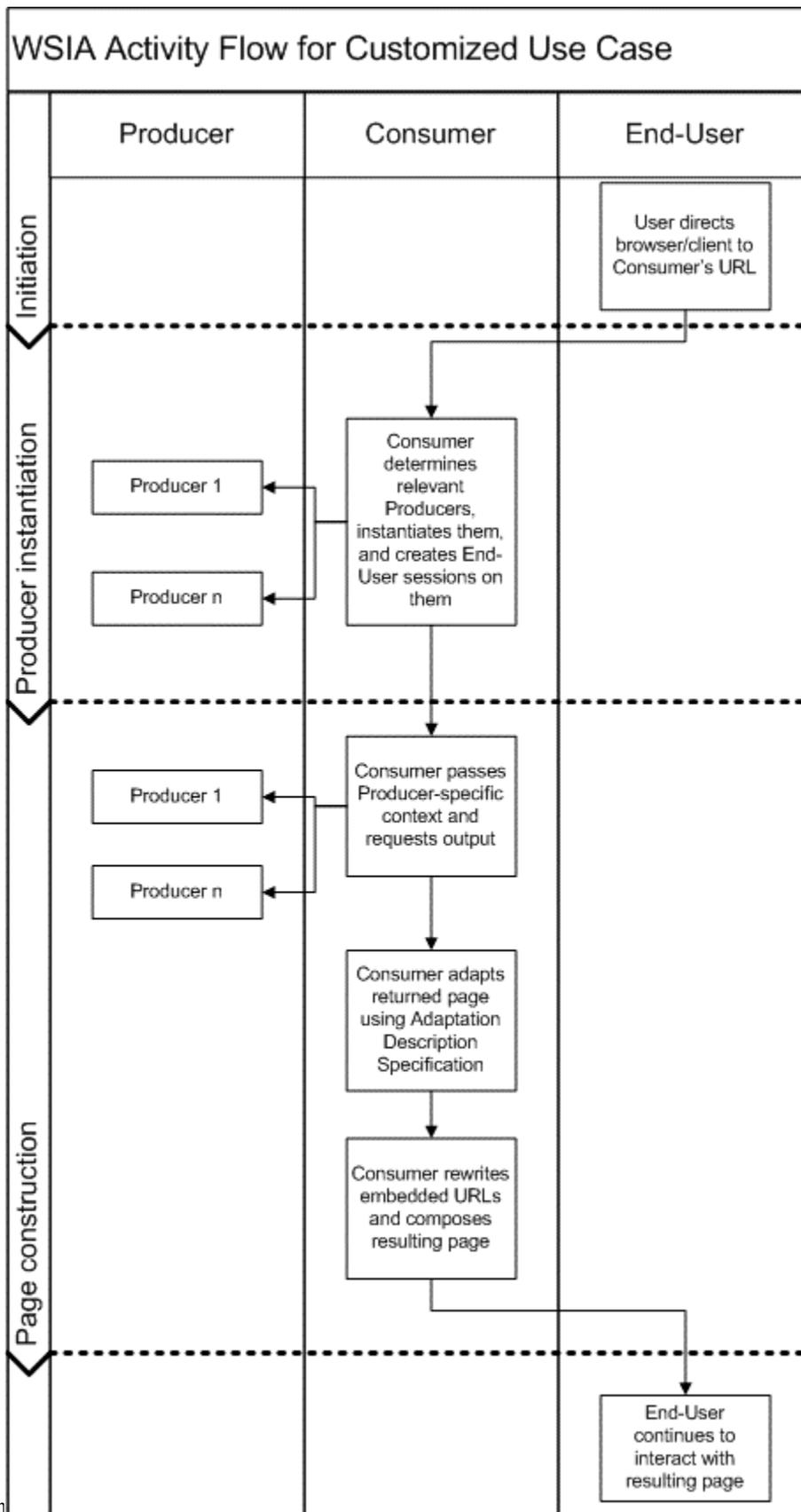
Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	



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

#### **4.2 Main flow for Customized use case**

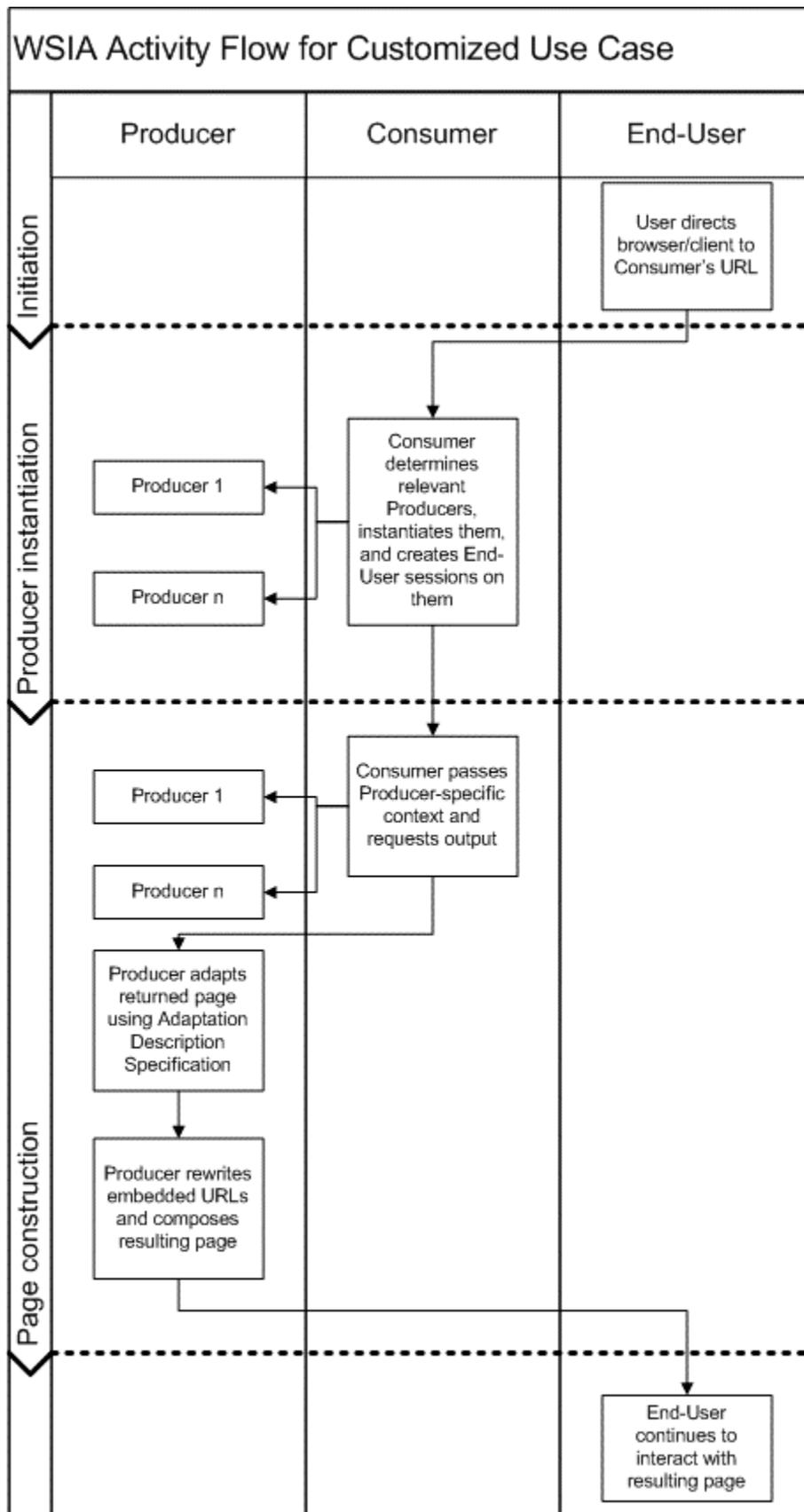
Requirements Document	Version: <1.0>
Use Case Report: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	



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

#### **4.3 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>	

**4.4 < 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 >**