ÍNDICE

1. Evidencia

1.1 Evidencia de la investigación

1.1.1 Tema de Investigación

WSRP (Web Services for Remote Portlets).

1.1.2 Descripción del tema

Estándar de OASIS acerca de la configuración y manejo de Web Services para la comunicación con Portlets remotos. Actualmente, este estándar se encuentra en la versión aprobada 2.0.

1.1.3 Resultado de la Investigación

1. Se debe tener en claro que los actores que interactúan con la aplicación del estándar son: End-User, Producer, Consumer y Portlet.
2. El flujo típico del proceso para la comunicación de Portlets es el siguiente:
   a. El Consumer "descubre" al Producer. Esto implica que el Cosumer tiene que aprender la URL del "endpoint" del Web Service del Producer y obtener los metadatos del Producer con la descripción de los requisitos del registro y, posiblemente, una indicación de que Portlets está exponiendo el Producer.
   b. Establecimiento de una relación entre el Producer y el Consumer. Esto implica el intercambio de información respecto a las capacidades, los requerimientos de seguridad u otros aspectos técnicos o de negocio de la relación.
   c. Aprendizaje del Consumer de todas las capacidades y servicios del Producer basado en la relación ya establecida.
   d. Establecimiento de una relación entre el Consumer y el End User. Esto permite que el Consumer autentique al End User y pueda permitir la personalización las páginas agregadas presentadas por el Consumer.
   e. Producción de páginas agregadas. Esto suele implicar la definición de un nivel básico de diseño de la página (a menudo con Portlets personalizados) y puede implicar una mayor personalización de las páginas por el End-User.
   f. Solicitud de una página. Esto normalmente se produce cuando el End-User dirige a un agente usuario (ej. Navegador) a la URL del Consumer, pero también ocurre indirectamente como resultado del procesamiento de una interacción con el "markup" de una página anterior.
   g. Procesamiento de interacciones. Algunas interacciones del End-User con el "markup" de una página se convertirá en una invocación al Consumer para proporcionar alguna función lógica. El Consumer será parte del proceso de

invocación para determinar el Producer/Portlet a la cual la interacción se ha dirigido y la naturaleza de la invocación solicitada al portlet.

3. La interfaz "markup" es necesaria, ya que define las operaciones para obtener el marcado de un portlet. Esta interfaz también contiene la operación de asistencia del consumidor en la pre-inicialización de las cookies de HTTP. Tras esta operación la interfaz evita los problemas asociados al movimiento de las cookies entre enlaces. Código ejemplo:
4. La interfaz de registro, es una interfaz opcional, define las operaciones para el establecimiento, actualización y destrucción de un registro. Por cada inscripción se refleja una relación particular entre un Consumer y Producer. Código ejemplo:
5. La interfaz de Gestión de Portlets, es una interfaz opcional, define las operaciones para obtener los metadatos del portlet, la clonación de Portlets para una mayor personalización e interacción con las propiedades de la interfaz
6. La interfaz de Descripción de Servicio presenta las siguientes estructuras de datos:
    a. Tipo Extension: La estructura de extensión contiene el mecanismo de extensión de carga útil para los proveedores y extensiones de aplicaciones.
    b. Tipo Handle: Los Handles son referencias privadas que se pasan entre el consumidor y el productor. Los Handles son representados como cadenas restringidas en el protocolo. Aunque es principalmente una cadena de longitud ilimitada, la longitud del Handle está restringido por las siguientes razones: Los Handles pueden ser almacenados en bases de datos y puede ser utilizado para la indexación. El Consumer podrá probablemente integrar handles en las direcciones URL del cliente. La comparación de los handles deben ser eficientes.
    c. Tipo Key: Las Keys son similares a los Handles, excepto que no se encuentran referencias privadas. Se utilizan para introducir datos. Es recomendable usar los primeros 127 caracteres del Unicode ya que el valor representado no puede tener más de 255 bytes de almacenamiento.
    d. Tipo ID: Se usan para hacer referencias, pero es improbable que se utilicen como Keys. La restricción de longitud esta impuesta a 4096 caracteres. El incumplimiento de esta regla puede causar que se pierda el valor que se desee almacenar.
    e. Tipo LocalizedString: La estructura LocalizedString describe un valor local determinado y el nombre que se puede utilizar para extraer otros valoreslocales de un ResourceList.
    f. Tipo ResourceValue: Esta estructura proporciona el valor de un recurso para una configuración local.

g. Tipo Resource: La estructura Resource lleva a los valores de un recurso en un conjunto de localidades.

h. Tipo ResourceList: Es un arreglo de la estructura Resource.

i. Tipo ItemDescription: Esta estructura se utiliza para describir los elementos personalizados que se permite al Consumer recurrir al interactuar con los portlets del Producer.

j. Tipo MarkupType: se utiliza para llevar los metadatos de portlet que es mimeType específico.

k. Tipo EventDescription: proporciona la información necesaria para describir los eventos del portlet.

l. Tipo PropertyDescription: Cada propiedad de un portlet es ilustrada con la esta estructura.

m. Tipo ModelTypes: contiene el mecanismo de carga para la declaración de los tipos de referencia por los tipos de descripción.

n. Tipo ModelDescription: El conjunto de propiedades de un portlet se describen en sus metadatos usando esta estructura.

o. Tipo ParameterDescription: Los parámetros del portlet se modelan como una matriz de cadenas, de modo que el tipo de información no necesita ser declarada, y se describen mediante esta estructura.

p. Tipo PortletDescription: Contiene un conjunto de campos que ofrecen los metadatos para describir el portlet, así como cualquier clones del portlet.

q. Tipo Property: Se utiliza para llevar escrito información entre los Consumer y los Producer.

r. Tipo ResetProperty: lleva el nombre de una propiedad para que el Consumer restablezca los valores predeterminados.

s. Tipo PropertyList: Reúne un conjunto de estructuras para la transmisión de la propiedad en conjunto entre el Consumer y el Producer.

t. Tipo CookieProtocol: Este tipo es una restricción sobre el tipo de cadena que se ve limitada a los valores "none", "peruser" o "perGroup".

u. Tipo ExtensionPart: Contiene un conjunto de campos para la descripción de una parte de una extensión del protocolo.

v. Tipo ExtensionDescription: Contiene un conjunto de campos para describir un protocolo de extensión.

w. Tipo ExportDescription: Contiene un conjunto de campos para describir la capacidad de exportación y restricciones.

x. Tipo ServiceDescription: Contiene un conjunto de campos que describen los servicios ofrecidos por el productor.

y. Tipo LifeTime: La estructura proporciona información acerca del tiempo de vida cuando un artículo en particular está programado para ser borrado.

z. Tipo RegistrationState: Contiene los campos relacionados con un registro particular de un Consumer con un Producer.

aa. Tipo RegistrationContext: Contiene los campos relacionados con un registro particular de un Consumer con un Producer. Es devuelto por la operación de registro, es un parámetro necesario en la mayoría de otras operaciones.

7. La interfaz "Markup" posee la siguiente estructura de datos:

a. Tipo SessionContext: Esta estructura contiene el ID y la información que expira, el consumidor la necesita para hacer referencia a la sesión en invocaciones posteriores.

b. Tipo SessionParams: Esta estructura contiene la información de sesión, el Consumer abastece a los portlets para la conexión a una sesión de portlets.

c. Tipo RuntimeContext: Define una colección de campos utilizados sólo en las interacciones transitorias entre el Producer y el Consumer.

d. Tipo PortletContext: Se utiliza como parámetro en muchas operaciones para suministrar la información de portlets que fue retornada al Consumer.

e. Estándar UserScopes: Esta especificación define los valores iniciales para los UserScopes. Los UserScope son un conjunto abierto de valores, en el cual, el Producer debe restringir los valores suministrados a los Consumer. Si se especifica otro valor y el Consumer no lo entiende, el Consumer debe ignorar el control de la caché y tratar el contenido como no-cacheable.

f. Tipo CacheControl: Contiene un conjunto de campos necesarios del portlet para administrar en caché fragmentos de markup. Se sugiere tener en cuenta que cualquier clave utilizada por el sistema de caching para localizar este marcado debe incluir la estructura MarkupParams.

g. Tipo Templates: Contiene un conjunto de campos que permiten a los Producer URL escribir.

h. Tipo CCPPProfileDiff: Contiene la información definida por el CC / PP norma para declarar las diferencias con los perfiles de referencia. (CC/PP = Composite Capabilities/Preference Profiles).

i. Tipo ClientData: Contiene la información del cliente suministrado a los Consumer acerca de sí misma, incluyendo la identificación de usuario-agente y capacidades.

j. Tipo NamedString: Proporciona una forma estandarizada de llevar un nombre o valor simple juntos.

k. Tipo NavigationalContext: Proporciona un medio para llevar tanto la parte pública como privada del Estado de navegación del portlet.

l. Tipo MimeRequest: Contiene información frecuentemente usada para controlar la generación de ítems con diferentes tipos de mime (Multipurpose Internet Mail Extensions).

m. Tipo MarkupParams: La definición del esquema de la estructura MarkupParams se extiende de la definición común MimeRequest sin ningún tipo de campos adicionales y proporciona los datos necesarios para que el portlet pueda generar markup que permitirá al usuario final visualizar el estado del portlet.

n. Tipo ResourceParams: La definición del esquema de la estructura ResourceParams se extiende de la definición común MimeRequest agregando campos específicos para invocar la operación GetResource.

o. Tipo MimeResponse: Contiene campos comunes relativos al retorno de un ítem descrito por un tipo de MIME.

p. Tipo ResourceContext: La definición del esquema de la estructura ResourceContext se extiende de la definición común MimeResponse sin campos adicionales.

q. Tipo ResourceResponse: Contiene campos para el retorno de los diversos temas en respuesta a una invocación GetResource.

r. Tipo MarkupContext: La definición del esquema de la estructura MarkupContext se extiende de la definición común MimeResponse, agregando campos relativos al retorno de un markupPortlet.

s. Tipo MarkupResponse: Contiene campos para el retorno de los diversos ítems en respuesta a una invocación getMarkup.

t. Tipo EventPayload: Proporciona un contenedor para los datos que son asignados a un evento.

u. Tipo Event: Proporciona las referencias de retorno a la carga útil QName y a los tipos de datos proporcionados por el EventDescription.

v. Tipo UpdateResponse: Normalmente contiene los elementos devueltos por el performBlockingInteraction o el handleEvents.

w. Tipo BlockingInteractionResponse: Contiene varios ítems performBlockingInteraction que pueden ser retornados.

x. Tipo HandleEventsFailed: Contiene el índice 0-base de un evento en el arreglo de eventos entrantes que no pudieron ser procesados completamente por el Producer, y la razón por la cual fallaron.

y. Tipo HandleEventsResponse: Contiene varios ítems handleEvents que pueden ser retornados.

z. Tipo StateChange: Este tipo de estructura es una restricción sobre el tipo de cadena que se ve limitada a los valores "ReadWrite", "cloneBeforeWrite" o "readOnly".

aa. Tipo UploadContext: Contiene campos específicos para cargar datos al portlet.

bb. Tipo InteractionParams: Contiene campos específicos que invocan la operación perfomBlockingInteraction.

cc. Tipo EventParams: Contiene campos específicos que invocan la operación handleEvents.

dd. Tipo User Profile: Esta estructura es la que contendrá información relevante y personal de los End-Users para realizar operaciones.

ee. Tipo UserContext: Suministra datos específicos del End-User a las operaciones.

Bibliografía:

http://docs.oasis-open.org/wsrp/v2/wsrp-2.0-spec-os-01.html
http://docs.oasis-open.org/wsrp/v2/wsrp-2.0-spec-os-01.pdf
http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/