

Perfiles de Usuario Generados de Acuerdo a las Preferencias de Usuario y al Contexto de Uso

User Profiles Generated According to User Preferences and Context of Use

Angela Carrillo Ramos, PhD. ¹, Marlène Villanova-Oliver, PhD. ², Jérôme Gensel, PhD. ² y Hervé Martin, PhD. ²

¹Pontificia Universidad Javeriana, Bogotá, Colombia

²Laboratorio LIG - Equipo STEAMER, Francia

angela.carrillo@javeriana.edu.co, {villanov, gensel, martin}@imag.fr

Recibido para revisión 28 de Noviembre de 2007, aceptado 14 de Febrero de 2008, versión final 28 de Febrero de 2008

Resumen—Con el fin de adaptar la información a un usuario nómada equipado con un *Dispositivo Móvil (DM)*, un *Sistema de Información (SI)* debe tener en cuenta datos relacionados tanto con el *contexto de uso* como con las *preferencias de usuario*. Los datos relacionados con el *contexto de uso* proveen en particular una descripción de las condiciones (temporales, espaciales, hardware, etc.) bajo las cuales los usuarios acceden los *SI*. Los datos concernientes a las *preferencias de usuario* se orientan a expresar qué es lo que el usuario desearía obtener del sistema considerando diferentes aspectos (funcionalidades, contenido, despliegue, etc.). En este artículo se presenta una propuesta para la adaptación de *SI* la cual se basa en la generación de un *Perfil Contextual de Usuario (PCU)*. Tal perfil se compone exclusivamente de *preferencias de usuario* que pueden ser satisfechas considerando el *contexto de uso* para la sesión actual. Este artículo presenta un formalismo de tres tipos de *preferencias de usuario (actividad, resultado y despliegue)*. Luego describe el proceso propuesto para la generación del *PCU*. Finalmente se presenta un ejemplo que ilustra la propuesta así como algunos elementos para el diseño e implementación de una herramienta que la utiliza.

Palabras Clave—Sistemas y Organización de Información, Preferencias de Usuario, Adaptación, Perfil de Usuario, Contexto.

Abstract—In order to adapt information for a nomadic user equipped with a *Mobile Device (MD)*, an *Information System (IS)* must take into consideration data that deals with on the one hand, the *context of use* and on the other hand, *user preferences*. Data about *context of use* provides in particular a description of the conditions (temporal, spatial, hardware, etc.) under which users accesses the *IS*. Data about *user preferences* aims at expressing what the user would like to obtain from the system considering different aspects (functionalities, content, display, etc.). This paper presents an approach for the adaptation of *IS*, which rests on the

generation of a *Contextual User Profile (CUP)*. Such a profile is made up only of the *user preferences* which can be satisfied considering the current *context of use*. This paper presents formalism for three different types of *user preferences (activity, result and display)*. Then, it describes the process proposed for generation of the *CUP*. An example illustrating our proposition is given. We also give the main elements for the design and implementation of a tool supporting our approach.

Keywords—Information Systems and Organization, User Preferences, Adaptation, User Profile, Context of Use.

I. INTRODUCCIÓN

Teóricamente, las aplicaciones que se ejecutan en *Dispositivos Móviles (DM)* permiten a usuarios nómadas consultar datos en cualquier momento y en cualquier lugar. Esta es la idea subyacente de la *Computación Ubicua* [18]. Los *DM* pueden ser usados para acceder *Sistemas de Información (SI)* o para ejecutar aplicaciones simples que manejan solamente algunos archivos localizados en un *DM* (por ejemplo, el *Manejador de Archivos* de un teléfono celular, o aplicaciones para manejar listas de contactos o álbumes de fotos). El problema de la adaptación de aplicaciones ejecutándose en *DM* puede ser considerado desde diferentes puntos de vista. Uno de ellos consiste en definir a *qué* tiene que adaptarse una aplicación. Por ejemplo, una aplicación puede ser adaptada teniendo en cuenta las características personales de usuario, sus preferencias, su experiencia, su cultura, su historia en el sistema, su localización actual, etc. y/o las características del dispositivo del acceso, red, etc. Estos diferentes criterios son generalmente (y algunas

veces en diferentes formas) agrupados para generar los *perfiles de usuario* y/o *modelos de contexto* [5] [11] [17]; ambos constituyen las bases para la adaptación de información de acuerdo al contenido de la misma y al *DM*.

Cuando los usuarios nómadas acceden un *SI* utilizando un *DM*, uno de los cambios en la adaptación es que el valor de los elementos pertenecientes al *contexto de uso*, puede evolucionar durante una sesión [17]. De acuerdo a Tamine *et al.* [17], el *contexto de uso* es un conjunto de elementos tales como la localización, el tiempo de conexión, la aplicación actual y, las intenciones del usuario durante una sesión de búsqueda de información. Más aún, un proceso de adaptación puede utilizar *preferencias de usuario* definidas para una sesión dada o para todas las sesiones. Los diseñadores de *SI* deben proveer a los usuarios con herramientas que los "empoderen" para expresar sus *preferencias* con el fin de: *i*) escoger y clasificar la información obtenida del *SI*; *ii*) especificar lo que los usuarios quieren llevar a cabo en el *SI* (es decir, actividades de un usuario en el sistema tales como consulta o modificación de datos); *iii*) indicar al sistema la manera en la que ellos quieren que la información sea desplegada en sus *DM*. Estas herramientas le permitirían al sistema conocer las *preferencias de usuario* con el fin de adaptarles la información. En este artículo, una "*preferencia de usuario*" se define como un conjunto de descripciones incluyendo: las *actividades* que un usuario planea llevar a cabo en el sistema (*e.g.*, consulta de datos) y la manera en la que es ejecutada (*e.g.*, secuencial, concurrente, condicional), el tipo y el orden de los resultados de estas *actividades* (*e.g.*, el "*contenido*"), y la manera en la cual el usuario quiere que la información sea desplegada en el *DM* (especificación del formato esperado -imagen, texto - con sus características).

El objetivo de este trabajo es proveer a un usuario nómada la información adaptada con respecto al *contexto de uso* de la sesión actual. El término "*contexto de uso*" hace referencia al conjunto de datos que permiten la caracterización de la interacción entre el usuario y el sistema. El *contexto de uso* está compuesto de una representación de diferentes elementos tales como las actividades del usuario en el *SI*, las características del *DM* utilizado y la localización o momento en el cual el usuario se conecta. Uno de los cambios a ser tenido en cuenta por los diseñadores de *SI* en cuanto al *contexto de uso* de un *usuario nómada* consiste en que los elementos de dicho *contexto* son variables y en constante evolución [17]. Para propósitos de adaptación, los mecanismos de representación y de explotación de este *contexto de uso* deben ser capaces de soportar esta evolución. El trabajo desarrollado por Kirsch-Pinheiro [12] constituye una propuesta en esta dirección pues permite describir los *contextos de uso potenciales* en los cuales el sistema sabe cómo reaccionar, es decir, conoce con respecto a qué adaptar la información. Esta propuesta consiste en aplicar

un algoritmo de comparación de la descripción del *contexto de uso* actual con respecto a la descripción de contextos potenciales. Con la correspondencia realizada (es decir, el contexto actual es asimilado a un contexto conocido), el sistema aplica acciones recomendadas para adaptar sus respuestas a este nuevo contexto. Este artículo se orienta a una adaptación del sistema que toma en cuenta el *contexto de uso* actual, adoptando una aproximación diferente, y no se utiliza una definición preliminar de contextos potenciales. En su lugar, se reemplaza esto con una descripción de un conjunto de *preferencias* que traducen los deseos del usuario en cuanto a sus actividades en el sistema, los contenidos que el sistema debe entregarle y el despliegue de esta información en su *DM*. Su nivel de satisfacción es evaluado de acuerdo al *contexto de uso actual*: las *preferencias* pueden, sin embargo, ser insatisfechas si no se garantizan ciertas condiciones. Este es, por ejemplo, el caso en el que un usuario desea ver de manera privilegiada un video pero su *DM* no soporta este tipo de datos.

El tener en cuenta las *preferencias* conlleva a la adaptación de la información recuperada como resultado de una *consulta de información*. La *adaptación* de un *SI* se orienta a proveer a los usuarios de servicios e información relevantes. De acuerdo a Bouzeghoub *et al.* [5], la relevancia de información (y también de servicios) es evaluada en función de un conjunto de *preferencias* específicas de un usuario o de una comunidad. Estas *preferencias* son generalmente utilizadas para describir los intereses del usuario, el nivel deseado en la calidad de los datos o, la forma de presentarlos. Este conjunto de *preferencias* conforman un *perfil de usuario* [5] [17]. La *adaptación* de la información se orienta a entregar a los usuarios la información considerando su perfil, más precisamente, las *preferencias* incluidas en dicho perfil. Este artículo trata particularmente herramientas y mecanismos de representación de *preferencias*. La expresión "*perfil de usuario*" se refiere a un conjunto de *preferencias*. Vale la pena mencionar que otros elementos que pueden formar parte de un perfil (*e.g.*, un nivel de conocimiento del usuario o su edad) no son considerados en este trabajo. Además, este artículo se enfoca en la adaptación de la información proveniente de *SI* accedidos usando *DM*.

La **primera contribución** de este artículo consiste en formalizar la noción de *preferencia*. Este formalismo permite la representación de tres tipos de *preferencias*: *actividad*, *resultado* y *despliegue*. La **segunda contribución** es un *Algoritmo de Correspondencia Contextual (ACC)* que utiliza las *preferencias* y el *contexto de uso* para definir un *Perfil Contextual de Usuario (PCU)* de un usuario dado, durante una sesión dada. Este algoritmo examina un *perfil de usuario* y analiza las *preferencias* que lo componen a la luz del *contexto de uso* actual con el fin de seleccionar sólo aquellas que pueden ser aplicadas durante esta sesión. Por ejemplo, el resultado de la selección puede estar compuesto de las *preferencias de despliegue* consideradas en función de las características del *DM*, o de las *preferencias de actividad y de resultado* que no

1. En adelante, el término "preferencia(s)" se referirá a "preferencia(s) de usuario".

están en contravía con los derechos de acceso del usuario en el *SI*. Las *preferencias* que se retienen son los únicos componentes del *PCU*. La *tercera contribución* de este trabajo concierne la gestión de conflictos la cual no es considerada en la versión actual del *ACC*. Estos conflictos pueden ocurrir, por ejemplo, cuando las *preferencias* analizadas corresponden a diferentes deseos expresados por el usuario para la misma información. De esta manera, se han identificado ciertos *tipos de conflictos*, sus causas y la manera en la cual el sistema debe reaccionar con el fin de resolverlos, o al menos, con el objetivo de informar al usuario acerca de la existencia de estos conflictos. Entre las causas de los conflictos identificadas se pueden mencionar: *i)* la incompatibilidad entre las *preferencias* y los derechos de acceso del usuario al *SI*, y *ii)* la incompatibilidad entre las *preferencias* y las características técnicas del *DM*.

La estructura de este artículo es la siguiente: Primero se posiciona este trabajo de acuerdo a un objetivo global: la adaptación de la información de acuerdo al usuario y al *contexto de uso*. La sección III presenta una clasificación para las preferencias de usuario: *actividad, resultado y despliegue*. En la sección IV se muestra cómo el *PCU* generado mediante la ejecución del *ACC*, considera el *perfil de usuario* y el *contexto de uso* de la sesión actual. Luego, se discute acerca de la resolución de conflictos que pueden aparecer entre las *preferencias* pertenecientes al *PCU*. La sección V presenta un ejemplo orientado a explicar las etapas de representación de *preferencias* y de generación del *PCU*. En la sección VI se integra el *Sistema de Gestión de Perfiles Contextuales (SGPC)* en *PUMAS* [6], un *framework* que adapta la información a *usuarios nómadas*. Finalmente se presenta en la sección VII algunos trabajos relacionados con el presentado en este artículo antes de concluir en la sección VIII.

II. POSICIONAMIENTO DE LA PROPUESTA

En este artículo se presentan los elementos necesarios para implementar un *Sistema de Gestión de Perfiles Contextuales (SGPC)* (ver Fig. 1). Dicho sistema es un componente (que puede ser integrado en la arquitectura de un *SI* o solicitado por el *SI* como un componente externo) que opera como un componente dedicado para implementar la adaptación en el *SI*. La utilización del *SGPC* como un componente externo puede permitir que se compartan *perfiles de usuario* entre aplicaciones. El rol del *SGPC* es proveer los elementos que permitan llevar a cabo la adaptación de la información manejada por el *SI*. Dicha adaptación considera las características del usuario y su *contexto de uso*.

El *SGPC* toma como datos de entrada un *contexto de uso* y un *perfil de usuario* (este último compuesto de un conjunto de *preferencias*). Luego ejecuta el *Filtrado de Preferencias* (ver Fig. 1), considerando el *contexto de uso*, por medio de la aplicación de un *Algoritmo de Correspondencia Contextual* (ver sección IV). Sólo las *preferencias* del *perfil de usuario* que pueden ser aplicadas dado el *contexto de uso*, son retornadas

para constituir el *Perfil Contextual de Usuario (PCU)*. Dicho *perfil*, artefacto producido por el *SGPC*, constituye una representación realista de aquellas preferencias a las cuales la información debe ser adaptada, *i.e.* las preferencias aplicables de acuerdo al *contexto de uso*.

El carácter realista de esta representación refleja el hecho de que este *perfil contextual*, *i.e.* reconstruido para cada sesión del usuario, basado en las *preferencias más relevantes*, seleccionadas de acuerdo con el *contexto* de la sesión actual. La noción de *perfil de usuario* es ampliamente explotada como dato de entrada del proceso de adaptación. En esta propuesta se refleja la contextualización de este *perfil de usuario* como medio para refinar este proceso, volviéndolo dinámico y evolutivo: la adaptación debe siempre tener en cuenta la evolución del contexto (*e.g.*, un cambio del *DM*).

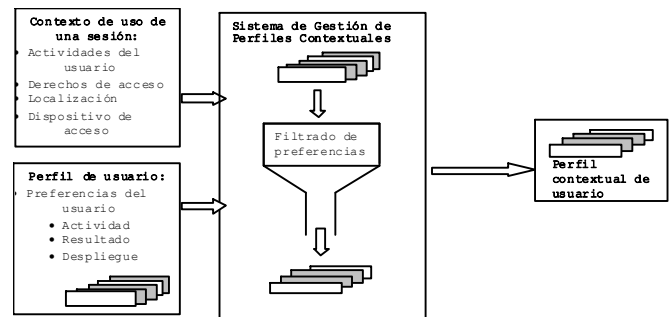


Fig. 1. Sistema de Gestión de Perfiles Contextuales

Desde el punto de vista de la adaptación de la información al usuario, este trabajo se concentra en un *PCU* compuesto de *preferencias*. Una *preferencia* se define tomando en cuenta las actividades del usuario, o aún más, la naturaleza y forma del contenido entregado. Tomando en cuenta la manera de adquisición del perfil presentada por Kassab *et al.* [10], el *SGPC* es desarrollado considerando que la información de este perfil puede ser: *i)* provista por el usuario a través de interfaces dedicadas; *ii)* definida como perfiles generales de usuario; *iii)* deducida de la historia del usuario en el sistema [4] [17].

Diversos trabajos consideran el *contexto de uso* con el fin de adaptar la información [4][17][10][5][11]. En esta propuesta se utiliza un modelo de *contexto de uso* compuesto esencialmente de cuatro tipos de información: la descripción de las *actividades* que un usuario desea ejecutar durante una sesión, los *derechos de acceso* de un usuario al *SI*, la *localización* del usuario (*e.g.*, datos obtenidos usando *GPS*) y las características del *DM* usado para conectarse durante la sesión (características definidas usando *CC/PP* [9]). En este artículo, no se presenta en detalle el modelo de *contexto de uso* tomado parcialmente del trabajo de Kirsch-Pinheiro [12]; una descripción a nivel global de este modelo es suficiente para la comprensión de esta propuesta. En la siguiente sección, se presenta la noción de

preferencia y las diferentes maneras en las cuales es usada para constituir un *PCU*.

III. PREFERENCIAS DEL PERFIL DE USUARIO

En este artículo se clasifican las *preferencias* en tres tipos: *i*) Las *preferencias de actividad* relacionadas con las actividades que un usuario quiere y puede llevar a cabo en el sistema. Desde el punto de vista del sistema, una *actividad* está compuesta de un conjunto de funcionalidades (particularmente las funcionalidades de consulta que permiten obtener información). *ii*) Las *preferencias de resultado* conciernen el contenido: un usuario puede escoger los resultados, entre aquellos obtenidos de la ejecución de las funcionalidades, y determinar el orden de presentación. *iii*) Las *preferencias de despliegue* conciernen la manera en la cual el usuario desea que la información sea desplegada en su *DM*. Esto incluye por un lado la apariencia, tipo de caracteres, *etc.*, y por otro lado, las características de los formatos de despliegue (*i.e.*, las características del video, de las imágenes, *etc.*).

Dentro de cada tipo de *preferencia*, se distinguen las *preferencias generales* (que se aplican a todas las sesiones; valor por defecto) y las *preferencias específicas* (aplicadas a la sesión actual). Una *sesión* comienza cuando un usuario se conecta al sistema y ejecuta una o más *funcionalidades*. Se define una *funcionalidad F* de nombre *fnombre*, como una tupla compuesta de una lista de parámetros de entrada ($\langle i_1, i_2, \dots, i_n \rangle$) y una de parámetros de salida ($\langle o_1, o_2, \dots, o_k \rangle$):

$$F = \text{fnombre}(\langle i_1, i_2, \dots, i_n \rangle, \langle o_1, o_2, \dots, o_k \rangle)$$

Con el fin de ilustrar la noción de *funcionalidad*, se considera el ejemplo de un *usuario-doctor* quien accede al *SI* del hospital donde trabaja, usando su *DM*. Cuando comienza una sesión, el doctor ejecuta la *actividad* "Consultar la historia clínica de un paciente". Esta *actividad* es llevada a cabo por el sistema mediante la invocación de dos funcionalidades: "Consultar exámenes médicos" y "Consultar medicamentos prescritos". Estas dos funcionalidades tienen como parámetros de entrada "el nombre del paciente" y como resultados respectivos los "exámenes médicos" y los "medicamentos prescritos" para este paciente.

Un formalismo de cada uno de los tres tipos de *preferencia* es presentado e ilustrado a continuación:

Las *preferencias de actividad* describen la forma en la que un usuario pretende llevar a cabo sus actividades en el sistema. Se define este tipo de *preferencia* de la siguiente manera:

$$\text{Preferencia_de_Actividad}(\text{tipo}, \text{criterios}, A)$$

Donde **tipo** toma como valor "general" o "específica". **criterios** es un conjunto de **criterios** de adaptación (*e.g.*, la localización y el tipo de *DM*) considerados para la ejecución de las funcionalidades en una sesión. **A** es la *actividad* que el usuario quiere ejecutar en el sistema. Esta *actividad* es expresada usando una *cadena de funcionalidades* ejecutada de manera

secuencial, concurrente o condicional. Dicha cadena es expresada usando una gramática definida en notación *BNF* ("") de la siguiente manera:

```
Actividad ::= funcionalidad [op funcionalidad]
| condicional | bucle | nil ;
op ::= secuencia | concurrente ;
secuencia ::= ";" ;
concurrente ::= "|" ;
condicional ::= "if" <condición> "then" Actividad ["else"
Actividad] "end if" ;
bucle ::= "while" <condición> "do" Actividad "end while" ;
```

Una *preferencia de actividad* para F_i , "general" y sin otro criterio de adaptación es definida por:

$$\text{Preferencia_de_Actividad}(\text{General}, (), F_i)$$

Toda funcionalidad F_i puede ser asociada a una *preferencia de resultado* como sigue:

$$\text{Preferencia_de_Actividad}(\text{General}, (), (F_i, \text{ResPref}_i))$$

Donde ResPref_i es una *preferencia de resultado* (ver sección III). En los siguientes ejemplos, por razones de claridad, las funcionalidades no están asociadas a estas *preferencias de resultado*. Una *preferencia de actividad* puede expresar la ejecución de una o más funcionalidades considerando criterios tales como su localización, las características de su *DM*, *etc.*, definidos por el usuario. Los resultados de una funcionalidad pueden ser diferentes de acuerdo a los valores de estos criterios. Por ejemplo, cuando un *doctor* consulta su *programación* en el *hospital*, obtiene como resultados las *cirugías* en las cuales participa, pero cuando la consulta en el *consultorio*, obtiene como resultados las *citas médicas* con sus pacientes. La *preferencia de actividad* y las *funcionalidades* son definidas como sigue:

```
F1 = Consultar_cirugías(<nombre_doctor, fecha>, <cirugías>)
F2 = Consultar_citas(<nombre_doctor, fecha>, <citas>)
Preferencia\_de\_Actividad(General, (localización), {if
(localización == "hospital") then F1 else (if
(localización == "consultorio") then F2)})
```

Las *preferencias de resultado* permiten a un usuario escoger y ordenar los contenidos de la información entregada como resultado de una consulta. Dichos contenidos corresponden a los resultados de la ejecución de una *funcionalidad*. Se define una *preferencia de resultado* así:

$$\text{Preferencia_de_Resultado}(\text{tipo}, F, \langle (o_1, Pdespliegue_1), (o_2, Pdespliegue_2) \dots (o_k, Pdespliegue_k) \rangle)$$

Donde *F* es el nombre de la *funcionalidad* y el último término es una lista ordenada de pares $(o_i, Pdespliegue_i)$ donde o_i representa un resultado de *F* y *Pdespliegue_i* corresponde a la *preferencia de despliegue* que es aplicada a este resultado. El orden de la lista expresa el orden de presentación de los resultados. La primera tupla de la lista corresponde al primer resultado presentado. Un usuario puede también escoger entre los resultados de la funcionalidad, aquellos que quiere obtener (*e.g.* solamente o_3 , o_5 , o_8). En este caso, la *preferencia de resultado* es definida como sigue:

```
Preferencia_de_Resultado (tipo, F, <(o3,
Pdespliegue1), (o3, nil), (o3, Pdespliegue3)>
```

Donde *nil* significa que el usuario no definió ninguna *preferencia* para el despliegue de este resultado. Una *preferencia de resultado* que expresa que un doctor desea "Consultar exámenes médicos", y que sólo quiere recibir los resultados concernientes a la curva de glicemia (como una imagen con las características de la *preferencia de despliegue Pdespliegue_imagen*) y los *exámenes médicos* asociados a la *tiroides* (definidos como *TSH*, *T4 Libre* en formato texto, respectivamente asociados a las *preferencias de despliegue Pdespliegue_texto₁* y *Pdespliegue_texto₂*) es definida así:

```
Preferencia_Resultado(General, "Consultar exámenes médicos"
<(curva_glicemia, Pdespliegue_imagen1), ("TSH", Pdespliegue_texto1), ("T4
Libre", Pdespliegue_texto2)>)
```

Las *preferencias de despliegue* describen la forma en la cual un usuario desea que su *DM* despliegue la información (*e.g.*, como una imagen). Estas preferencias son definidas como tuplas de la siguiente forma:

```
Preferencia_Despliegue(formato, {características}, substitución)
```

En esta tupla, el *formato* puede tomar como valor: "video", "texto", "imagen" y "audio" y, *características* especifican los valores tomados por los atributos que caracterizan al formato. El término *substitución* corresponde a otra *preferencia de despliegue* que el sistema tratará de utilizar en el caso en el que la *preferencia de despliegue* que está siendo definida no pueda ser satisfecha (*substitución* puede tomar como valor *nil*). La *preferencia de despliegue P₁* corresponde a una preferencia para el despliegue de un video, dadas sus dimensiones y el tipo de archivo (*e.g.*, ancho, alto, tipo):

```
P1 = Preferencia_Despliegue (video, {200,300, AVI}, P2)
```

Donde *P₂* es la *preferencia de substitución* de *P₁* y contiene las características para desplegar texto (*i.e.*, fuente, tamaño, color, tipo de archivo):

```
P2=Preferencia_Despliegue(texto, {Arial,10, azul, .doc}, nil)
```

Vale la pena anotar que las *preferencias de despliegue* que pueden ser referenciadas en las *preferencias de resultado*, corresponden en este último caso, a las *preferencias* a ser aplicadas a un formato, independientemente de un contenido particular, *e.g.*, es útil privilegiar el texto para una sesión.

IV. PERFIL CONTEXTUAL DE USUARIO (PCU)

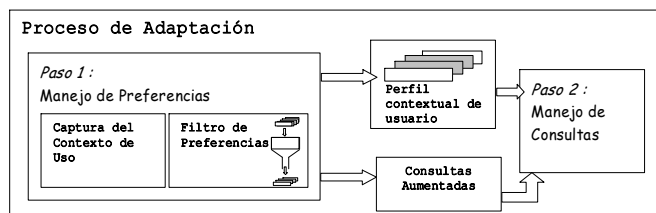


Fig. 2. Proceso de Adaptación

En esta sección se muestra cómo el *PCU* es generado con la

ayuda del *Algoritmo de Correspondencia Contextual (ACC)*, usando las *preferencias* y el *contexto de uso* de la sesión actual. También se discute la resolución de conflictos y otras incompatibilidades que pueden aparecer durante la adición de *preferencias* al *PCU*.

Con el fin de adaptar la información a las características de usuario y a aquellas de su *DM*, se adopta un proceso de dos etapas (ver Fig. 2). La primera etapa consiste en el *Manejo de Preferencias* (ver Fig. 2, *Paso 1*) y es llevada a cabo por el *Sistema de Gestión de Perfiles Contextuales (SGPC)*. En esta etapa se ejecuta tanto la captura del *contexto de uso* como la selección de *preferencias (actividad, resultado y despliegue)* que pueden ser aplicadas considerando el *contexto de uso* de la sesión actual (ver Fig. 2, *Filtro de Preferencias*). Por ejemplo, para la captura de localización, es posible usar un dispositivo *GPS* o métodos tales como el *SNMP* ("Simple Network Management Protocol") propuesto por Nieto-Carvajal *et al.* [15]. El resultado del *Paso 1* es un *PCU* (compuesto de las preferencias seleccionadas, ver sección III) y una o más *consultas* que son "*aumentadas*". Una *consulta aumentada* corresponde a la *consulta inicial* asociada a cada funcionalidad de consulta (la cual debe llevarse a cabo para ejecutar parcial o totalmente una actividad, ver sección III) a la cual el *SGPC* adiciona información sobre las *preferencias de despliegue y de resultado*. El *PCU* y las *consultas aumentadas* son los parámetros de entrada para la etapa de *Manejo de Consultas* (*Paso 2* del *Proceso de Adaptación*, ver Fig. 2). Este último paso se explica en detalle en [6]. A continuación se detalla la etapa de *Manejo de Preferencias* ejecutada por el *SGPC*.

En esta propuesta, el *Perfil Contextual de Usuario (PCU)* está compuesto de las *preferencias de actividad, resultado y despliegue* que son aplicadas teniendo en cuenta el *contexto de uso* de una sesión. Por ejemplo, sólo se retienen las *preferencias de despliegue* que pueden ser satisfechas dadas las características del *DM*, las *preferencias de resultado* que no están en contradicción con los derechos de acceso del usuario al sistema.

El $PCU(u, s, DM) = \{P_1, P_2, P_3, \dots, P_k\}$ de un usuario "*u*", que se conecta a través de un dispositivo "*DM*" durante una sesión "*s*" es el conjunto de *k* *preferencias* que son retenidas tomando en cuenta el *contexto de uso*. El *PCU* es construido mediante el análisis del conjunto $P_u = \{P_1, P_2, P_3, \dots, P_i, \dots, P_n\}$ constituido por todas las *preferencias* definidas por el usuario *u*, que incluye las *preferencias* definidas independientemente del *contexto de uso (preferencias de tipo "general")* y aquellas definidas para la sesión en cuestión (de tipo "*específica*").

El análisis del conjunto P_u efectuado por el *Algoritmo de Correspondencia Contextual (ACC)* recae en una fase preliminar de la organización de las *preferencias*. No se describe aquí de manera detallada el sistema de prioridades establecido por la organización de las *preferencias* pero se presentan sus principales puntos: *i)* Las *preferencias de actividad* son

analizadas, luego las *preferencias de resultado* y finalmente las *preferencias de despliegue*. *ii*) En cada tipo de *preferencias*, las *preferencias específicas* tienen la prioridad con respecto a las *preferencias generales*. Si no hay *preferencias específicas* establecidas, el sistema sólo considera las *generales*. *iii*) Si ninguna *preferencia* es definida, el sistema considera la historia del usuario y las características técnicas de su *DM*. En este caso, el sistema construye por sí mismo las *preferencias*.

Para cada *preferencia* P_i del conjunto completo de *preferencias* P , se aplica el *ACC* con el fin de verificar si esta *preferencia* puede ser adicionada al *PCU* (ver líneas 2 al 4 del algoritmo): el sistema analiza cada P_i (para $i \in [1, n]$) y el conjunto de *preferencias de sustitución* (PS) de P_i . Vale la pena recordar que las PS son solamente definidas para las *preferencias de despliegue*. Este análisis termina para cada P_i si: P_i puede ser satisfecha (ver línea 7) o, si una PS de P_i puede ser satisfecha (ver líneas 18 a 37) o finalmente si la PS es *nil*. El análisis de la PS de P_i comienza si P_i no puede ser satisfecha (ver línea 15). Cuando la *preferencia* P_i (respectivamente, la PS de P_i) puede ser satisfecha, P_i (respectivamente, la PS de P_i) es adicionada al *PCU* (ver líneas 8 y 22). El resto de la *cadena de PS* no es analizada (*i.e.*, son adicionadas a la *Lista de Preferencias Rechazadas*, ver líneas 13, 16, 26, 34).

Se usa una *Lista de Preferencias Rechazadas* (*LPR*) para mantener las *preferencias* que no son retenidas en el *PCU*, por dos razones: *i*) Algunas de estas *preferencias* rechazadas pueden ser aún útiles. Por ejemplo, suponga que una *preferencia* "a" es la PS de la *preferencia* "b" y "c". Si "b" es satisfecha, "a" será adicionada a la *LPR*. En el caso en el que "c" no pueda ser satisfecha, "a" sería analizada de nuevo con el fin de determinar si puede ser satisfecha. *ii*) Ciertas *preferencias* de la *LPR* pueden describir las características de un formato que debe ser desplegado en el *DM*. Por ejemplo, suponga que "a" es la PS de "b". "a" específica las características asociadas al texto y "b" especifican las características asociadas al video. Si el video es soportado por el *DM*, el sistema adiciona "a" (las características de texto) a la *LPR* privilegiando de esta manera la *preferencia* concerniente al video. Si hay alguna información que debe ser desplegada como texto, el sistema es capaz de encontrar de nuevo en la *LPR*, las características preferidas por el usuario para el texto y así puede ser aplicada.

En el algoritmo abajo, se usan las siguientes abreviaciones: **PA** es la *Preferencia Analizada*, **LPR** es la *Lista de Preferencias Rechazadas*, **NPA** es la *Nueva Preferencia Analizada*, y **PCU** es el *Perfil Contextual de Usuario*.

```
(1) i=1
(2) While (i <= n) do // Para cada  $P_i$  donde  $i \in [1, n]$  :
(3)   PA =  $P_i$  // Preferencia que es analizada en esta
           iteración
(4)   if ( $P_i \in PCU$  OR  $P_i \in LPR$ ) then
(5)     skip //el sistema analizará  $P_{i+1}$  (i.e. PA=  $P_{i+1}$ )
(6)   else
(7)     if ( $P_i$  puede ser satisfecha) then
(8)       Adicione  $P_i$  en PCU
```

```
(9)   PA= Preferencia de sustitución de  $P_i$ 
(10)  While (PA <> nil) do
(11)    NPA= PA;
(12)    PA= Preferencia de sustitución de NPA
(13)  Adicione NPA en LPR ssi NPA  $\notin$  PCU AND NPA  $\notin$  LPR
(14)  end While
(15) else
(16)  Adicione  $P_i$  en LPR ssi  $P_i \notin$  PCU AND  $P_i \notin$  LPR
(17)  PA= Preferencia de sustitución de  $P_i$ 
(18)  While (PA <> nil) do
(19)    if (PA puede ser satisfecha AND PA  $\notin$  PCU AND
        PA  $\notin$  LPR) then
(20)      NPA = PA;
(21)      PA= Preferencia de sustitución de NPA
(22)      Adicione NPA en PCU ssi NPA  $\notin$  PCU AND
        NPA  $\notin$  LPR
(23)      While (PA <> nil) then
(24)        NPA = PA;
(25)        PA= Preferencia de sustitución de NPA
(26)        Adicione NPA en LPR ssi NPA  $\notin$  PCU AND
        NPA  $\notin$  LPR
(27)      end While
(28)    else
(29)      if (PA  $\in$  PCU OR PA  $\in$  LPR) then
(30)        PA= Preferencia de sustitución de PA
(31)      else
(32)        NPA = PA;
(33)        PA= Preferencia de sustitución de NPA
(34)        Adicione NPA en LPR ssi NPA  $\notin$  PCU AND
        NPA  $\notin$  LPR
(35)      end if
(36)    end if
(37)  end While
(38) end if
(39) end if
(40) Incremente i. // i = i + 1
(41) End While
```

Con el fin de ilustrar el algoritmo de generación del *PCU*, suponga que el conjunto P_u de *preferencias* de este usuario es aquel presentado en la Fig. 3. Si P_1 es satisfecho, P_5 , P_2 y P_4 son adicionados a la *LPR*.

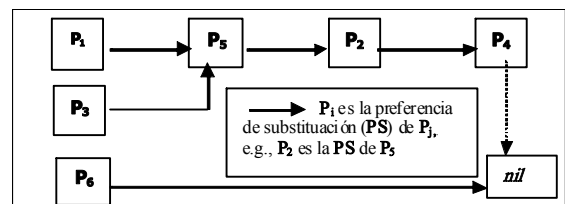


Fig. 3. Preferencias de usuario y sus preferencias de sustitución.

Cada *PCU* corresponde a una sesión dada y es generado en función del *contexto de uso* de la sesión. Por ejemplo, suponga que el *DM* del usuario sólo soporta texto. En este caso, todas las *preferencias* asociadas al despliegue de imágenes no serán satisfechas durante las sesiones en las cuales el usuario está conectado a través de este *DM*. Estas *preferencias* no aparecerán en el *PCU* generado para estas sesiones; dichas *preferencias* pertenecen a la *LPR*. La Tabla 1 presenta por ejemplo que el *PCU* está compuesto de las *preferencias* P_1 , P_3 y P_6 (P_2 , P_4 y P_5 están en la *LPR*). También se muestra que el *PCU* es "vacío" (*i.e.*, no contiene ninguna *preferencia*). La Tabla 1 presenta los *PCU* generados por el algoritmo:

TABLA 1. PERFILES GENERADOS POR EL ACC.

	PCU1	PCU2	PCU3	PCU4	PCU5	PCU6	PCU7	PCU8	PCU9	PCU10	PCU11	PCU12	PCU13	PCU14	PCU15	PCU16	PCU17	PCU18	PCU19	PCU20
P1	X	X	X	X																
P2									X	X	X	X								
P3	X		X		X	X			X	X			X	X			X	X		
P4							X	X					X		X	X				
P5																				
P6	X	X			X		X		X		X		X		X		X			X

Si no hay *cadena de preferencias de sustitución* (ninguna *preferencia* tiene *preferencias de sustitución*) en un conjunto de "*p*" de *preferencias*, luego, el número de perfiles (*P*) será calculado de la siguiente manera:

$$P = f(p) = \sum_{i=1}^p C_p^i = \sum_{i=1}^p \binom{p}{i} = \sum_{i=1}^p \frac{p!}{(p-i)! i!} \quad (1)$$

Para calcular el número de *PCU* que el algoritmo *ACC* genera considerando el conjunto de *preferencias* que contiene las cadenas de *preferencias de sustitución*, se calcula *P* usando la fórmula (1). El número de *PCU* es obtenido sustrayendo a *P* el número de combinaciones que contienen una *preferencia* con una o más de sus *preferencias de sustitución*. Por ejemplo, considere un conjunto de cuatro *preferencias* P_1, P_2, P_3 y P_4 y que P_2 es la *preferencia de sustitución* de P_1 . El número total de posibles combinaciones es luego 15 (4 combinaciones de 1 *preferencia*, 6 de 2, 4 de 3 y 1 de 4). Si el número total de combinaciones donde P_1 y P_2 aparecen juntas ((P_1, P_2) , (P_1, P_2, P_3) , (P_1, P_2, P_4) , (P_1, P_2, P_3, P_4)) es 4, luego, el número total de *PCU* generado por el algoritmo (para diferentes sesiones) es 11 (i.e., 15-4). Para el ejemplo mostrado en la Fig. 3, el algoritmo puede generar 20 diferentes *PCU* (un *PCU* por cada sesión considerando las características contextuales de esta sesión). Para un conjunto de 6 *preferencias* sin cadenas de *preferencias de sustitución*, se tienen 63 *PCU* diferentes (6 combinaciones de 1 *preferencia*, 15 de 2, 20 de 3, 15 de 4, 6 de 5 y 1 de 6), pero 43 de estas combinaciones contienen una *preferencia* con una o más de sus *preferencias de sustitución*. Entonces, solamente 20 *PCU* son generados por el algoritmo.

Se critica del *Algoritmo de Correspondencia Contextual* su relativa simplicidad en la medida en la que no se evidencia la forma en la cual ciertos problemas que ocurren son manejados durante la etapa de *manejo de preferencias* (ver Fig. 2). Vale la pena notar que la introducción de prioridades entre las *preferencias de usuario* (ver sección IV) ya es considerada como una respuesta a la gestión de conflictos. En esta sección se presentan algunos conflictos entre las *preferencias* y otras incompatibilidades que se han identificado y la forma en la que el sistema tiene que reaccionar si estos ocurren. Es importante notar que estas maneras de reaccionar no son siempre soluciones a los conflictos pero permiten a los usuarios estar informados acerca de la presencia de estos conflictos.

Entre las posibles causas de conflictos se consideran: *i) Incompatibilidad entre las preferencias y los derechos de acceso a SI*. Por ejemplo, una *preferencia* que corresponde a una solicitud de información cuyo acceso es prohibido para el

usuario. En este caso, la *preferencia* no será satisfecha y el usuario no obtendrá la información deseada. *ii) La aparición de un conflicto como consecuencia de la adición de una preferencia al PCU*. Por ejemplo, una *preferencia* puede consistir en obtener la información "*solamente*" en un formato *k* y otra *preferencia* en el *PCU* consiste en obtener la información "*solamente*" en un formato *j*, con *k* diferente e incompatible con *j*. En el caso de que existan conflictos entre *preferencias* de la misma prioridad, una solución es escoger la *preferencia* que puede ser satisfecha con respecto a los formatos soportados por el *DM*. *iii) Incompatibilidad entre las preferencias y las características y restricciones técnicas del DM*. Por ejemplo, el formato de despliegue deseado por el usuario no es soportado por su *DM*. En este caso, el sistema analiza las eventuales *preferencias de sustitución* de la *preferencia* analizada y verifica si hay una *preferencia* que puede ser satisfecha. En el caso en el que la información no pueda ser desplegada en un formato soportado por el *DM*, el sistema debe desplegar un mensaje indicando la imposibilidad para desplegar la información y la causa de esta imposibilidad. *iv) Incompatibilidad entre las preferencias relacionadas con los formatos expresados por el usuario y la disponibilidad de la información requerida* (la información no está disponible en el formato requerido). El sistema despliega la información utilizando el formato en el cual está disponible y busca, entre las *preferencias de despliegue*, las características que deben aplicarse para este formato.

Así que, para decidir si una *preferencia "puede ser satisfecha"* (ver sección IV, líneas (7) y (19)), una base de conocimiento registra los conflictos pre-identificados durante el diseño del sistema. El *Algoritmo de Correspondencia Contextual* explota este conocimiento con el fin de administrar los conflictos. Debido a que los conflictos pueden ocurrir entre las *preferencias*, el *SGPC* debe integrar un mecanismo para especificar los conflictos y resolverlos. Para su especificación, se utiliza el lenguaje propuesto por Bell [3] que permite representar eventos conflictivos cuando ocurren simultáneamente. Esta representación permite por un lado, especificar por ejemplo, cómo la adición de una *preferencia* puede producir conflictos con otras *preferencias pertenecientes al PCU*, y por otro lado, definir los criterios con el fin de decidir si una *preferencia* será incluida en el *PCU*.

V. EJEMPLO

Considere el siguiente escenario: un médico general llamado *Germán Arango* desea consultar una *ecografía* prescrita a un paciente. Las *preferencias de actividad y de resultado* del *doctor Arango* se definen así:

P_1 = El *doctor Arango* desea obtener las imágenes y el video de la ecografía.

P_1 = Preferencia_Resultado (Específica, "Consultar_ecografia", <(Resultado_eco, Pdespliegue_imagen), (Resultado_eco, Pdespliegue_video)>)

P_2 = El *doctor Arango* desea obtener el análisis de la ecografía

preparado por el radiólogo que la efectuó.

```
F1=Consultar_análisis_ecografia(<nombre_paciente,
ecografia, fecha >, <análisis_radiólogo >)
```

```
P2= Preferencia_de_Actividad(Especifica, (), F1)
```

P₃ = El *doctor Arango* quiere obtener el análisis de las ecografías previas en un formato gráfico (imágenes o video).

```
P3=Preferencia_Resultado(Especifica, "Consultar ecografia previa",
<Resultado_eco, Pdespliegue_imagen>, (Resultado_eco,
Pdespliegue_video)>)
```

P₄ = Cada vez que el *doctor Arango* solicita los resultados de una ecografía, también desea conocer el último registro (en la historia clínica del paciente) del doctor que la prescribió.

```
F2=Consultar_ecografia(<nom_pac, fecha>,<Resultado_eco>)
```

```
F3=Consultar_últ_reg(<nom_pac,dr,ecografia>,<últ_reg>)
```

```
P4 = Preferencia_de_Actividad(General, (), F2;F3)
```

P₅ = Cada vez que el *doctor Arango* solicita los resultados de una ecografía, también desea consultar las ecografías previas del mismo tipo (*e.g.*, si consulta una ecografía renal, los resultados de ecografías renales previas serán presentados).

```
F4 = Consultar_eco_previas (<nombre_paciente, tipo > ,
< resultado_eco >)
```

```
P5 =Preferencia_de_Actividad(General, (), F2;F4)
```

Suponga que las *preferencias de despliegue* del *doctor Arango* son $\{P_{1Af}, P_{2Af}, P_{3Af}, P_{4Af}\}$ donde: P_{1Af} expresa las características de las imágenes que el *doctor Arango* desea para la sesión actual. P_{2Af} expresa que si para la sesión actual su *DM* no soporta imágenes, luego la información será desplegada a manera de texto (P_{2Af} es una *preferencia de sustitución* de P_{1Af} y muestra las características del texto que serán desplegadas en caso en el que el *DM* no soporte imágenes). P_{3Af} expresa las *preferencias* del *doctor Arango* en cuanto al despliegue de texto para la sesión actual. Finalmente, P_{4Af} expresa sus *preferencias de despliegue* del texto para todas sus sesiones. El sistema analiza cada *preferencia*, de P_{1Af} a P_{4Af} . Cabe anotar que P_{1Af} , P_{2Af} y P_{3Af} tienen una prioridad más alta que P_{4Af} ya que las primeras son *preferencias específicas* y la última es una *preferencia general*. Si las imágenes especificadas en P_{1Af} son soportadas por el *DM*, esta *preferencia* es adicionada al *PCU* y P_{2Af} es adicionada a la *LPR*. De lo contrario, P_{1Af} es adicionada a la *LPR* y P_{2Af} es incluida en el *PCU*. La información solicitada en formato texto será desplegada de acuerdo a las características definidas en P_{3Af} (ya que P_{3Af} es *específica* para la sesión actual y tiene una prioridad más alta que P_{4Af}).

Puesto que las *preferencias de resultado* y *de actividad* del usuario son también restringidas por los derechos de acceso del usuario, el sistema debe verificar si el usuario puede acceder la información. Ciertos derechos de acceso toman en cuenta las reglas establecidas por la empresa a la que el usuario o el sistema pertenecen. En este ejemplo, el hospital ha establecido ciertas reglas concernientes a la consulta de los exámenes por parte de los diferentes tipos de médicos. La expresión de estas reglas y hechos es más detallada en [6]:

```
If (Examen_médico.prescrito_por.nombre == Doctor.nombre) OR
(Doctor.tipo == "especialista") OR
```

```
(Examen_médico.prescrito_por.tipo == Doctor.tipo)
then acceso_autorizado
```

```
If (Examen_médico.prescrito_por.tipo == "especialista" AND
Doctor.tipo == "médico general")
then acceso_denegado
```

En cuanto al *DM* del doctor Arango, suponga que su *PDA* (*DM₁*) soporta video e imágenes, pero no puede desplegarlos al mismo tiempo. Con el fin de definir el *PCU* del *doctor Arango*, el sistema debe analizar sus *preferencias* y por supuesto, compararlas con los formatos soportados por su *DM*. Para una sesión " s_1 ", el *PCU₁* ("*doctor Germán Arango*", s_1 , *DM₁*) está compuesto de $\{P_1, P_2, P_3, P_4$ y $P_5\}$ después del análisis efectuado por el sistema: Primero, P_1 y P_2 son satisfechas porque *DM₁* soporta video, texto e imágenes. Segundo, P_3 puede también ser satisfecha cuando se supone, por ejemplo, que las ecografías previas han sido prescritas por el *doctor Arango* o por otros médicos generales (siguiendo las reglas del hospital). Si las ecografías previas han sido prescritas por médicos especialistas, luego P_3 no se incluye en el *PCU* debido a las reglas establecidas en el hospital. La misma consideración se aplica para P_4 y P_5 . De igual manera, P_4 puede ser también satisfecha considerando que el doctor que ha prescrito esta ecografía y que ha escrito el último registro en la historia clínica del paciente es un médico general. Finalmente, P_5 es también satisfecha si las ecografías previas del mismo tipo han sido prescritas por el *doctor Arango* o por otros médicos generales. Actualmente se supone que las *preferencias de resultado*, P_6 y P_7 son respectivamente asociadas a F_3 en P_4 y F_4 en P_5 . De esta manera se definen:

```
P6=Preferencia_Resultado(Especifica, "Consultar_últ_reg",
<últ_reg,nil>)
```

```
P7=Preferencia_Resultado(Especifica, "Consultar_eco_previas",
<Resultado_eco,nil>)
```

De lo anterior se concluye que el usuario no precisó las *preferencias de despliegue* en P_6 y P_7 . El sistema cambia el valor de " \langle (Resultado_eco,nil) \rangle " por " \langle (Resultado_eco, Pdespliegue_imagen), (Resultado_eco, Pdespliegue_video) \rangle " en P_7 , ya que el *doctor Arango* solicitó en P_1 que toda la información asociada a una ecografía sea desplegada como imágenes o video (*PDespliegue_imagen* corresponde a P_{1Af}). Luego, el sistema cambia el valor de "*nil*" en P_6 por P_{3Af} debido a que el último registro de la historia clínica del paciente es recuperado en formato texto.

VI. USANDO LAS NOCIONES DE PREFERENCIAS DE USUARIO Y DE CONTEXTO DE USO EN PUMAS

Esta sección presenta cómo integrar los conceptos presentados en las secciones previas en *PUMAS* [6], un framework que recupera información distribuida entre diferentes *SI* y diversos tipos de *DM*. La arquitectura de *PUMAS* está compuesta de cuatro *SMA*: *i)* El *SMA de Conexión* provee mecanismos para facilitar la conexión de diferentes tipos de *DM* al sistema. *ii)* El *SMA de Comunicación* asegura una comunicación transparente entre el *DM* y el sistema, y aplica un *filtro de despliegue* para adaptar y desplegar la información de acuerdo a las características y restricciones del *DM* del usuario.

iii) El *SMA de Información* recibe las consultas de los usuarios, las redirecciona a los *SI* capaces de responderlas (e.g., el *SI* más cercano, el más consultado), aplica un *filtro de contenido* (con la ayuda de los agentes del *SMA de Adaptación*) tomando en cuenta el perfil de usuario en el sistema y retorna los resultados al *SMA de Comunicación*. iv) El *SMA de Adaptación* se comunica con los agentes de los otros tres *SMA* con el fin de proveerles información acerca del usuario, las características de conexión y de comunicación, las características del *DM*, etc.

Los servicios y tareas de sus agentes esencialmente consisten en manejar archivos *XML* específicos que contienen información acerca del usuario y de su *DM*. Estos agentes también poseen cierto conocimiento que les permite seleccionar y filtrar la información a los usuarios. Este conocimiento proviene del análisis de la historia del usuario en el sistema (e.g., últimas conexiones, consultas, *preferencias*). A través de *PUMAS*, el objetivo final es ofrecer un framework que además de manejar los accesos a los *SI* usando *DM*, está también a cargo del proceso de adaptación de la información de acuerdo al perfil de usuario (compuesto de sus necesidades, *preferencias*, historia en el sistema, etc.) así como a las capacidades técnicas de su *DM*. Una descripción más detallada de *PUMAS* puede encontrarse en [6].

Debido a que un usuario puede acceder uno o más *SI* utilizando su *DM* y puede llevar a cabo diferentes actividades durante sus conexiones al *SI*, el *SGPC* utiliza un conjunto de cuatro archivos *XML*, almacenados en el *DM*, con el fin de definir el *Perfil Contextual de Usuario (PCU)* y de adaptar la información: i) El *archivo sesión.XML* contiene las actividades (y sus funcionalidades) de una sesión y la información acerca de sus características: quién es el usuario conectado, cuándo comenzó la sesión y cuál es el *DM* conectado. ii) El *archivo localización.XML* contiene la localización física y lógica del usuario: ciudad, país, calle, dirección IP, orientación, velocidad. iii) El *archivo perfil.XML* contiene las *preferencias* registradas de la manera en que se presentó en la sección III. iv) El *archivo dispositivo.XML* contiene las características del *DM* usado (e.g., el tamaño de la pantalla, la memoria, la batería) para conectarse al sistema. Estos cuatro archivos son definidos usando las extensiones de *CC/PP* propuestas por [9].

Las características de adaptación de *PUMAS* recaen en un proceso de filtro de dos etapas, orientado a proveerle al usuario la información adaptada de acuerdo a sus características y a las de su *DM*. Por un lado, el *filtro de contenido* que permite seleccionar información relevante de acuerdo al *PCU* definido en el sistema. Por otro lado, el *filtro de despliegue* se aplica a los resultados del primer filtro y considera las características y restricciones técnicas del *DM* del usuario. Estos filtros son aplicados por los agentes de *PUMAS* de la siguiente manera (ver Fig. 4):

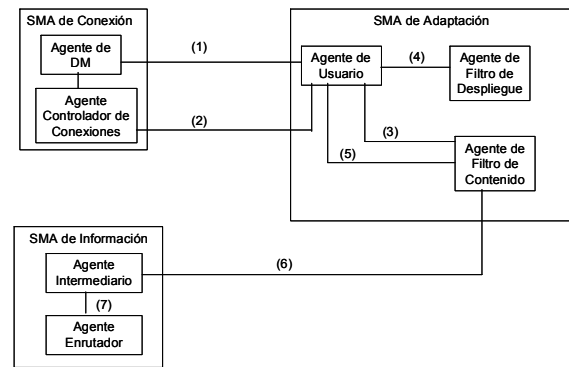


Fig. 4. Proceso de filtro de información en PUMAS.

(1) El *agente de DM* envía tanto el archivo *perfil.XML* (que contiene las *preferencias*) como el archivo *sesión.XML* (que contiene las características de la sesión) al *agente de usuario*. (2) Si no hay un archivo *sesión.XML*, el *agente de usuario* solicita esta información al *agente controlador de conexiones* (que controla todas las conexiones de los *agentes de DM*). (3) El *agente de usuario* y el *agente de filtro de contenido* verifican si hay conflictos entre las *preferencias de actividad y de resultado*. El *agente de filtro de contenido* verifica cada *preferencia de actividad y de resultado* contra los derechos de acceso del usuario en el sistema. (4) El *agente de usuario* y el *agente de filtro de despliegue* verifican si hay conflictos entre las *preferencias de despliegue*. El *agente de filtro de despliegue* verifica cada *preferencia de despliegue* con respecto a las características del *DM* (definidas en el archivo *dispositivo.XML*) y a su conocimiento acerca de problemas previos presentados cuando este tipo de *DM* se ha conectado al sistema. (5) El *agente de usuario*, el *agente de filtro de contenido* y el *agente de filtro de despliegue* verifican si hay conflictos entre las *preferencias de actividad y de resultado* y, las *preferencias de despliegue*. Si hay problemas con respecto al formato soportado por el *DM*, el *agente de usuario* y el *agente de filtro de despliegue* cambian las *preferencias de resultado* (el campo concerniente al formato preferido). La información acerca de los formatos soportados es extraída del archivo *dispositivo.XML* y de la *base de conocimiento* del *agente de filtro de despliegue*. Cuando el campo "formato" de las *preferencias de resultado* es cambiado por uno de los formatos soportados por el *DM*, el *agente de filtro de despliegue* busca si hay *preferencias de despliegue* relacionadas con el nuevo formato para conocer las características preferidas por el usuario con respecto a este formato. (6) El *agente de usuario* selecciona las *preferencias de actividad y de resultado* y las envía al *agente de filtro de contenido*. Luego, el *agente de usuario* selecciona las *preferencias de despliegue* y las envía al *agente de filtro de despliegue*. Finalmente, el *agente de usuario* genera el *PCU* (ejecutando el *Algoritmo de Correspondencia Contextual*) con las *preferencias* que no generan conflictos y envía este perfil al *agente de filtro de contenido*. Este último almacena un registro de los *PCU* en el sistema. (7) Cuando el *agente intermediario*

recibe una consulta de un usuario, le solicita el *PCU* al *agente de filtro de contenido* y adiciona esta información a la consulta. El *agente intermediario* adiciona las *preferencias* a las consultas y envía esta *consulta aumentada* (*i.e.* "*consulta original*" + "*contexto de uso*" + "*preferencias*") al *agente enrutador* que es el encargado de analizar la consulta y de redireccionarla a las fuentes de información capaces de responderla. En conclusión, las funcionalidades del *SGPC* son ejecutadas en *PUMAS* por los agentes del *SMA de Adaptación*.

VII. TRABAJOS RELACIONADOS

Teniendo en cuenta la definición del *perfil de usuario*, diversos trabajos han sido desarrollados tomando en consideración los *intereses del usuario* [4][17][10], su *historia* en el *SI* [4] [17], sus *preferencias* [17][5][11] para adaptarles la información. Algunos trabajos tales como los presentados en [5] y [17] definen el *perfil de usuario* de una manera multidimensional: el trabajo presentado en [17] expone un perfil de dos dimensiones, representado por las *consultas anteriores* y las *necesidades* recurrentes de información del usuario. El trabajo expuesto en [5] propone un modelo genérico de perfiles compuesto de seis dimensiones: *i) datos personales; ii) intereses del usuario; iii) calidad esperada; iv) preferencias; v) seguridad; vi) historia del usuario*. Sin embargo, estos trabajos no detallan el mecanismo de representación del *contexto de uso* ni de las *preferencias de un usuario nómada* con el fin de adaptarle la información.

Diversos trabajos toman en cuenta la noción de *contexto de uso* con el fin de adaptar la información a *usuarios nómadas* [1], [13] [16]. Sin embargo, estas propuestas no son enteramente satisfactorias. Por ejemplo, el trabajo presentado en [13] no considera ciertas características del *contexto de uso* (tales como la localización del usuario, las características del *DM*), o sus *preferencias*. El sistema *PIA* [1] limita el manejo de información a formato texto y no considera la localización del usuario. Trabajos tales como [1] y [16] no consideran la adaptación de la información considerando las características del *DM*. Ciertos trabajos basados en agentes tales como *MADSUM* [7] y *AmbieAgents* [14] tienen mecanismos específicos para adaptar la información al usuario considerando las *preferencias*, en el caso de *MADSUM*, y su *contexto*, en el caso de *AmbieAgents*. Sin embargo, estas propuestas no especifican una representación del *contexto de uso* ni de las *preferencias* para adaptar la información.

VIII. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se propuso un formalismo y mecanismos para el *manejo de las preferencias* de un *usuario nómada*, con el fin de adaptarle la información entregada por un *Sistema de Información* de acuerdo a sus necesidades y al *contexto de uso* de la sesión actual (*i.e.*, localización del usuario, sus características y aquellas de su *DM*). Se clasificaron las *preferencias de usuario* en tres tipos: *actividad, resultado y*

despliegue. De igual manera se presentó el *Algoritmo de Correspondencia Contextual (ACC)* que define el *Perfil Contextual de Usuario (PCU)* tomando en cuenta los tres tipos de *preferencias* expresadas por el usuario. Con el fin de generar el *PCU* para una sesión, la ejecución de este algoritmo considera la información del *contexto de uso* de dicha sesión y de las *preferencias de usuario*. También se identificaron algunos tipos de conflictos que pueden ocurrir durante la generación del *PCU* y por ello se propusieron algunas formas de solucionarlos o al menos de informarle al usuario que tales conflictos han ocurrido. Finalmente se ilustró la representación de las *preferencias de usuario* y la definición del *PCU* por medio de un ejemplo hospitalario.

Actualmente se mejora el *Algoritmo de Correspondencia Contextual* que genera el *PCU*. Esta versión considera especialmente los eventuales conflictos que podrían presentarse en el momento de la adición de una *preferencia de usuario* al *PCU* (conflictos entre una *preferencia* y aquellas pertenecientes al *PCU*). En cuanto a la resolución de conflictos, actualmente se estudian diversos métodos (*e.g.*, *Argumentación* [2], *Branching rules* [8]) para implementarlos en el *Algoritmo de Correspondencia Contextual*.

REFERENCIAS

- [1] S. Albayrak, S. Wollny, Varone, N., A. Lommatzsch and D. Milosevic. "Agent Technology for Personalized Information Filtering: the PIA-System", in Proceedings of the 20th Annual ACM Symposium on Applied Computing (Santafe, New Mexico, USA, March 13-17, 2005), SAC 2005, ACM Press, New York, NY, pp. 54-59.
- [2] L. Amgoud and S. Parson. "An Argumentation Framework for Merging Conflicting Knowledge Bases". In Proceedings of the Logics in Artificial Intelligence, European Conference (Cosenza, Italy, September, 23-26, 2002). JELIA 2002. LNCS, Vol. 2424, Springer, pp. 27-37.
- [3] J. Bell. "Simultaneous Events: Conflicts and Preferences". In Proceedings of the 6th Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, (Toulouse, France, Sept.19-21, 2001). ECSQARU 2001. LNCS, Vol. 2143, Springer, pp. 714-725.
- [4] A. Birukov, E. Blanzieri and P. Giorgini. "Implicit: An Agent-Based Recommendation System for Web Search". In Proc. of the 4th Int. Conference on Autonomous Agent and MAS (Utrecht, Netherlands, July 25-29, 2005). AAMAS 2005. ACM Press, New York, NY, pp. 618-624.
- [5] M. Bouzeghoub, and D. Kostadinov. "Personnalisation de l'information: aperçu de l'état de l'art et définition d'un modèle flexible de profiles. In Proceedings of CORIA 2005 (Grenoble, France, March 9-11, 2005), pp. 201-218.
- [6] A. Carrillo Ramos, M. Villanova-Oliver, J. Gensel, and H. Martin. "Knowledge Management for Adapted Information Retrieval in Ubiquitous Environments". In Proc. of the 2nd International Conference on Web Information Systems and Technologies (Setubal, Portugal, April 11-13, 2006). Webist 2006. LNBP, Vol. 1, Springer, pp. 84-96.
- [7] T. Harvey, K. Decker and S. Carberry. "Multi-Agent Decision Support Via User Modeling". In Proceedings of the 4th International Conference on Autonomous Agent and MAS (Utrecht, Netherlands, July 25-29, 2005). AAMAS 2005. ACM Press, New York, NY, pp. 222-229.

- [8] M. Herbstritt, and B. Becker. "Conflict-Based Selection of Branching Rules". In Proc. of the 6th International Conference on Theory and Applications of Satisfiability Testing (Santa Margherita Ligure, Italy, May 5-8, 2003). SAT 2003. LNCS, Vol. 2919, Springer, pp. 441-451.
- [9] J. Indulska, R. Robinson, A. Rakotonirainy, and K. Henriksen "Experiences in Using CC/PP in Context-Aware Systems". In Proc. of the 4th International Conference MMD (Melbourne, Australia, Jan. 21-24, 2003). MMD 2003. LNCS, Vol. 2574, Springer, pp. 247-261.
- [10] R. Kassab, J.C. Lamirel, and E. Nauer. "Une nouvelle approche pour la modélisation du profil de l'utilisateur dans les systèmes de filtrage d'information basés sur le contenu: le modèle de filtre détecteur de nouveauté". In Proceedings of CORIA 2005 (Grenoble, France, March 9-11, 2005), pp. 185-200.
- [11] S. Kechid, and H. Drias. "Accès personnalisé à de multiples serveurs d'informations". In Proceedings of CORIA 2006 (Lyon, France, March 15-17, 2006), pp. 249-254.
- [12] M. Kirsch-Pinheiro. "Adaptation Contextuelle et Personnalisée de l'information de Conscience de Groupe au sein des Systèmes d'Information Coopératifs". Doctoral Thesis. Université Joseph Fourier, Grenoble, Septembre 2006 (En français).
- [13] K. Kurumatani. Mass User Support by Social Coordination among Citizen in a Real Environment. In Proceedings of International Workshop on Multi-Agent for Mass User Support (Acapulco, Mexico, Aug. 10, 2003). MAMUS 2003. LNAI, Vol. 3012, Springer, pp. 1-16.
- [14] T. Lech and L. Wienhofen. "AmbieAgents: A Scalable Infrastructure for Mobile and Context-Aware Information Services". In Proc. of the 4th Int. Conference on Autonomous Agent and MAS (Utrecht, Netherlands, July 25-29, 2005). AAMAS 2005. ACM Press, NY, NY, pp. 625-631.
- [15] I. Nieto-Carvajal, J.A. Botia, P.M. Ruiz, A.F. Gomez-Skarmeta. "Implementation and Evaluation of a Location-Aware Wireless Multi-Agent System". In Proc. of the International Conference on Embedded and Ubiquitous Computing (Aizu-Wakamatsu, Japan, August 25-27, 2004). EUC 2004. LNCS, Vol 3207, Springer, pp. 528-537.
- [16] A. Sashima, N. Izumi, and K. Kurumatani. "Bridging Coordination Gaps between Devices, Services, and Humans in Ubiquitous computing". In Proc. of the Workshop on Agents for Ubiquitous Computing (NY, USA, July 20, 2004). UbiAgents04. <http://www.ift.ulaval.ca/~mellouli/ubiagents04/> (UA: Oct. 2007).
- [17] L. Tamine, and W. Bahsoun. "Définition d'un profil multidimensionnel de l'utilisateur". In Proceedings of CORIA 2006 (Lyon, France, March 15-17 th, 2006), pp. 225-236.
- [18] <http://www.w3.org/TR/webont-req/> (Ultimo Acceso: Oct. 2007).

Laboratorio de Informática de Grenoble (Francia). Sus temas de interés son los sistemas de información multimedia, la adaptación de la información y la representación de conocimiento teniendo en cuenta aspectos espacio-temporales.

Jérôme Gensel. DEA en Informática del Instituto Nacional Politécnico de Grenoble, Francia (1990). Doctorado en Informática (1995) y Habilitación para dirigir proyectos de Investigación (2006) de la Universidad Joseph Fourier, Grenoble, Francia. Actualmente se desempeña como Profesor Asociado en la Universidad Pierre Mendes France (UPMF) y profesor de Informática en la UFR Ciencias del Hombre y de la Sociedad (UFR SHS), en el departamento de Informática y Matemáticas y Ciencias Sociales (IMSS). Además es investigador en el equipo STEAMER del Laboratorio de Informática de Grenoble (Francia). Sus temas de interés son los sistemas de información web, la adaptación de la información, sistemas de información geográficos, representación de conocimiento, movilidad y ubicuidad.

Hervé Martin. Doctorado en Informática (1991) y Habilitación para dirigir proyectos de Investigación (2000) de la Universidad Joseph Fourier, Grenoble, Francia. Actualmente se desempeña como Profesor Asociado de la Universidad Joseph Fourier y del Instituto de Geografía Alpina en Grenoble, Francia. Director e investigador del equipo STEAMER y director Adjunto del Laboratorio de Informática de Grenoble (Francia). Sus temas de interés son los sistemas de información multimedia pervasivos y los sistemas de información geográficos.

Angela C. Carrillo Ramos. Ingeniera de Sistemas y Computación (1996) y Magister en Ingeniería de Sistemas y Computación (1998) de la Universidad de los Andes, Bogotá, Colombia. Doctorado en Informática de la Universidad Joseph Fourier, Grenoble, Francia (2007). Asistente de Investigación y Profesora de Cátedra (1996-1997), y Profesora Asistente (1998-2003) de la Universidad de los Andes. Actualmente es Profesora Asociada e Investigadora de los grupos ISTAR y SIDRE de la Pontificia Universidad Javeriana de Bogotá. Su trabajo se ha enfocado en el acceso a sistemas de información a través de dispositivos móviles utilizando la tecnología de agentes. Otros de sus intereses son la personalización de la información en ambientes nómadas teniendo en cuenta el usuario, su dispositivo de acceso y su contexto de uso.

Marlène Villanova-Oliver . Doctorado en Informática del Instituto Nacional Politécnico de Grenoble, Francia (2002). Actualmente se desempeña como Profesora Asistente en la Universidad Pierre Mendes France (UPMF) y el Instituto Universitario de Tecnología (IUT2), Grenoble, Francia. Además es investigadora en el equipo STEAMER del

Universidad Nacional de Colombia Sede Medellín

Facultad de Minas



Escuela de Ingeniería de Sistemas

Misión

La misión de la Escuela de Ingeniería de Sistemas es fomentar y apoyar la generación o la apropiación de conocimiento, la innovación y el desarrollo tecnológico en el área de ingeniería de sistemas e informática sobre una base científica, tecnológica, ética y humanística.



Visión

La formación integral de profesionales desde el punto de vista científico, tecnológico y social que les permita adoptar, aplicar e innovar conocimiento en el campo de los sistemas e informática en sus diferentes aspectos, aportando con su organización, estructuración, gestión, planeación, modelamiento, desarrollo, procesamiento, validación, transferencia y comunicación; para lograr un desempeño profesional, investigativo y académico que contribuya al desarrollo social, económico, científico y tecnológico del país.



Escuela de Ingeniería de Sistemas
Dirección Postal:
Carrera 80 No. 65 - 223 Bloque M8A
Facultad de Minas. Medellín - Colombia
Tel: (574) 4255350 Fax: (574) 4255365
Email: esistema@unalmed.edu.co
<http://pisis.unalmed.edu.co/>