



Especialista en Ingeniera de Software

Computación Ubicua, Sensibilidad al Contexto y
Mashups

Autor: Ing. Guillermo Alberto Caserotto

Director: Dr. Gustavo Rossi

Universidad Nacional de La Plata

Febrero 2012

Índice

Introducción	5
Objetivos	7
Organización del Trabajo	8
Capítulo I	9
Computacion Ubicua	9
1. Computación ubicua	9
2. Definición formal de Computación Ubicua.....	10
2.1 Primera aproximación.....	10
2.2 Segunda aproximación	11
3. Tendencias fundamentales de la informática	12
3.1 La era del Mainframe.....	13
3.2 La era del Computador Personal.....	14
3.3 Internet y Computación Distribuida	14
3.4 La era de la Computación Ubicua	15
3.5 Computación Ubicua como una tendencia fundamental	17
4. Tecnología Calma	18
4.1 Periferia	18
4.2 Características de la Tecnología Calma	19
5. Ejemplo de un sistema de Computación Ubicua	20
AULA: Un Sistema Ubicuo de Enseñanza-Aprendizaje Colaborativo	20
5.1 Un modelo ubicuo para la enseñanza de idiomas	20
5.2 Realización del modelo.	21
5.2.1 Arquitectura para soportar computación ubicua y colaboración en el aula	22
5.2.2 Coordinador de sesiones.	24
5.2.3 Dispositivos móviles.	24
5.2.4 El software desarrollado para los dispositivos móviles.....	25
5.3 Conclusión sobre el sistema	28
6. Computación ubicua, dispositivos y tendencias de investigación.	29
Capítulo II	32
Sensibilidad al Contexto	32
1. Que es el Contexto.....	32
1.1 Definición de Contexto.....	34
2. Informática Sensible al Contexto.	34
2.1 Selecciones próximas (Proximate Selection)	37

2.2 Reconfiguración Contextual automática	38
2.3 Comandos Contextuales	40
2.4 Acciones contextuales disparadoras “Context-triggered actions”	41
3. Obteniendo información del contexto.	42
3.1 Tipos de información contextual	42
3.1.1 Ubicación.....	42
3.1.2 Tiempo	44
3.1.3 Objetos o dispositivos cercanos.....	44
3.1.4 Ancho de banda	44
3.1.5 Otro tipo de información contextual.....	45
3.1.6 Obteniendo información de contexto de alto nivel.....	45
3.1.7 Obteniendo información de contextos cambiantes.....	46

Capítulo III 47

Mashup 47

1. Definición	47
1.1 Mashup	47
2. Clasificación de los Mashups.	47
2.1. Mashup empresariales	47
2.2. Mashup de Consumidores	48
2.3. Mashup de Datos	48
2.4. Mashup de Mapas	48
2.5. Mashup de Videos y Fotos.....	48
2.6. Mashup para Búsquedas y Compras.....	49
2.7. Mashup de Noticias	49
3. Tecnologías relacionadas	50
3.1 Presentación.....	50
3.2 Servicios Web	51
3.3 Datos.....	52
4. Arquitectura de los Mashups	53
4.1 Basados en el Servidor	53
4.2 Basados en navegador.....	55
5. Problemas en la integración de datos.....	57
5.1 Diferencia entre el texto y dato.....	57
5.2 Identificación de Objetos	57
5.3 Niveles de Abstracción	57
5.4 Calidad de Datos.....	58
6. Del escritorio a dispositivos móviles.....	59
7. Mashup en dispositivos móviles.....	60

Capitulo IV 63

Uniando Criterios..... 63

1. Mashups sensibles al Contexto.....	63
---------------------------------------	----

2. Sensibilidad al Contexto y aplicaciones en la Web	64
2.1 Características de las aplicaciones web sensibles al contexto.....	66
2.2 Aproximaciones para suministrar información contextual.....	66
3. Sensibilidad al Contexto en las Telecomunicaciones.....	67
4. Integración de datos en aplicaciones Mashups sensibles al contexto.....	71
Conclusión	73
Trabajos Futuros.....	74
Bibliografía	75

Introducción

Hace no menos de una década el paradigma predominante en la informática era una Computadora Personal utilizada por una persona (o varias) y generalmente conectada a Internet. Con la proliferación de los dispositivos móviles en los últimos años esta tendencia cambió, a tal punto que ya se habla del fin de la era del Computador Personal como tendencia predominante.

En el año 2010 las ventas de los dispositivos móviles superaron a las ventas de las PC.

Según IDC¹ las ventas de celulares en los últimos tres meses del 2010 ascendieron a 100.9 millones (un incremento del 87.2% por sobre el mismo período en el 2009), mientras que en el mismo período se vendieron 92.1 millones de PC y Laptops².

Luego en el primer cuatrimestre del año 2011 se vendieron 371.8 millones de unidades mientras que en el mismo periodo del año 2010 se vendieron 310.5 millones, por lo tanto, según IDC las ventas de teléfonos inteligentes aumenta casi un 20% cada año³.

También en ese mismo periodo Oracle anuncia en su informe “Opportunity Calling” que el 69% de los usuarios de teléfonos móviles utiliza teléfonos inteligentes⁴.

Lo mismo ocurre con las tabletas. Ya en el mundo empresarial empiezan a ser más utilizadas que los teléfonos inteligentes, según la encuesta que realizó CISCO el 25 de Enero de 2012, a nivel global, los departamentos de TI reportan que los empleados realizan un pedido de tabletas por cada tres pedidos de teléfono inteligente en la actualidad⁵.

Esta tendencia está más que clara, en los próximos años se utilizarán más

1 www.idc.com

2 http://www.theregister.co.uk/2011/02/08/idc_smartphone_pc_shipments

3 <http://www.idc.com/getdoc.jsp?containerId=prUS22808211>

4 http://www.diarioti.com/noticia/69_por_ciento_de_usuarios_moviles_utilizan_smartphones/30775

5 http://www.diarioti.com/noticia/Encuesta_mundial_destaca_entusiasmo_por_tabletas_en_el_entorno_empresarial/31001

dispositivos móviles que otro tipo de computadora y ya estamos presenciando una revolución en el mundo de las comunicaciones y la relación que tenemos con dichos dispositivos.

Esto que está ocurriendo hoy en día, quizás con la presentación del iPhone en Junio de 2007 y luego el iPad en Enero de 2010 como tecnologías disruptivas, no era novedad para Mark Weiser, un investigador de Xerox PARC en Palo Alto California, quien en su laboratorio en la década del 80 ya hablaba de una nueva era en la cual muchos dispositivos iban a ser utilizados por varias personas saliendo del paradigma PC-Usuario y yendo al concepto de “computadoras en todos lados” interconectadas entre sí.

Él pensaba en una nueva generación de computadoras e interacción con ellas, en la cual cada persona podría estar interactuando con cientos de computadoras en todo el entorno físico.

También propuso varios modelos de dispositivos de diferentes tamaños y formas tal como está ocurriendo en este momento. A esta nueva tendencia la llamo “Computación Ubicua”.

El presente trabajo pretende ahondar en dicho concepto tan interesante como también en una de las ramas de investigación que hará posible cada vez más la interconexión entre los dispositivos y enriquecimiento de la experiencia de usuario, la informática sensible al contexto, en inglés, Context-Aware Computing.

También trataré sobre una palabra que hoy en día está de moda, Mashup, una innovadora manera de utilizar Internet, permitiendo al usuario y desarrollador no ser un actor pasivo dentro de ella sino que todo lo contrario, ser un sujeto activo, un generador de contenidos.

Esto también es una nueva manera de relacionarse con la tecnología. Aunque los Mashup se utilizan principalmente en la Web, ya se está investigando la utilización de este enfoque en dispositivos móviles permitiendo una interconexión mucho más interesante entre dispositivos.

Objetivos

Los tres conceptos mencionados en la introducción, que considero están relacionados, según su nivel de abstracción, como la computación ubicua que abarca todo, la sensibilidad al contexto dando una experiencia más rica al usuario y Mashup como manera de interactuar entre ellos es el objeto del presente trabajo.

Por lo tanto los objetivos propuestos son:

1. Realizar un compendio sobre el estado del arte de las distintas tecnologías y herramientas utilizadas para el desarrollo de aplicaciones Mashup y aplicaciones sensibles al contexto.
2. Presentar los conceptos teóricos que dieron soporte y dirección para desarrollo de las tecnologías mencionadas como ser el significado de la computación ubicua y el gran aporte que hizo a la comunidad informática.

Organización del Trabajo

El presente trabajo esta dividido en tres secciones bien diferenciadas. El primer capítulo tratará sobre la Computación Ubicua como tendencia fundamental de la informática.

En el segundo se abarcará el concepto de Contexto, Sensibilidad al Contexto, dispositivos Sensibles al Contexto y diferentes aproximaciones para el desarrollo de la Informática Sensible al Contexto.

En el capítulo tres se describen los Mashups tradicionales y Mashups para dispositivos móviles.

En el cuatro se unen criterios para un desarrollo de aplicaciones Mashup Sensibles al Contexto.

Y por último, se presentan las conclusiones y trabajos futuros.

Para facilitar la lectura se agregaron al pie de página la fuente de donde se obtuvo la información y también al final de la obra se encuentra toda la bibliografía consultada.

Capítulo I

Computación Ubicua

1. Computación ubicua

La idea de computación “en todos lados”, por la manera en que están hechas las computadoras, no va a ser posible, ya que se encuentran separadas y separan al usuario de la situación en general.

Por otro lado si estuviéramos rodeados de computadoras, de la manera en que están diseñadas, sería tanta la información que no podríamos manejarla.

Por estas razones Mark Weiser⁶ sugirió que los dispositivos del futuro deberían tener otros diseños tanto en hardware como en software, y ser tan efectivos que sean “invisibles” al usuario, sacándolos del foco de atención, tal como una birome lo es cuando estamos escribiendo.

Afirmó que habría que tener en cuenta todo el contexto, por ejemplo, el teclado, el peso, posición de las pantallas, etc. Para ello pensó que se necesitaba cambiar las dimensiones de las computadoras, haciéndolas de varios tamaños y formas y una nueva manera de interacción con ellos, la cual lo llamo *“tecnología calma”* (Trataremos más adelante sobre este tema).

Esta idea de nuevos tipos de dispositivos surgió de ver las herramientas que utilizamos para realizar nuestras actividades, los cuales son de varios tamaños y formas. Entonces pensó en la creación de tres tamaños de dispositivos, cada uno con un alcance particular para determinadas tareas, así que tomo como analogías a elementos de oficina para diseñarlos.

El primero fue una pantalla del tamaño de una pizarra, análogo a un pizarrón, una

6 Científico del laboratorio de Xerox Parc en Palo Alto California, creador del concepto Computación Ubicua.

cartelera o inclusive a una heladera cubierta de imanes.

El segundo tamaño fue como el de un bloc de notas, previsto no como un computador personal sino análogo a un pedazo de papel, utilizando muchos al mismo tiempo por una persona.

Y el tercero con un tamaño de una computadora pequeña, análogo a un bloc de notas individual o un Post-it.

Estos tamaños de computadoras, no los vio como una computadora personal sino que los vio como elementos que se encuentren en cada parte de la vida cotidiana. Logrando así una computación “pervasiva” o computación “ubicua”. Estos tres tipos de dispositivos los llamo, Boards, Pads y Tabs.

Casi exactamente lo que hoy conocemos como teléfonos inteligentes, tabletas y aunque no tan utilizadas Pizarras Digitales.

2. Definición formal de Computación Ubicua⁷

Para definir computación ubicua Mark Weiser presenta dos aproximaciones de lo que debería ser.

2.1 Primera aproximación

“Inspirado por sociólogos, filósofos y antropólogos en PARC, hemos tratado de tomar una mirada radical en como la computación y su interconexión debería ser. Creemos que las personas viven a través de la experiencia y su conocimiento tácito, por eso las cosas más poderosas son las que son invisibles para el usuario. Esto es un reto para todos los científicos en computación. Nuestra aproximación preliminar: Activar al mundo. Proporcionar cientos de dispositivos inalámbricos por persona, por oficina, en todas las escalas (desde pantallas de una pulgada al tamaño de una pared).

⁷ <http://www.ubiq.com/ubicomp/>

Esto ha requerido trabajar de nuevo en sistemas operativos, interfaces de usuario, redes inalámbricas, pantallas (displays) y en otras áreas más.

Llamamos a nuestro trabajo “computación ubicua”. Esto es diferente de los PDAs, dynabooks (laptop), o información en la punta de los dedos.

Esto es invisible, computación en todos lados que no vive en un dispositivo personal de ningún tipo, pero está en todos lados.”

2.2 Segunda aproximación

“Durante treinta años la mayoría del diseño de interfaces y computadoras ha seguido la línea de la maquina “espectacular”. Su gran ideal era crear una computadora tan excitante, tan maravillosa, tan interesante, que nunca pensaríamos estar sin ella.

Un camino menos transitado lo llamo “invisible”, su ideal es crear computadoras tan integradas, tan adaptables, tan naturales, que lo podemos usar sin siquiera pensar en ello. (También llame a esta noción, “computación ubicua”, y puse sus orígenes en el post-modernismo).

Creo que en los próximos 20 años el segundo camino va a ser el dominante. Pero no va a ser fácil, muy poco de la infraestructura de los sistemas actuales va a permanecer. Hemos estado desarrollando versiones de la infraestructura-del-futuro en PARC para los últimos cuatro años, en formatos de computadoras de tamaños de pulgadas, pies y yardas que llamamos Tabs, Pads y Boards.

Nuestros prototipos a veces han tenido éxito, pero la mayoría de las veces han fallado en ser invisibles. De lo que hemos aprendido, ahora estamos explorando nuevas direcciones para la ubicomp [este término se utiliza también para referirse a la computación ubicua]”

La Computación Ubicua está lejos de ser el ideal de Mark Weiser pero creo que se dirige en esa dirección.

Hoy en día un usuario puede disponer de dispositivos en varias formas y dimensiones, por ejemplo, un teléfono inteligente, tableta, notebook, ultrabook⁸, etc. Estos dispositivos no son "invisibles" pero si están cada vez más alrededor nuestro, dando una nueva experiencia, alejándonos cada vez más del paradigma original de la computadora personal y acercándonos a una "computación en todos lados".

3. Tendencias fundamentales de la informática

En este punto mencionaremos como Mark Weiser describe ciertas tendencias en la informática que han cambiado la vida de millones de personas y como llega a la conclusión que en un futuro no muy lejano vamos a vivir otra tendencia donde la relación de las personas con la informática va a cambiar drásticamente.

Según dicho autor, las olas tecnológicas importantes son aquellas que cambian radicalmente el lugar que ocupa en nuestras vidas, y afirma que lo que importa no es la tecnología misma sino la relación que tenemos con ella.

En los últimos cincuenta años de la informática existieron dos grandes eras. En primer lugar, la era del Mainframe y luego, la era del computador personal.

Hoy en día la Internet nos está llevando a través de una era de "Computación Distribuida" hacia la relación de "computación en todos lados".

Esta "computación en todos lados" la cual lo llamó *Computación Ubicua* va a requerir un nuevo enfoque en el diseño de dispositivos y aplicaciones para poder adaptar la tecnología en nuestras vidas, este enfoque lo llamó "tecnología calma".⁹

8 <http://es.wikipedia.org/wiki/Ultrabook>

9 En inglés es Calm Technology, se puede traducir también como tecnología invisible

Por lo, tanto existen cuatro grandes tendencias en la informática, que se describe en el siguiente cuadro:

Principales tendencias en la informática	
Mainframe	Muchas personas compartiendo una computadora
Computador Personal	Una persona, una computadora
Internet, Computación distribuidaen transición...
Computación Ubicua	Muchas computadoras compartidas por muchas personas

3.1 La era del Mainframe

La primera era la llamó "Mainframe" para describir la relación que tenían las personas con las computadoras que mayormente eran utilizadas por expertos y tras puertas cerradas en su laboratorio.

Cada vez que la computadora sea un recurso escaso y tenga que negociarse y repartirse con otras personas, estamos hablando de la era del Mainframe.

Hay una informática del tipo Mainframe hoy en día, por ejemplo, una Computadora Personal compartida en la oficina y la virtualización, que tienen en común que comparten un recurso escaso.

3.2 La era del Computador Personal

La segunda gran tendencia es la del Computador Personal. En 1984 la cantidad de personas utilizando PC superaba la cantidad de personas que utilizaban computadoras compartidas¹⁰.

Según Mark Weiser, la relación de la PC es intimidante, una persona tiene su computadora, con todas sus cosas e interactúa directamente con ella.

Cuando estás trabajando con la PC estas ocupado, no estás haciendo otras cosas. Algunas personas nombran sus PC y tienen una relación muy estrecha.

El comparó a la PC con un automóvil (Aunque hoy día hay una diferencia significativa de precio entre estos dos ítems)

“El automóvil es un ítem relativamente caro y especial, que mientras te lleve donde quieras ir, requiere una atención considerable para operarlo.

También como uno puede tener varios autos, también puede tener varias computadoras, para la casa, el trabajo, etc.”

Por lo tanto, cualquier computadora en la que se tenga una relación especial o que requiera de toda su atención para operarla, pertenece a la era del Computador Personal.

3.3 Internet y Computación Distribuida

Aquí se encuentran elementos de la era del Mainframe y la PC. Es una informática del tipo cliente-servidor a una escala masiva, con clientes Web (PC) y los servidores Web (Mainframe).

Aunque sea transicional, Internet hoy en día es un fenómeno masivo y esta influenciando la vida de millones de personas.

¹⁰ "Transition to the Information Highway Era" in *1995-96 Information Industry and Technology Update*.p. 2.

La última década el resultado de la interconexión masiva de información personal, de negocios y de gobierno, creó un nuevo campo, un nuevo medio, en el cual tendrá que surgir una nueva gran relación entre personas y computadoras, que será la siguiente gran tendencia o era informática.

3.4 La era de la Computación Ubicua

La tercera ola de la informática será la de la Computación Ubicua. Mark Weiser escribió sobre ello en la década de los 90 y vaticinaba que podría llegar a aparecer alrededor del 2005-2020.

Ahora sabemos que todavía estamos en una etapa de transición pero en la cual cada día más gente utiliza Smartphones, Tabletas, Notebooks, etc. y donde el acceso a Internet mediante tecnología Wifi o telefonía 3G y 4G es cada vez más común.

Una persona puede acceder a Internet en un bar, Restaurant, Shopping, etc. En su casa puede tener una PC, una Notebook, un Celular y una Tableta, todas interconectadas mediante una LAN con acceso a Internet.

Un ejemplo sencillo se puede encontrar en los dispositivos con sistema operativo Android, si se utiliza un Celular y Tableta con dicho sistema, la agenda se actualiza en los dos dispositivos, tanto el calendario como los contactos que uno tiene en su cuenta de Google.

Incluso dicho Sistema Operativo se está utilizando en relojes, por ejemplo en l'm Watch¹¹ como se muestra en la siguiente imagen.

11 http://en.wikipedia.org/wiki/l%27m_Watch



Figura 1 Reloj con sistema operativo Android

No es precisamente lo que Mark Weiser soñó pero creo que en los próximos años veremos el inicio de esta nueva era.

Esta era tendrá cientos de computadoras compartidas para cada uno de nosotros, algunas estarán dentro de lugares como paredes, sillas, autos, etc.

Mark Weiser afirma que en esta era no se hablará de los thin clientes, sino que se hablara de thin servers, que lograrán el acceso a Internet para cada electrodoméstico o equipamiento de oficina.

También destaca dos puntos claves para tomar como "presagio" del comienzo de esta nueva: Los Microprocesadores e Internet¹².

Hoy en día se pueden encontrar fácilmente 40 microprocesadores en una casa de clase media en los países desarrollados.

Se encontraran en los relojes, televisores, controles remotos, hornos eléctricos, etc. pero esto no califica como Computación Ubicua si no están interconectados.

Si se utilizan por separado estarían enmascarando a los electrodomésticos antiguos.

Pero, si los conectas entre ellos, y acá entra en juego la Internet, tendrán millones de fuentes de información con cientos de sistemas en su casa y ahí si estarían

¹² <http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm>

disponibles para realizar tecnologías para la Computación Ubicua,

Por ejemplo, relojes que corrijan su hora después de un apagón, microondas que bajen recetas de cocinas, juguetes que tengan su software este actualizado y con nuevas funcionalidades, paredes que emitan sonidos, etc.

La tecnología de la información en esta era también apuntará a problemas de la vida cotidiana, como ¿Donde están las llaves del auto?, ¿Encontraré lugar para estacionar? , ¿Estas son las zapatillas que me quería comprar?

3.5 Computación Ubica como una tendencia fundamental

¿Por qué Mark Weiser califica a la computación ubicua como tecnología fundamental?

Primero, porque es sobre las relaciones básicas del ser humano, y por eso son tendencias sobre lo que nos importa a nosotros.

Segundo, tienen las propiedades de construirse uno sobre otro.

Ninguna de las otras dos tendencias han desaparecidos, tanto las del Mainframe como las del Computador Personal, sino que cada uno se ha utilizado como base para la próxima era.

Tercero, cada una es una abundante fuente de innovación que ha requerido reconsiderar antiguos supuestos y readecuar tecnología antigua en nuevos contextos.

El cambio más interesante que implica a la era de la computación ubica es enfocarse en la "calma".

“Si las computadoras van a estar en todos lados, deberían pasar desapercibidas y eso significa diseñarlas de alguna manera para que su utilización sea serena y en control“

Cuando las computadoras se usan en un laboratorio y por expertos, la calma es relevante solo para unos pocos. También las computadoras personales están enfocadas en la interacción del usuario.

Pero cuando las computadoras estén en todos lados y uno quiera utilizar un dispositivo mientras hace otra cosa, ahí es cuando hay que re pensar los objetivos y considerar la tecnología y el contexto y toda la tecnología que este alrededor nuestro.

Según dicho investigador esto es un reto fundamental para el diseño tecnológico para dentro de los próximos cincuenta años.

4. Tecnología Calma

Diseños que tengan "*calma*" e informen son dos necesidades humanas que no van de la mano. La tecnología de la información es usualmente la enemiga de la "*calma*". Teléfonos celulares, servicios de noticias, la WWW, los emails, TV y radio nos bombardean todos los días.

Pero algunas tecnologías si nos llevan a la "*calma*" y confort.

"No existe tecnología dentro de un buen par de zapatos, una buena lapicera o en el diario del domingo como en una computadora personal"

¿Por qué uno nos irrita y otros nos calman? Mark Weiser cree que la diferencia está en cómo nos llama la atención.

La tecnología calma se basa en el "centro" y la "periferia" de nuestra atención y se va moviendo de uno a otro.

4.1 Periferia¹³

Utiliza la palabra periferia para describir a los objetos que están alrededor que sutilmente están dentro de nuestro foco de percepción pero que no están en el centro de nuestra atención.

Por ejemplo, si estamos manejando un auto, estamos centrados en el camino, la radio, etc. pero no somos conscientes del ruido del motor hasta que algo funciona mal y automáticamente pasa a ser nuestro centro de atención.

13 - Brown, J.S. and Duguid, P. Keeping It Simple: Investigating Resources in the Periphery Solving the Software Puzzle. Ed. T. Winograd, Stanford University.

Una tecnología calma tendría que pasar los objetos de la periferia al centro y viceversa, y eso sería "calmo" por dos razones.

Primero, ubicando a los objetos en la periferia nos permitiría tener más cosas en consideración que si estuvieran todos en el centro de nuestra atención.

Entonces, la periferia nos estaría informando sin sobrecargarnos de información, solo cuando lo necesitamos.

Segundo, pasar a los objetos de la periferia al centro nos permite atender de ellos cuando sea necesario.

La noción de la periferia está relacionada con la noción de "affordances", dicho concepto se utiliza en psicología perceptual y se define como la cualidad de un objeto que "sugiere" cuando y para que deberíamos usarlo¹⁴.

4.2 Características de la Tecnología Calma

La tecnología es calma mientras permita estar en la periferia. Esto pasa de dos maneras.

Primero, como se mencionó anteriormente, es la que permite pasar de centro a periferia y viceversa.

Segundo, que pueda ampliar nuestro "alcance periférico" mostrando más detalles sobre el mismo.

Un ejemplo común es una videoconferencia que, comparado con una conferencia telefónica, nos permite apreciar la postura corporal y expresiones faciales que de otra manera sería inaccesible. Esto "da calma" cuando la periferia aumenta nuestro conocimiento y por lo tanto nuestra habilidad de actuar sin sobrecargar la información.

14 Gibson, J. The Ecological Approach to Visual Perception. New York: Houghton Mifflin, 1979.

El resultado de la tecnología calma es hacernos sentir cómodos. Cuando nuestra periferia está funcionando bien, sabemos lo que pasa alrededor y qué puede ocurrir.

Según Mark Weiser:

“La "calma" es un nuevo reto que la Computación Ubicua trae a la informática”

5. Ejemplo de un sistema de Computación Ubicua¹⁵

AULA: Un Sistema Ubicuo de Enseñanza-Aprendizaje Colaborativo

El sistema AULA (A Ubiquitous Learning Appliance) se trata de un sistema que utiliza el paradigma ubicuo para informática educativa. Fue realizado por Maximiliano Paredes Velasco con la colaboración del grupo CHICO (Computer Human Interaction and Collaboration) de la Universidad de Castilla la Mancha para su tesis de doctorado.

A continuación presentaremos el prototipo de dicho sistema que nos servirá para entender mejor a que se refieren con la Computación Ubicua.

5.1 Un modelo ubicuo para la enseñanza de idiomas

Se trata de aprendizaje de idiomas, en particular ingles.

Aprender y practicar la escritura del inglés en un entorno colaborativo entre un grupo de estudiantes y al mismo tiempo, todo el proceso está guiado por la computadora, que a su vez participará en su desarrollo mediante el paradigma de Computación Ubicua.

En la enseñanza de inglés, por lo general, se le da al alumno un tema sobre el que escribir y algunas indicaciones, a lo sumo, sobre el tipo de composición a realizar, ya sea una descripción, una explicación, una narración, etc.

¹⁵ IE Comunicaciones Revista Iberoamericana de Informática Educativa Número 4, Julio-Diciembre 2006, pag 13-16 , Tesis Doctoral de Maximiliano Paredes Velasco .

Por lo general la escritura de composiciones se realiza en casa como práctica individual, con acceso a todo tipo de ayuda posible, o con más frecuencia en exámenes.

En dicha clase se presenta esta tarea con las instrucciones habituales, pero los alumnos deben realizarla en grupo y pueden, en todo momento, obtener ayuda de la computadora y ser guiados por el profesor.

Respecto de la composición escrita, generalmente el estudiante escribe primero un esquema con los principales aspectos sobre los que va a tratar el cual se realiza principalmente por medio de anotaciones y luego va escribiendo algunas ideas sobre el texto a escribir.

Una vez que esas ideas han sido organizadas y seleccionadas, los alumnos comienzan a escribirlas en frases correctas y párrafos coherentes. Esta es la parte sintáctica, la de la formación de las frases gramaticalmente correctas y la de la organización textual de éstas y de los párrafos.

De este modo, en un entorno colaborativo y ubicuo, se le da a los estudiantes el tema y el tipo de texto que han de escribir. Los alumnos comienzan la tarea y escriben las ideas, sugiriendo aspectos e ideas al grupo por medio de sus dispositivos móviles, que sirven para enviar y recibir todo el conjunto de sugerencias del grupo, que a su vez se irá representando en la pizarra o pantalla electrónica.

Una vez acabado el texto se propone para revisión y aprobación del grupo y se presenta al profesor, quien la corrige y la incluye en la biblioteca de casos.

5.2 Realización del modelo.

Para cumplir dicho objetivo hay que definir un modelo de aula que se apoye en los principios de la Computación Ubicua y del aprendizaje colaborativo asistido por computador.

5.2.1 Arquitectura para soportar computación ubicua y colaboración en el aula

La arquitectura del modelo se apoya en la utilización de diversas tecnologías de redes inalámbricas como se muestra la siguiente imagen:

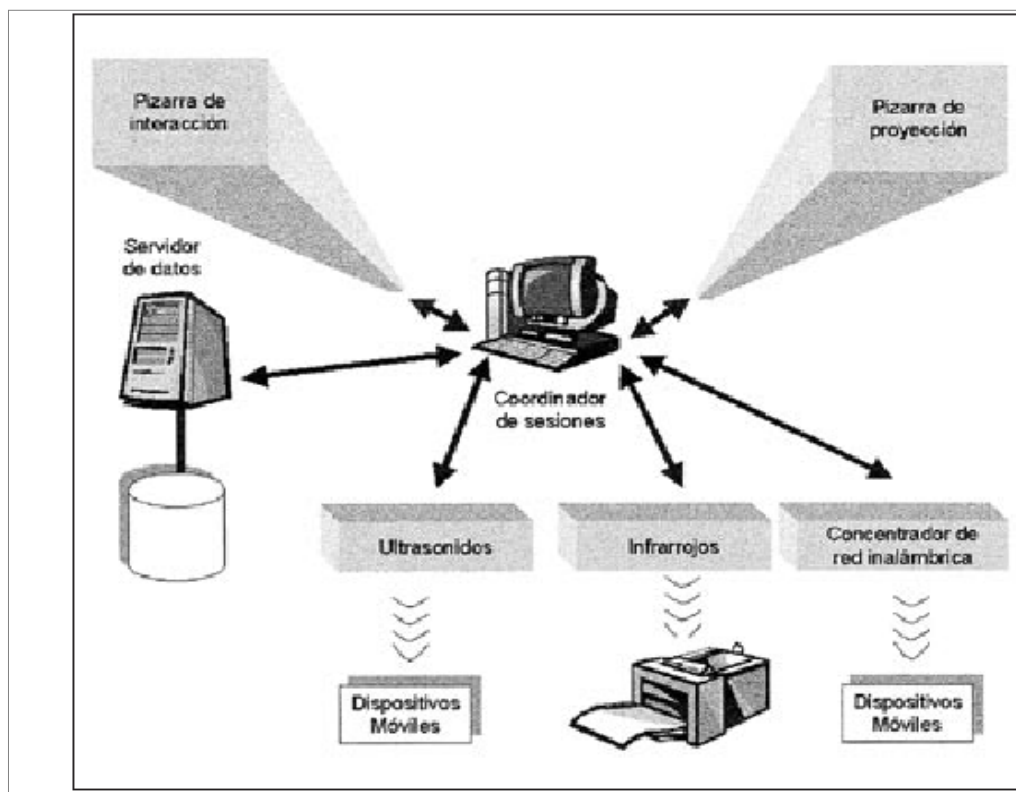


Figura 2 Arquitectura del sistema AULA

A continuación se describen los distintos componentes utilizados:

Red inalámbrica

Como red inalámbrica utilizaron ELSA AirLancer, que sustituye el cableado estándar de red de área local (Local Area Network, LAN) por enlaces de radio, dando lugar a redes de área local inalámbricas (Wireless Local Area Network, WLAN) que incorporan la misma funcionalidad que una LAN, pero además facilitan la integración de dispositivos móviles.

Las tarjetas de red utilizan el estándar IEEE 802.11b.

Emisión y recepción de infrarrojos

La transferencia de información por infrarrojos se realiza haciendo uso de dispositivos que incorporan el estándar IrDA (Infrared Data Association), que se diseñó especialmente para comunicación entre dispositivos de tipo palmtops, notebooks portátiles y terminales de telefonía móvil.

Dispositivos de ultrasonidos

En este modelo la tecnología de ultrasonidos se emplea para determinar la posición de los dispositivos móviles que incorporan un dispositivo emisor para estar localizados, del mismo modo que Weiser empleaba las marcas activas.

Pizarra de proyección

Como pizarra de proyección utilizaron un retroproyector conectado a una tarjeta VGA de la computadora que utilizaron como coordinador. De este modo, cuando se considera oportuno se puede proyectar información sobre una pantalla blanca.

Pizarra para edición

Además de la pizarra de proyección utilizaron un dispositivo capaz de captar lo que se dibuja o escribe con rotulador (o tiza electrónica) sobre una pizarra. Una vez captada, esta información puede ser procesada y enviada a todos los dispositivos móviles a través de la red inalámbrica, compartiendo esta información en tiempo real.

Para tal efecto utilizaron Mimio¹⁶, de Virtual Ink™, que constituye un hardware portable y fácil de utilizar para almacenar, reproducir e imprimir la información dibujada sobre una pizarra blanca. La pizarra de proyección se puede superponer sobre la pizarra de edición para así dar la impresión de disponer de una única pizarra.

16 <http://www.mimio.dymo.com/es-LA.aspx>

5.2.2 Coordinador de sesiones.

El coordinador de sesiones es una computadora que dispone de interfaces de comunicación para todas las tecnologías descritas anteriormente y es el encargado de recibir, procesar, almacenar y distribuir la información generada.

También utilizaron otra computadora encargada de la gestión de los datos, con un motor de bases de datos, en este caso MySQL.

5.2.3 Dispositivos móviles.

Los dispositivos móviles, como muestra la figura 3, están formados por un PDA (Cassiopea de Casio o iPAQ de Compaq) que utilizando un módulo de ampliación para tarjetas PCMCIA o MMCard pueden incorporar el hardware necesario para acceder a la red inalámbrica y de este modo intercambiar información.

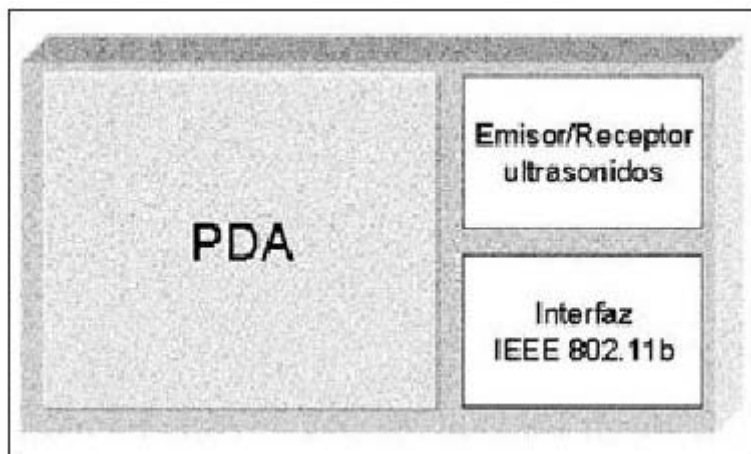


Figura 3 PDA Utilizado para el sistema AULA

Además se le añade un dispositivo emisor/receptor de ultrasonidos para que de este modo se pueda localizar la situación del dispositivo en el entorno así el coordinador de sesiones puede determinar cuando una persona con un dispositivo móvil se acerca a la pizarra y directamente proyectar la información contenida en su PDA sobre la pantalla de proyección, reproducir los mensajes que almacena, etc.

5.2.4 El software desarrollado para los dispositivos móviles

Utilizaron los lenguaje de programación Java y Visual C++.

La aplicación que se ejecuta en los dispositivos móviles se desarrollo Visual C++ para Windows CE, empleando como recursos de comunicación los sockets proporcionados por clase CCESocket y como mecanismo para acceso bases de datos utiliza las clases CCeDBRecord y CCeDBDatabase, en ambos casos proporcionadas por las MFC (Microsoft Foundation Classes) de C++.

El objetivo del alumno es elaborar colaborativamente un documento final en lengua inglesa mediante un procedimiento constructivista. Para cumplir este objetivo el sistema guía en todo momento al alumno y le proporciona herramientas de colaboración síncronas y asíncronas, junto con herramientas lingüísticas.

El sistema se basa en una arquitectura cliente/servidor, por este motivo la aplicación necesita un paso inicial en el que se conecta al servidor para iniciar la sesión. En este inicio de sesión se identifica el alumno mediante el nombre del usuario y su clave. Una vez que el servidor confirma la validación correcta del usuario, se muestra el espacio de trabajo a partir del cual el alumno podrá acceder a las herramientas colaborativas y lingüísticas.

Este espacio está dividido en tres áreas:

- 1- Área de Edición Individual
- 2- Área de Edición del Documento Final
- 3- Y el área de Visualización Colaborativa (Ver figura 2).

Las aplicaciones software que forman el sistema están estructuradas en cuatro grandes módulos: Gestor de Localización, Coordinador de Sesiones, varios módulos PDA (representan los dispositivos móviles) y un Servidor de Chat como se muestra en la siguiente figura.

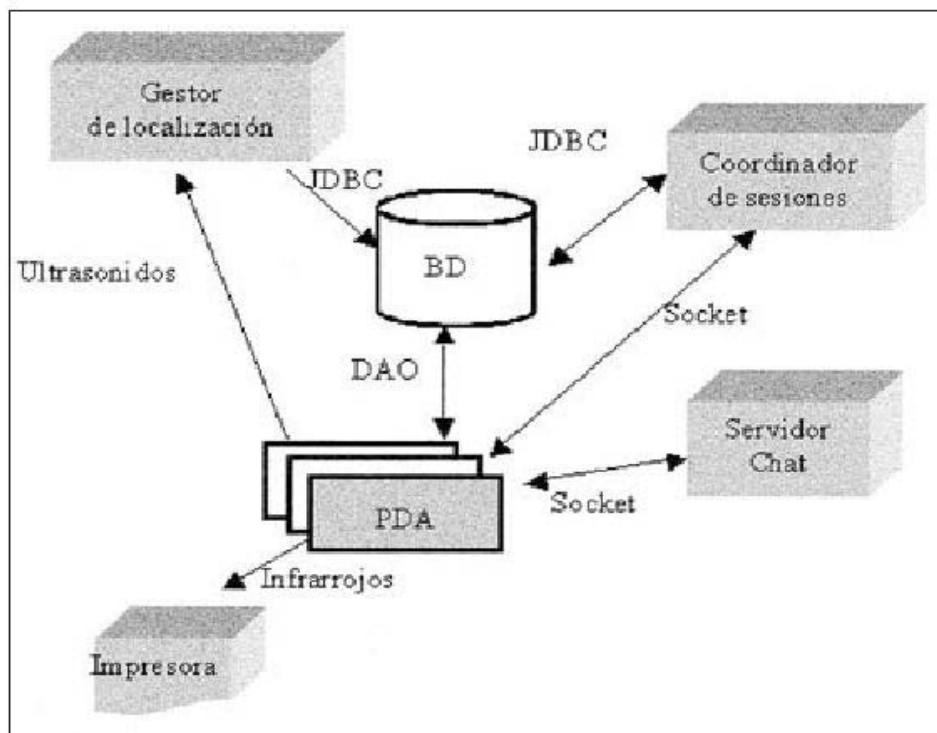


Figura 4 Arquitectura de software del sistema AULA

Los tres primeros módulos comparten una base de datos.

La comunicación entre todos los dispositivos se basa en tres canales diferentes: red inalámbrica, ultrasonidos e infrarrojos (La red inalámbrica el medio que soporta mayor flujo de información)

A partir del tema central (tópico) propuesto por el profesor en clase, el alumno elaborará su propuesta particular en el área de edición individual, por lo tanto, cuando el alumno inicia su primera sesión, en esta zona aparecerá únicamente el texto del tópico elegido.

Luego el alumno escribe en esta área las características que a su criterio son más apropiadas y que deben formar parte del documento final. Estas aportaciones que realiza el alumno se organizan en lo que se denominaron aspectos (Aspect) e ideas, de tal forma que un aspecto se refina en varias ideas.

El texto de los aspectos e ideas es completamente de edición libre, no estableciendo

el sistema pautas ni guías al alumno. La única restricción que impone el sistema es que no se pueden crear más de 5 aspectos para un tópico ni más de 5 ideas por aspecto, ya que un número mayor complica bastante la elaboración del documento final y no aporta nada constructivo al mismo.

Los aspectos e ideas se organizan mediante una estructura de árbol, facilitando así la comprensión y organización al alumno por tratarse de una estructura natural y familiar para él. En este árbol, el nodo raíz es el texto del tópico y sus nodos hijos son el texto de los aspectos. Los nodos hijos de los aspectos son las ideas y constituyen las hojas del árbol. Haciendo clic sobre el símbolo + o - de cada nodo se expande o contrae dicho nodo, permitiendo al alumno tener una visión más global o detallada de la propuesta que está elaborando.

La aplicación suministra un menú de contexto y un menú de sistema para la creación y organización de los aspectos e ideas con las siguientes acciones:

Enviar propuesta: El texto de los aspectos e ideas se muestra en el área de Visualización Colaborativa para su discusión entre todos los miembros del grupo.

Modificar: Permite modificar el texto de un aspecto o idea. Evidentemente el texto del tópico no puede ser modificarlo por el alumno.

Eliminar todo: Borra el texto de los aspectos e ideas, quedando únicamente el texto del tópico.

Eliminar selección: Borra el aspecto o idea seleccionado en ese momento.

Nuevo Aspect: Permite añadir un nuevo aspecto.

Nueva IDEA: Permite añadir una idea nueva.

En todo este proceso el sistema ofrece unas herramientas al alumno para reforzar su contacto con el grupo. El alumno dispone de un canal chat y un sistema de mensajería electrónica, ambos para dialogar con los miembros del grupo y el profesor. El canal de chat se utiliza para realizar consultas en línea y para dar soporte al proceso de discusión y argumentación sobre las propuestas en la fase de brainstorming.

La herramienta de mensajería electrónica estructura toda la información en aspectos e ideas. Se utiliza una estructura de árbol cuyo nodo raíz es el texto del tópico y sus descendientes son los aspectos, siendo las ideas los nodos hoja.

Cada vez que un alumno propone al grupo un nuevo aspecto o idea, la aplicación actualiza esta estructura creando un nuevo nodo según corresponda, por lo que esta estructura sigue la misma línea que el área Visualización Colaborativa.

Todos los mensajes están relacionados con un aspecto o idea en concreto mediante el texto del asunto (se hace corresponder el mismo).

Además, cuando el usuario responde a un correo recibido se establece un enlace con el primero, de tal forma que un alumno puede seguir la pista y navegar por todo lo que se ha comentado acerca de una idea o aspecto.

5.3 Conclusión sobre el sistema

El artículo propuesto, remarca que el objetivo de dicho sistema (AULA) da un marco para investigar y tratar de extraer las implicaciones de la computación ubicua en sistemas computacionales para el aprendizaje en grupo en particular y el trabajo en grupo en general.

También destaca que ha sido posible adaptar la realización de ejercicios en clase en tareas de carácter colaborativo con la aplicación del paradigma de Computación Ubicua sin que a primera vista sea necesario llevar a cabo cambios extraordinarios en la dinámica de la clase.

El alumno sigue tomando notas y escribiendo en lo que parece un pequeño cuaderno, dirige con frecuencia su atención a la pizarra, donde el profesor presenta cierta información.

No hay necesidad de cambiar de posición en el aula y se mantiene la participación individual. Pero en toda esta actividad subyace la aportación de una serie de dispositivos electrónicos que posibilita esta técnica de trabajo colaborativo basado en la computación ubicua.

6. Computación ubicua, dispositivos y tendencias de investigación.

La computación ubicua nos presenta una nueva manera de ver a las computadoras y la informática, esta visión busca un nuevo uso cualitativo del computador, desde una perspectiva de la relación de los dispositivos y las personas. Como dice Mark Weiser, *"Lo que importa no es la tecnología en sí, sino su relación con nosotros"*.

Las computadoras fueron evolucionando, por ejemplo convirtiéndose en *"computadoras como herramientas"* para el tratamiento de la información y ayudar a las personas en sus tareas diarias, hasta *"computadoras como medio"*, con la utilización de Internet y la World Wide Web.

Al mismo tiempo que las computadoras fueron cambiando, también lo hicieron y evolucionaron las personas que utilizan la informática.

La noción de simbiosis Persona-Computador dio comienzo a una época en el uso del computador en la que las personas llegaron a ser consideradas parte de los sistemas interactivos, si bien una parte periférica, y dio lugar al estudio de sistemas y el factor humano, esto se llamo el estudio de la interacción Persona-Computador (Human Computer Interaccion, HCI).

La Computación Ubicua desafía ahora esta perspectiva y propone una nueva época en el uso del computador donde las personas estén rodeadas de un gran número de dispositivos informáticos, utilizándolos de forma implícita en la realización de actividades cotidianas.

Agregar a las personas en el estudio de la informática ha supuesto ampliar los límites en los que se estudiaba inicialmente la informática, ahora, la Computación Ubicua está llevando incluso a romper otra barrera más, la del sistema persona-computador como se estudia en HCI, y a considerar el uso de la tecnología informática en el contexto más amplio de la vida cotidiana.

En dicho contexto, las actividades humanas se basan en situaciones y se expresan mediante la interacción con entornos de diseños de la vida moderna. El estudio de la informática en la vida cotidiana requerirá nuevos adelantos en la ciencia de la computación y en la interacción persona-computador.

En el artículo "The Computer of the 21st Century"¹⁷ Mark Weiser sugirió que *"la computación ubicua fundamentalmente se caracteriza por la conexión de objetos en el mundo real con la computación"*.

Hans-Werner Gellersen¹⁸ afirma que para que la computación sea ubicua en el sentido de la visión de Weiser, es fundamental preguntarse cómo se relacionan los dispositivos computacionales con otros dispositivos, también cómo la interacción de las personas con los dispositivos computacionales se relaciona con la interacción de éstas con otros dispositivos.

Aunque esto no ha quedado claro en las primeras investigaciones de computación ubicua, estas preguntas se han convertido en un tema recurrente para el diseño de un mundo de "computación en todos lados".

Estas corrientes de investigación se pueden dividir en las siguientes¹⁹:

1. La investigación de la interfaz entre dispositivos tradicionales y computación, principalmente desde una perspectiva HCI. El tema general es unir la interacción de personas con dispositivos computacionales en una interacción con los dispositivos más tradicionales. Los investigadores de HCI exploran específicamente interfaces tangibles, el medio ambiente y la realidad aumentada, todo lo que consideramos bajo la denominación de interacción situada (Situated Interaction).
2. Otra línea de investigación parte de los dispositivos no computacionales en entornos diseñados por el hombre y considera su implementación aumentada con la información para crear dispositivos aumentados por computador con una presencia digital. Esta línea de investigación está influida por el diseño, y se interesa por la computación y la interactividad como materiales (Augmented Reality).
3. La tercera corriente de investigación toma los dispositivos computacionales como punto de partida e investiga cómo pueden ser mejorados/aumentados

17 <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>

18 Trabaja de editor del Journal on Personal and Ubiquitous Computing y coordina el proyecto Smart-Its dentro del programa europeo de investigación The Disappearing Computer.

19 Computación Ubicua: el punto de encuentro entre computación y dispositivos, Upgrade Novatica, Revista de la asociación de técnicos de Informática, Septiembre – Octubre 2001 Nro 153, pag. 13.

con la capacidad de percibir y utilizar aspectos del entorno circundante como contexto. Surge esta línea de investigación principalmente de un entorno informático y puede ser considerado bajo la denominación de ***Informática Sensible al Contexto***.

Capítulo II

Sensibilidad al Contexto

1. Que es el Contexto.

El termino contexto fue introducido por primera vez por Schilit y Theimer en un artículo de la revista IEEE Network en 1994 “Disseminating Active Map Information to Mobile Hosts”²⁰. En dicho documento se referían al contexto “*como la ubicación de una entidad en relación a otros*”.

Schilit²¹ lo divide en tres categorías:

- Computing context: Redes, costo en la comunicación, ancho de banda, recursos cerca del dispositivo (Impresoras, Displays y Ordenadores), etc.
- User context: Perfil de usuario, ubicación, personas cercanas al dispositivo y hasta la situación social actual.
- Physical context: Esto sería la luz, el ruido, el tráfico y la temperatura.

Y lo define como:

“Contexto es conocimiento sobre el estado del usuario y dispositivo, incluso alrededores, situación y ubicación”.

Para Guanling Chen y David Kotz²² el tiempo también es importante y proponen agregar una cuarta categoría.

- Time context: día, mes, semana, estaciones del año.

Y definen:

²⁰ <http://impact.asu.edu/~cse591uc/papers/00313011.pdf>

²¹ Context-Aware Computing Applications. Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, paginas 85-90, Diciembre 1994.

²² A Survey of Context-Aware Mobile Computing Research, Guanling Chen and David Kotz, pagina 2

“Contexto es un conjunto de estados y características que determinan tanto el comportamiento de una aplicación como así también algún evento de la aplicación que ocurra y sea interesante o relevante para el usuario.”

Por su parte, Jason Pascoe²³ define al contexto como:

“La habilidad de un programa o dispositivo de obtener información de varios estados de su entorno y de sí mismo.

Los aspectos importantes del contexto son: donde estas, con quien estas y cuáles son los recursos más cercanos.”

Para Anin K. Dey, este tipo de aproximaciones que utilizan ejemplos son difíciles de aplicar para dar una definición más exacta y generalizada, ya que, en ese caso si hay algún tipo de información que no esté dentro de los ejemplos previos, no se puede saber si esta dentro de la definición de contexto.

Dichas definiciones son muy específicas y no se puede enumerar que aspectos de toda la situación son relevantes para el usuario, ya que van cambiando de entorno. Por lo tanto para definir al contexto hay que tomar toda la situación relevante a una aplicación y a su usuario.

Anin K. Dey propone definir al contexto como²⁴:

“Cualquier información que pueda ser utilizada para caracterizar la situación de una entidad. Una entidad puede ser una persona, lugar u objeto que es considerado relevante para la interacción entre un usuario y su aplicación, incluyendo el usuario y la aplicación misma.”

23 Adding Generic Contextual Capabilities to Wearable Computers, pagina 1.

24 Understanding and Using Context, Future Computing Environments Group, Anin K. Dey, pagina 3

1.1 Definición de Contexto.

Habiendo enunciado definiciones de contexto de varios autores, creo que la definición de Anin Dey se aproxima a una definición más general y que se puede utilizar como marco conceptual y directriz para futuras investigaciones, por lo tanto definimos a contexto de la siguiente manera:

“Contexto es cualquier información, o sea, conjunto de datos, estados, eventos y características que sean relevantes; que determinan el comportamiento de una aplicación para que se pueda utilizar para caracterizar la situación de una entidad.

Una entidad puede ser una persona, lugar, objeto o dispositivo que sea considerado relevante para la interacción entre un usuario y su aplicación, incluyendo el usuario y la aplicación misma”

2. Informática Sensible al Contexto.

La informática sensible al contexto se refiere a un modelo de programación en el cual las aplicaciones obtienen y utilizan información contextual para aprovecharla y así enriquecer dicha aplicación.

Guanling Chen y David Kotz²⁵ afirman que uno de los retos de la informática móvil distribuida es aprovechar los entornos cambiantes por medio de aplicaciones que utilicen información contextual y por lo tanto estén “conscientes del contexto” en el cual están corriendo.

Este software sensible al contexto se tendría que adaptar en función de su ubicación, las personas cercanas, hosts y dispositivos accesibles, entre otros. Un sistema con estas capacidades puede examinar el entorno y actuar en función de dichos cambios.

Como mencionamos en la definición de contexto, no solo donde estas, con quien estas y cuáles son los recursos cercanos sería información importante, sino que cada aplicación determinaría la relevancia de los datos que debería obtener, procesar y compartir.

25 A Survey of Context-Aware Mobile Computing Research, Guanling Chen and David Kotz

Se estuvieron investigando este tipo de aplicaciones usando lo que llamaron PARCTAB²⁶ en la década del 90.

El sistema PARCTAB integraba a un dispositivo Móvil, el cual llamaron Tab, en una oficina con acceso a una red local. Dicho proyecto también sirvió para experimentar la computación ubicua.

La Tab actuaba como una terminal grafica y la mayoría de las aplicaciones corrían en hosts remotos.

El diseño aprovechaba procesamiento remoto para utilizar dispositivos baratos y pequeños. Para el ingreso de datos la Tab tenía tres botones, y una pantalla Touch de 128x64 pixeles y un speaker²⁷ como se muestra en la siguiente figura.



Figura 5 PALM utilizada en el proyecto PARCTAB

En esa oficina interconectaron a todos los dispositivos con este sistema que realizaba una identificación periódica de las tabs y un monitoreo preciso de las ubicaciones de los dispositivos, inclusive cuando los dispositivos no estaban en uso.

26 A Survey of Context-Aware Mobile Computing Research, Guanling Chen and David Kotz

27 <http://www.ubiq.com/parctab/csl9501/node3.html#SECTION00030000000000000000>

El sistema notificaba la ubicación de las aplicaciones y sus cambios, también daba la información de la ubicación a un servidor público que recolectaba y redistribuía información sobre los objetos y ubicaciones. En la siguiente figura se ilustra la disposición de las herramientas.

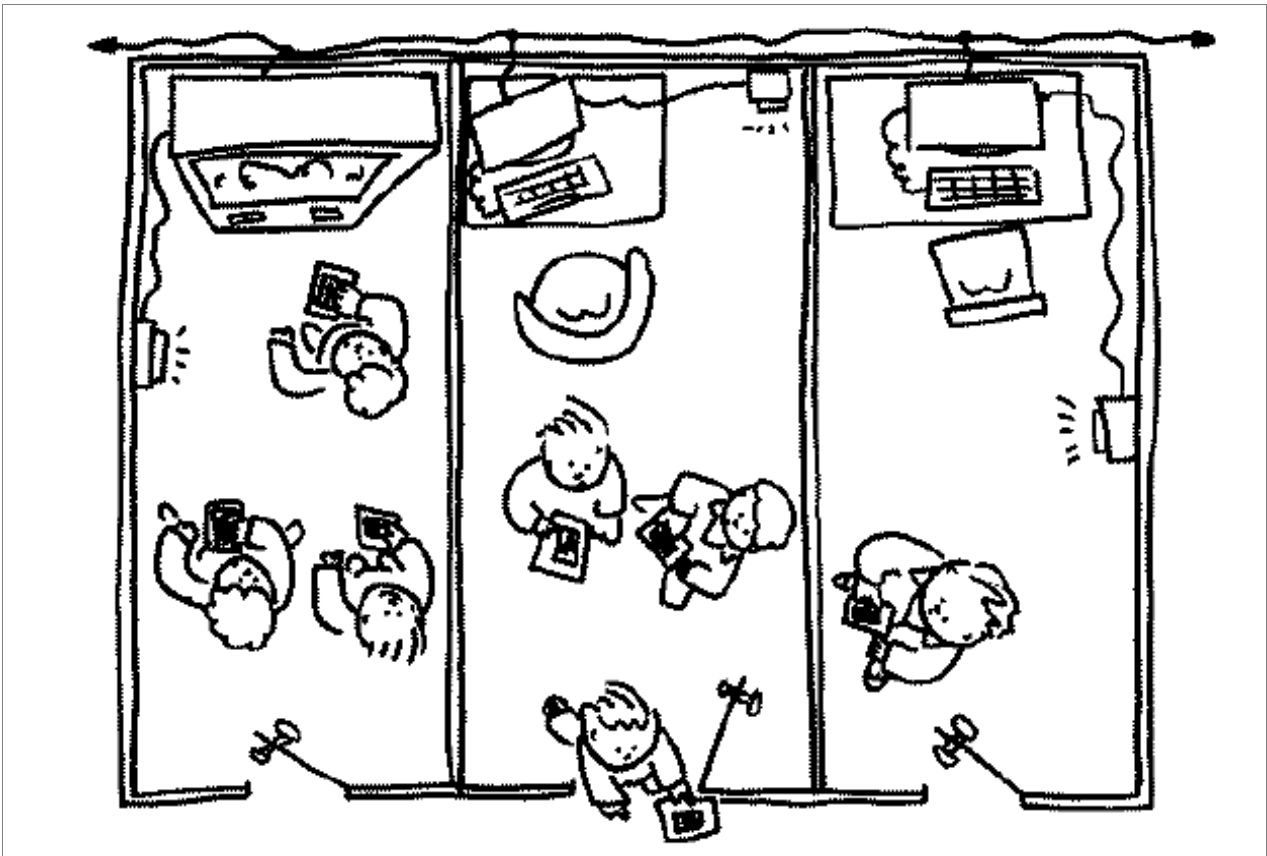


Figura 6 Sistema PARCTAB

Utilizando como base lo descrito más arriba, en el artículo “Context-Aware Computing Applications”²⁸ se propone una manera de construir aplicaciones sensibles al contexto, dividiéndolo en cuatro categorías, teniendo en cuenta si se trata de información o comandos y si se obtiene manual o automáticamente, como muestra el siguiente cuadro:

²⁸ <http://www.ubiq.com/want/papers/parctab-wmc-dec94.pdf>

	Manual	Automática
Información	“proximate selection” & información contextual	Reconfiguración contextual automática.
Comandos	Comandos contextuales	Acciones de contexto disparadoras “Context-triggered actions”

2.1 Selecciones próximas (Proximate Selection)²⁹

Es una técnica en la cual se utiliza una interfaz grafica de usuario con los objetos cercanos a un dispositivo para que sean más fáciles de seleccionar, ya sea tanto para su utilización como para su visualización.

En general esta técnica utiliza dos variables, “ubicación” y “selección”.

Se detectan por lo menos tres tipos de objetos a localizar que serian interesantes para utilizar esta técnica.

El primero seria dispositivos de entrada y salida que necesiten ser reubicados para su utilización, por ejemplo, impresoras, displays, altavoces, cámaras, faxes, etc..

El segundo, objetos en los que ya se está interactuando y que necesitan actualizar su dirección por algún proceso de software, como ser, personas en la misma habitación con la que se quiera compartir un documento.

El tercer tipo, son un conjunto de lugares que sean de interés, por ejemplo, restaurantes, pubs, boliches, estaciones de servicio, etc..

²⁹ Creo que una traducción alternativa y mas acertada seria “Selecciones Cercanas”

Como ejemplo podemos citar una oficina con impresoras en las que se utilizan cuadros para mostrar la impresora que mejor satisface a su usuario para su posterior selección y utilización como se muestra en la siguiente tabla:

Nombre	Oficina	Distancia
printer-dirección	001-00	6 metros
impresora-sistemas	003-04	10 metros
desarrollo-printer	002-011	5 metros

Se pueden visualizar de diferentes maneras, según su uso, por ejemplo, enfatizar la impresora más cerca o mostrar la lista ordenada por distancia o alfabéticamente. También marcar las impresoras que sean blanco y negro y las que sean de color, etc.

2.2 Reconfiguración Contextual automática

Es el proceso de agregar o quitar componentes o alterar las conexiones entre los mismos. Los componentes y conexiones típicas son los servidores y sus clientes, sin embargo, también podrían ser drivers, módulos de software, elementos de hardware, etc..

En los sistemas sensibles al contexto la parte interesante es como los distintos contextos pueden traer diferentes configuraciones de sistema y como se adaptaría la aplicación al mismo.

Cuando un grupo de personas están en un lugar, pueden compartir fácilmente los objetos físicos de ese lugar, por ejemplo, en una sala de reuniones las personas comparten una mesa con papeles y pizarras con diagramas.

Para simular este tipo de acciones, en PARCTAB se realizó un sistema que provee un espacio de trabajo para cada sala, algo parecido a una pizarra virtual (Muy parecido al trabajo presentado en el Capítulo I, Punto 5)³⁰.

Entrando en esa sala el dispositivo se enlaza automáticamente a la pizarra virtual, en ese sentido las personas pueden colaborar y compartir información mediante dicha pizarra. Cuando se mueven a otra sala el dispositivo se enlaza nuevamente y le trae la pizarra virtual de dicha sala.

La reconfiguración automática crea la ilusión de compartir los objetos virtuales como si fueran físicos.

La reconfiguración puede basarse no solo en la ubicación sino por ejemplo en las personas presentes en la sala. Si un equipo se reúne entonces la pizarra virtual se activa, y si no se encuentran dispositivos en la sala, se desactiva. Eso hace a las pizarras virtuales más útiles ya que la información en la pizarra puede persistir de reunión en reunión y puede seguir a los participantes de una sala a otra.

También puede incluir funciones del sistema operativo, Schilit y Duchamp en “Adaptive remote paging for mobile computers” describen como un sistema operativo puede utilizar la memoria de los dispositivos inactivos cercanos para expandir su memoria en vez de utilizar el disco local como swap.

En esta técnica se encontraron algunos problemas, por ejemplo, si el contexto cambia muy rápido puede ser poco práctico, debido a la utilización de recursos para tal fin que cause que baje el rendimiento del dispositivo en esa adaptación continua. También, ciertas adaptaciones pueden confundir al usuario, si el contexto fue cargado erróneamente, si no se sabe que contexto puede ser relevante para la aplicación, o si el contexto cambia mientras se está utilizando una aplicación.

³⁰ Nota: Vemos acá como están íntimamente relacionados la Computación Ubicua con la Informática Sensible al Contexto, siendo la primera una visión mas global.

2.3 Comandos Contextuales

Las acciones de las personas muchas veces pueden predecirse en función a su situación y entorno, por ejemplo, hay ciertas acciones que una persona casi siempre realiza en una cocina, librería u oficina.

Los comandos contextuales apuntan a aprovechar este hecho. Consultas de información contextual pueden producir diferentes resultados dependiendo en el contexto en donde se realizan.

De una manera similar se podrían parametrizar “comandos contextuales”, por ejemplo, el comando print (imprimir) podría, por defecto, imprimir en la impresora más cercana.

En PARCTAB se utilizó una aplicación que ve un sistema de archivos basado en su ubicación (location-based filesystem), lo llamaron “Location Browser”.

En este sistema los directorios llevan el nombre de su ubicación y contienen archivos, links, carpetas, programas, etc.. y cuando un usuario va cambiando de habitación, también cambia su vista de carpetas con los de la ubicación correspondiente.

Por ejemplo, cuando un usuario está en su oficina ve su calendario, planos, etc.. cuando se va a la sala de trabajo ve apuntes de la reunión, cuando se va a la cocina ve un instructivo de cómo hacer café. Un empleado puede dejar una nota electrónica para sus colegas avisando que tuvo que realizar un trámite.

Este sistema (Location Browser), además de mostrar datos parametrizados por la ubicación del usuario, también ejecuta comandos. Este tipo de comandos contextuales se pueden presentar de dos maneras.

Primero, la apariencia del comando puede cambiar dependiendo del contexto en uso, por ejemplo, si un usuario está en la biblioteca el botón para invocar el catálogo está más accesible, cuando normalmente está oculto.

Segundo, el comando puede parecer lo mismo pero produce resultados parametrizados, por ejemplo, dicho sistema presenta un botón de “migrar” que es igual para todas las habitaciones pero la migración se realiza en el host remoto que está dentro de la habitación del contexto en uso.

2.4 Acciones contextuales disparadoras “Context-triggered actions”

Las acciones “context-triggered” son reglas IF-THEN que especifican como los sistemas sensibles al contexto deberían adaptarse.

Cuando se obtiene información sobre el contexto en uso que sea relevante puede disparar un comando si se considera pertinente. Por lo tanto se configura que información se podría utilizar para disparar comandos.

En PACTAB experimentaron este tipo de aplicaciones con Active Badge³¹, Watchdog y recordatorios contextuales.

Watchdog monitorea la actividad de los Active Badge y ejecuta comandos de consola con la siguiente forma:

```
badge location event-type action.
```

Badge y location son cadenas de caracteres que corresponden al usuario y su ubicación. Even-type es el tipo de evento: llegando(arriving), saliendo(departing), quedandose(settled-in), perdido (missing) o atención (attention) (es un botón que cuando se hace clic dos veces el evento atención se re reporta).

Cuando un evento ocurre, watchdog invoca una acción con una serie de variables de entorno de Unix como parámetros. Esto incluye el propietario del Badge, la oficina, ubicación y el nombre del host más cercano, por ejemplo:

```
coffe kitchen arriving "play -v 50 ~/sounds/rooster.au"
```

```
schilit * attention "emacs -display $NEARESTHOST:0.0"
```

El primer ejemplo monitorea el badge “coffe” (que está en la cafetera de la cocina) y emite el sonido de un gallo cuando alguien hace café. El segundo abre el emacs³² en una ventana en el host más cercano cada vez que detecta una señal.

31 Es una etiqueta que periódicamente difunde un identificador único para determinar su ubicación

32 Editor de texto de Unix

Además de este tipo de acciones, “Recordatorios Contextuales” también permite una descripción más detallada de la situación. De acuerdo donde, cuando y quien se abre una ventana con algún mensaje. Por ejemplo, “la próxima vez que vaya a la biblioteca” o “cuando vuelva a mi escritorio” o “la próxima vez que van a Juan”, etc. y el usuario puede editar el recordatorio, ignorarlo o borrarlo.

Al desarrollar este tipo de aplicaciones se encontraron varios problemas, por ejemplo, cuando el usuario tiene que pasar por varios estados hasta llegar al que necesita, puede que no se quiera que se ejecuten todas las acciones intermedias. También si el sistema no ejecuta el recordatorio a tiempo, este se vuelve sin utilidad.

Hay dos temas a tener en cuenta, la expresividad del predicado del lenguaje y la precisión y oportunidad de la información contextual subyacente.

3. Obteniendo información del contexto.

Para utilizar información del contexto en aplicaciones, obviamente, tiene que haber una manera de obtener datos del entorno y de alguna manera enviárselo a la aplicación que lo necesita.

Primero describiremos algunos tipos de información contextual³³:

3.1 Tipos de información contextual

3.1.1 Ubicación

La ubicación del dispositivo, y en consecuencia la del usuario que lo utiliza es un dato muy utilizado en las aplicaciones, de hecho es el más utilizado y que obtuvo la mayor atención en los últimos tiempos.

Existen dos maneras de obtener dicha información, manual y automático, la primera dependen de la voluntad del usuario para dar a conocer su ubicación.

Respecto de la relatividad de la ubicación del usuario se pueden utilizar dos criterios, afuera y dentro de los establecimientos:

33 A Survey of Context-Aware Mobile Computing Research, Guanling Chen and David Kotz

1. Afuera de los establecimientos (Outdoors):

La manera más común de obtener información de la ubicación es por medio del GPS (Global Positioning System).

También existe otra manera descrita en el artículo "GPS-Less Low Cost Outdoor Localization for very small devices"³⁴ de Nirupama Bulusu, John Heidemann, and Deborah Estrin, que permite obtener dicha información para dispositivos de bajo costo, utilizando algoritmos que producen una exactitud de 3 metros a la redonda del punto seleccionado.

2. Dentro de los establecimientos (Indoors):

El GPS no funciona dentro de edificios debido a la baja señal y en algunos casos la necesidad de una alta precisión.

Para tal efecto, la mayoría de los investigadores construyeron su propio sistema, por ejemplo, Active Badge System de Olivetti, Personal Shopping Assistant de AT&T, entre otros.

En el artículo "The Anatomy of a Context-Aware Application"³⁵ muestran como su sistema que utiliza técnicas de ultrasonido se pueden utilizar para localizar objetos inclusive hasta en movimiento.

Uno de los retos en esta área es desarrollar un sistema estándar de localización dentro de edificios tal como lo es el GPS.

3. Sistemas que necesiten las dos aproximaciones:

En la actualidad no existe una manera de informar la ubicación tanto para dentro de los establecimientos como fuera con una exactitud considerable para los dos contextos.

En la actualidad las aplicaciones tienen que consultar diferentes sistemas de localización para ubicar diferentes tipos de objetos.

Otras alternativas es utilizar redes bayesianas u otros algoritmos de inteligencia artificial que son demasiado complejos y que no dependen de los sensores, sino del software que se utilice.

³⁴ <http://www.isi.edu/~johnh/PAPERS/Bulusu00a/>

³⁵ <http://dl.acm.org/citation.cfm?id=313476>

3.1.2 Tiempo

Esta información simplemente se puede obtener del reloj del dispositivo.

El tiempo es una información contextual interesante para aplicaciones de oficinas y pueden ser de gran utilidad, no solo obteniendo el tiempo, sino el día de la semana , la semana del mes, el mes del año y hasta inclusive la estación corriente. Se podría utilizar con calendarios u otras aplicaciones que ayuden a organizar el usuario en sus tareas diarias, entre otras cosas.

Un ejemplo interesante y que ya se menciona en el capítulo anterior es determinar un horario y una ubicación que notifique al usuario cuando entra a una oficina, si por ejemplo, ya paso el horario y entro a determinada habitación que comunique sobre alguna tarea que debería realizar.

3.1.3 Objetos o dispositivos cercanos

Si el sistema obtiene la ubicación de personas, dispositivos u objetos, se podría utilizar dicha información para avisar cuando estén cerca de nosotros. Como ejemplo podemos citar el sistema “Teleporter” y el “Context-Aware Pager” que presentan [BBC97, Bro98].

La complejidad con este tipo de información es que los usuarios tienen que dar permiso para que otro usuario obtenga su ubicación. Como ejemplo, podría ser una red cerrada o una red correspondiente a un equipo de trabajo.

3.1.4 Ancho de banda

El ancho de banda es una información contextual interesante ya que, por ejemplo, las aplicaciones se podrían adaptar a los cambios en el ancho de banda del dispositivo.

El sistema Odyssey provee una API donde las aplicaciones pueden ser notificadas cuando el ancho de banda cambia.

Notar que aquí se presenta una API para obtener información contextual, se podrían utilizar distintas APIs como se menciona en el artículo siguiente de Mashup, para utilizar Mashups de información contextual, sobre este tema hablaremos en el último capítulo.

3.1.5 Otro tipo de información contextual

Fotodiodos para detectar el nivel de la luz, micrófonos para detectar el nivel de luz, sensores para detectar la temperatura, presión, etc.

Esta información contextual puede ser de utilidad para diferentes aplicaciones según su necesidad.

Por ejemplo para ahorrar batería el brillo de la pantalla se podría ajustar automáticamente dependiendo del nivel de luz o que cambie el brillo o contraste dependiendo del nivel de luz que pueda reflejar el display para que se pueda ver mejor.

Como se puede ajustar el display en función del nivel de luz también se podría ajustar el sonido del celular cuando llaman según el nivel de ruido en el ambiente.

3.1.6 Obteniendo información de contexto de alto nivel

Se podría obtener también información del usuario, como la actividad que está realizando.

Sin embargo, esto es más complejo ya que deberían intercambiar información con distintas aplicaciones para obtener información relevante.

Algún ejemplo simple sería utilizar el calendario para saber que debería hacer un usuario en determinada fecha y horario, y quizás utilizando el servicio meteorológico para avisar que el día que puso que iba a ir a un lugar se podría suspender por lluvia.

Algo un poco más complejo sería utilizar información de aplicaciones como Twitter o Facebook para obtener información contextual de la actividad del usuario y por ejemplo crear una base de conocimiento sobre todas las cosas que al usuario le gusta o interesa para luego ofrecer dependiendo la ubicación u otra información contextual alternativas o sugerencias sobre sitios de interés. Por ejemplo, está pasando por un lugar de comidas que le gusta y el dispositivo móvil le avisa que está cerca.

Otro ejemplo es el que se menciona en "Advanced Interaction in Context"³⁶ que mediante un proyecto llamado TEA, mostraron como reconocer si el usuario tenía un teléfono o un PDA en una mesa o maletín, utilizando sensores de luz y acelerómetros en dos direcciones. Esto se realizó con una red neuronal artificial llamada Kohonen Self-Organizing'.

Uno de los retos para manejar información contextual es la utilización de algoritmos de inteligencia artificial para obtener mejor calidad de información.

³⁶ http://www.teco.edu/~albrecht/publication/huc99/advanced_interaction_context.pdf

Y como ya hemos visto en la definición de contexto, la información recabada sobre el contexto tiene que ser relevante, así que cada sensor tendrá su importancia para cada tipo de aplicación.

3.1.7 Obteniendo información de contextos cambiantes.

Muchas aplicaciones necesitan ser notificadas cuando cambia el contexto.

Se puede utilizar una aplicación que monitoree el contexto y envíe los cambios a un servicio del contexto que publique los cambios del contexto. El servicio del contexto es responsable de enviar los cambios del contexto a los clientes (aplicaciones) que están suscritos para ser notificados.

Capítulo III

Mashup

1. Definición

1.1 Mashup³⁷

Se le da el nombre de Mashup a las aplicaciones que utilizan diferentes fuentes de datos que mediante técnicas de combinación, composición y/o visualización generando así una nueva aplicación o presentación.

Esta nueva aplicación genera información a partir de datos externos creando una nueva manera de utilizar dichos datos, dándole más utilidad.

2. Clasificación de los Mashups.

Se pueden clasificar de la siguiente manera (Teniendo en cuenta que algunos Mashups se pueden encontrar en más de un grupo):

2.1. Mashup Empresariales

Generalmente son aplicaciones que combinan los recursos propios de la empresa y los presentan en un nuevo portal Web.

Se enfocan en un único punto para presentar la información y que permita la colaboración de todos los usuarios.

Por ejemplo, el Lotus Mashup³⁸ de IBM es una herramienta gráfica basada en navegador que permite que los usuarios de la empresa con conocimientos Web monten las nuevas aplicaciones. También tiene un catálogo de Mashups que permite compartir y descubrir nuevos componentes de Mashup.

37 La palabra Mashup es utilizada en el ambiente musical cuando se crea una canción con la mezcla de otras canciones o partes musicales como fuente

38 <http://www.ibm.com/developerworks/lotus/products/mashups/>

2.2. Mashup de Consumidores

Este tipo es el más utilizado, combina diferentes tipos de datos desde múltiples fuentes públicas, pensado para el público en general.

Por ejemplo, Wikipediavision combina la API³⁹ de Google Maps y Wikipedia.

2.3. Mashup de Datos

Combina datos similares desde distintas fuentes y los presenta en un solo portal.

La combinación de todos estos recursos crea un nuevo y único Servicio Web diferente al resto de las otras fuentes.

2.4. Mashup de Mapas

Se refiere a las aplicaciones que utilizan las API que contienen datos geográficos.

Por ejemplo, Google libero la API Google Maps, permitiendo a los desarrolladores Web incluir en sus páginas las relaciones de los datos con su respectivo mapa.

También se encuentran las API Visual Earth de Microsoft, Yahoo Maps y MapQuest de AOL.

2.5. Mashup de Videos y Fotos

El sitio Flirck⁴⁰ libero su API el cual permitió compartir fotos a través de distintas páginas Web.

También permite realizar otros tipos de Mashup más interesantes, ya que el contenido de la foto contiene metadata asociada a dicha imagen, los desarrolladores pueden asociar dichas fotos con otro tipo de contenido, por ejemplo si la imagen es la tapa de un CD, buscar más información de dicho CD.

39 Application Programming Interface

40 www.flirck.com , es un sitio que permite compartir fotos.

2.6. Mashup para Búsquedas y Compras

Este tipo de sitios, es decir, herramientas que comparan productos, precios, etc.

Por ejemplo, BizRate, PriceGrabber, MySimon y Froogle, ya existían. Pero esto no se realizaba mediante alguna API de proveedores sino con técnicas mucho más complicadas como el Screen-Scraping⁴¹.

Ahora hay varias APIs que facilitan la creación de sitios de este tipo, como las de eBay y Amazon que permiten acceder a sus productos y la información asociados a ellos.

2.7. Mashup de Noticias

Las fuentes de noticias han utilizado tecnologías de intercambio de información como RSS⁴² y ATOM.

Estas tecnologías permiten agregar información en las páginas Web y hasta crear un diario personalizado para cada usuario. Por ejemplo Diggdot.us, el cual combina fuentes de los sitios digg.com, slashdot.org y del.icio.us.

Aunque estos dos últimos no son novedad, la novedad es que técnicas como las del Mashup o de las tecnologías relacionadas, facilitan el intercambio de información de manera que se puedan generar aplicaciones mucho más ricas en contenido sin recurrir a técnicas complicadas que requiera de conocimientos avanzados de programación, por esa razón, el Mashup está llegando cada vez a más personas y está generando una nueva manera de generar contenido en la Web.

41 Técnica de programación que consiste en tomar una presentación de una información y extraer los datos.

42 Really Simple Syndication

3. Tecnologías relacionadas

Un Mashup se puede dividir en tres capas:

- **Presentación:** Esto es la interfaz de usuario. Las tecnologías utilizadas son HTML/XHTML, CSS, JavaScript, AJAX, etc.
- **Servicio Web:** Se utilizan APIs para acceder a los datos o fuentes de datos. Las tecnologías utilizadas son JSON-RCP, SOAP, REST, XMLHttpRequest.
- **Datos:** Esta capa se refiere a la manipulación de los datos, es decir, enviar, recibir o almacenar dichos datos y se realiza con lenguajes como XML, JSON o KML.

A continuación explicaremos brevemente las distintas tecnologías.

3.1 Presentación

HTML

HyperText Markup Language («lenguaje de marcado de hipertexto»), es el lenguaje predominante para la elaboración de páginas Web. Se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>). Dicho lenguaje puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir lenguajes como JavaScript, el cual puede afectar el comportamiento de los navegadores Web.

XHTML

eXtensible HyperText Markup Language. XHTML es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros. En su versión 1.0, XHTML es solamente la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones de XML.

CSS

Cascading Style Sheets (hojas de estilo en cascada) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador Web permitiendo mejoras en la interfaz de usuario y páginas Web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS)., la más utilizada es la del lado del cliente.

AJAX

Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), no es un lenguaje sino una técnica de desarrollo Web para crear aplicaciones interactivas o RIA⁴³. Las aplicaciones que utilizan esta técnica se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

3.2 Servicios Web

Un Servicio Web es un programa que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar estos servicios para intercambiar datos.

43 Rich Internet Applications

La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C⁴⁴ son los comités responsables de la arquitectura y reglamentación de los Servicios Web.

SOAP

Simple Object Access Protocol es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado en 1998, llamado XML-RPC.

SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos más utilizados en los Servicios Web.

REST

La Transferencia de Estado Representacional (Representational State Transfer) o REST es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la Web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo.

3.3 Datos

XML

El eXtensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el W3C. Es una simplificación y adaptación del SGML⁴⁵ y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML).

JSON

Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

44 World Wide Web Consortium

45 Standard Generalized Markup Language

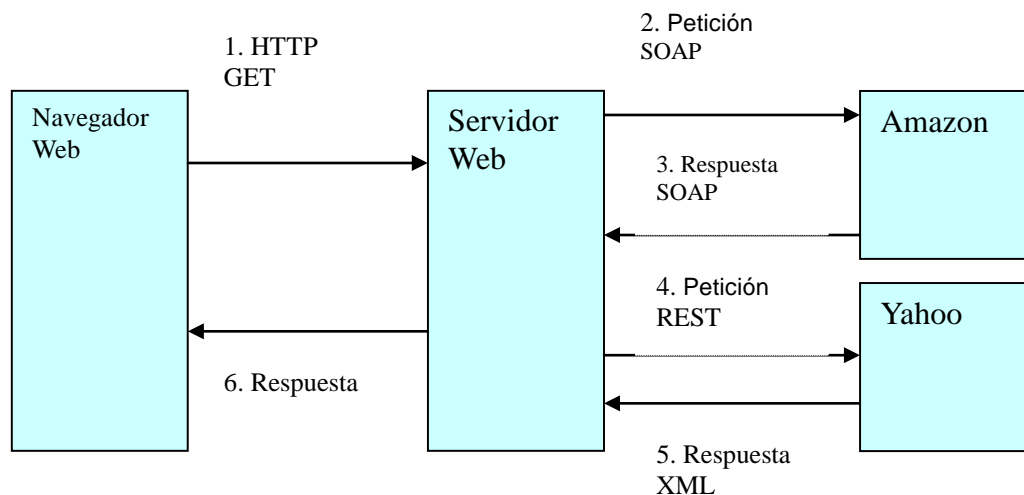
4. Arquitectura de los Mashups

Existen dos tipos de arquitecturas:

4.1 Basados en el Servidor

Utilizan el servidor remoto para analizar y re estructurar la información para luego transmitirla al navegador del usuario final en su forma final. Esta arquitectura es muy útil cuando se trata con navegadores de dispositivos móviles.

A continuación se muestra una típica solicitud de pagina HTTP para este tipo de Mashup.



1. El navegador Web se comunica con el servidor, le hace la petición de una página usando HTTP o HTTPS.
2. La pagina es construida por el servidor Web el cual llega a la fuente o sitio asociado (Amazon, Yahoo, Google, etc.). La primera petición que se muestra en el ejemplo es para Amazon, utilizando el protocolo SOAP sobre HTTP.
3. Amazon devuelve una respuesta SOAP
4. La segunda petición es para Yahoo, usando REST (Transferencia de estado representacional)
5. Yahoo responde con XML, sin formato, sobre HTTP.

6. El servidor Web agrega las respuesta, combinando y racionalizando la información de cualquier forma que tenga sentido
7. La información resultando es enlazada al HTML e insertada en la respuesta, la cual es enviada al navegador Web.

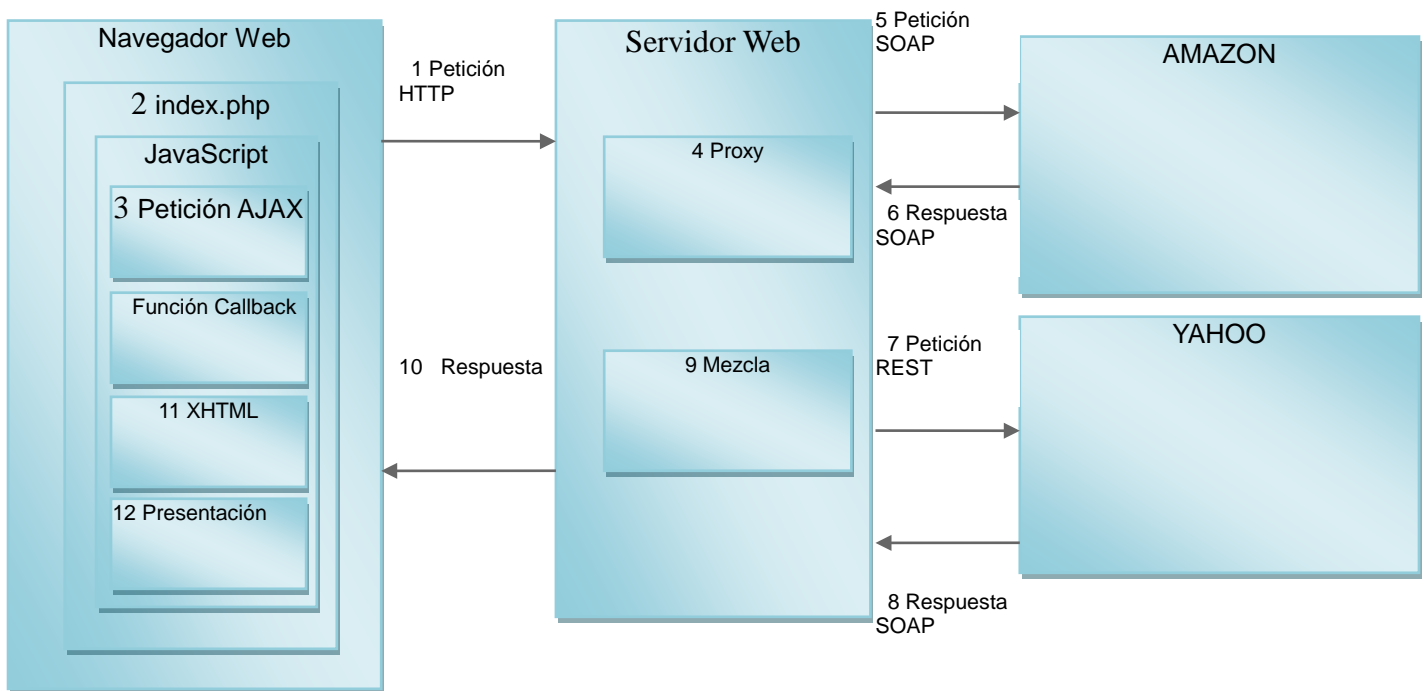
Uno de los principales beneficios que presenta esta arquitectura es la separación completa del navegador Web con los sitios que suministran la información, así el programador no se debe preocupar de aspectos como si el navegador Web soporta ciertas tecnologías.

Entre las desventajas se encuentran que el navegador Web hace la petición al servidor por una página completa la cual no es una forma eficiente ni sofisticada, además, como el servidor Web es el que hace todo el trabajo de la manipulación de la información, convierte a este enfoque a no escalable, ya que mientras más peticiones reciba el Mashup se incrementara el trabajo del servidor, mientras que el navegador del lado del cliente está relativamente desocupado.

4.2 Basados en navegador

Utilizan el navegador Web del usuario para combinar y re estructurar la información. Este enfoque permite tener una mejor experiencia de usuario, pero implica un mayor grado de complejidad al momento de manejar las peticiones Asíncronas. Petición

A continuación se muestra el flujo de información de dicha implementación:



1. El navegador Web se comunica con el servidor, le hace la petición de una página utilizando HTTP o HTTPS.
2. En este punto se produce la carga de la página por primera vez no existe contenido Mashup. Se descarga JavaScript al navegador junto con el HTML de la página.
3. El navegador emite una solicitud al servidor por contenido adicional, esta solicitud puede ser SOAP, REST o XML-RPC, pero todas con el mismo principio básico. Lo realmente importante en este punto es que la solicitud se realiza a través de JavaScript y se realiza en forma asíncrona, por lo tanto la página no se actualiza ni

se bloquea el navegador mientras procesa la llamada. Todo esto puede ser de forma transparente al usuario.

4. El servidor en este caso actúa como Proxy, el navegador ha realizado una petición a una fuente, en este caso Yahoo, el servidor no hace más que transmitir la petición a dicha fuente.
5. Una petición SOAP es hecha a Amazon
6. Amazon responde con una respuesta SOAP
7. Una petición REST es realizada a Yahoo
8. Yahoo procesa la petición y regresa la información al servidor con XML, sin formato, sobre HTTP.
9. En el servidor puede darse combinación de información. Este paso es opcional ya que la información puede solo ser enviada al navegador.
10. Una vez que la información esta lista es enviada al navegador mientras tanto en el navegador una función Callback en JavaScript que fue nombrada en la petición original maneja la respuesta cuando esta es enviada desde el servidor.
11. Una transformación es aplicada a la información XML para convertirla en XHTML, la misma que incluye información y marcas de presentación
12. El fragmento es insertado en la estructura de la página y se presenta al usuario.

Un aspecto positivo de esta técnica es que permite obtener una aplicación con una interfaz de cliente enriquecida y con los beneficios de una aplicación que no necesita ser instalada. Y como el navegador es el que realiza la mayor parte del trabajo, el servidor tiene recursos para poder servir más páginas, haciendo más escalable la aplicación resultante.

5. Problemas en la integración de datos⁴⁶

Hay diferentes problemas a tener en cuenta cuando se trata de la integración de datos; Se puede clasificar en cuatro grupos, diferencias entre el texto y el dato, identificación de objetos, diferencia en el esquema, diferencias en el nivel de abstracción, precisión de los datos.

5.1 Diferencia entre el texto y dato

Una gran parte de datos se encuentra en texto. El lenguaje humano es ambiguo. La ambigüedad hace que la asociación con los datos sea compleja. Asimismo, los datos expresados en lenguaje humano es difícil de procesar por software. Una de las funcionalidades de los sistemas de integración de datos es la de superar estas dificultades, la asociación entre documentos y datos.

5.2 Identificación de Objetos

Datos en forma estructurada se encuentra en varios formatos. Dejar los datos en un formato común es el primer paso. Inclusive si los datos están disponibles en un formato común, en la práctica las fuentes difieren en como... la diferencia existe en los dos niveles, en el de objetos individuales y en el nivel del esquema. Como ejemplo en el nivel de objeto: la SEC usa Central Index Key (CIK) para identificar personas, compañías, e instrumentos financieros mientras otras fuentes como DBpedia, utiliza URL para identificar entidades.

5.3 Niveles de Abstracción

Las fuentes de datos proporcionan datos a nivel de abstracción incompatibles o clasifican los datos de acuerdo a las taxonomías pertinentes a cierto sector. Dado que los datos se publican en diferentes niveles de abstracción (persona, empresa, país, sector) la integración de datos requiere de saber el nivel de abstracción el cual se esta tratando. Los datos para puntos de vista individuales la mayoría de las veces no concuerdan con datos para empresas u otro tipo de organización que caracterice el tipo de datos según su conveniencia. También hay diferencias en agregación geográficas, por ejemplo datos de

46 <http://sw.deri.org/2009/09/financiam-data/>

una fuente de datos de una región o país a otro. Otro problema es cuando se trata de las monedas de cada país, por ejemplo, pesos, dólar, euros, los cuales primero tienen que saber que tipo de moneda es para poder convertirlo o indicar el tipo de moneda cuando se integra el dato, aunque cada usuario le serviría solamente ver la moneda de su país.

5.4 Calidad de Datos

La calidad del dato es un tema complejo cuando se integra automáticamente desde diferentes fuentes de datos. En un entorno abierto los procesos que agregan datos no tienen ningún control sobre los procesos que lo suministran por lo tanto no puede controlar la calidad del dato suministrado. Un problema grave es cuando se quieren combinar datos incorrectos con unos ya existentes, lo cual en la mayoría de los casos los vuelve inutilizables, especialmente cuando se realiza algún tipo de inferencia en dicha combinación.

Por lo tanto, uno de los problemas a tratar es ver como los procesos de origen y destino pueden coordinarse para asegurar la calidad del dato.

Se necesitan métodos y técnicas para chequear la integridad y precisión de los datos.

6. Del escritorio a dispositivos móviles

Hoy en día se escucha mucho la palabra Web 2.0 para referirse a la “nueva Internet”, esto se refiere a un progreso importante en aplicaciones Web en los últimos 20 años.

El surgimiento de nuevas tecnologías como mencionamos anteriormente, AJAX, REST, Web Services, etc. trajo como consecuencia aplicaciones más dinámicas y amigables y en el que los usuarios dejaron de ser pasivos o receptores de información de Internet a ser más activos y participativos llenando de contenido la Web, es decir, ahora la información o transferencia de contenido es ida y vuelta.⁴⁷

Las dos características más importantes de las aplicaciones de la Web 2.0 son:

1. Lectura/Escritura Web: El usuario ya no es más un sujeto pasivo, un consumidor de datos publicados por otra persona, sino que contribuye continuamente con información nueva, como en Wikis, blogs, Facebook, Twitter, etc.
2. Web como plataforma: La Web ya no es mas solo como un conjunto de links interrelacionados sino también como un conjunto de aplicaciones Web que ofrecen APIs abiertas, las cuales puede componerse para crear aplicaciones más sofisticadas combinado datos desde diversas fuentes, como ya mencionamos en este apartado, los Mashups.

Una tecnología que es clave para la Web 2.0 como mencionamos anteriormente es AJAX.

Con AJAX, se envían fragmentos de páginas, en vez de recargar la pantalla entera (en una manera asíncrona sin interrumpir el control del usuario). Consecuentemente se reduce una cantidad enorme de datos que se envían entre cliente y servidor, llevando a un desarrollo de aplicaciones Web mas interactivas, ideales para el dominio móvil.

47 A Context-Aware Mobile Mashup for Ubiquitous Web, Dlego Lopez Ipina, Inaki Vazquez y Joseba Abaitua, pag. 3

Desde el punto de vista de la investigación, el nuevo paradigma Web 2.0 (desde el escritorio a dispositivos móviles) por ejemplo, yendo a Mobility 2.0, parece un reto interesante, que recién está comenzando. Vamos a ver en el siguiente punto un resumen de lo que ya se está empezando a utilizar y ciertos proyectos en que se está investigando Mashup a nivel de dispositivos móviles.

7. Mashup en dispositivos móviles

Estado del Arte

Realizar un Mashup para dispositivos móviles, es decir, realizar aplicaciones Web utilizando las API existentes como mencionamos anteriormente y que puedan accederse por teléfonos inteligentes, tabletas, etc. en su navegador, no es muy diferente al desarrollo convencional. Solo cambia la manera en que presentan el contenido ya que tienen que utilizar un diseño que se adapte a resoluciones más pequeñas, aproximadamente entre 128 x 160 a 320 x 320 pixeles⁴⁸ y que soporte varias dimensiones de pantallas, por la variedad de dispositivos móviles y tamaños.

Esto se logra utilizando los estándares para desarrollo Web Móvil, como ser, el lenguaje XHTML o XHTML-MP, Wireless CSS y también JQuery Mobile.

No vamos a profundizar en este tipo de desarrollo ya que el tema principal en este punto es utilizar Mashup puramente en dispositivos móviles.

Como vimos en el punto 3, la comunidad de Internet proporciona APIs para acceder a servicios y plataformas para desarrollar aplicaciones, hoy en día también los operadores de telefónicas han empezado a abrir sus plataformas y sus redes para, por ejemplo, enviar o recibir SMS, realizar llamadas a terceros o permitir localizar a sus clientes. Las empresas Ribbit Mobile (<http://web21c.bt.com/>) y OrangePartner (<http://www.orangepartner.com/>) disponen de APIs para comenzar a desarrollar e investigar en el tema.

48 Mobile Web Developer's Guide, pagina 22.

Estos son ejemplos de cómo los operadores tratan de construir una comunidad de desarrolladores alrededor de sus redes, proporcionándoles acceso controlado a los servicios de su red.

Este enfoque está siendo también adoptado por Google con Android (<http://code.google.com/android/>) y su Open Handset Alliance (<http://www.openhandsetalliance.com/>).

A pesar de que estas iniciativas suponen una evolución hacia el desarrollo de Mashups, ninguna de ellas está realmente dirigida a la creación de servicios por parte de los propios usuarios finales como ocurre en la WWW.

Lo que se está realizando, por ahora, es llevar hacia el dominio móvil varios portales en los que, si bien los usuarios no pueden contribuir con nuevos servicios, sí que son los responsables de actualizar y dotar de contenido a los mismos como Flickr, YouTube o Google Maps.

Muchos de ellos se han ido integrando en motores de Widgets como Webwag mobile (<http://www.webwag.com/mobile>), Nokia Widsets (<http://www.widsets.com/>), mobile Opera framework (<http://www.opera.com/products/mobile/>) o Yahoo! Go (<http://mobile.yahoo.com/go>).

Nokia Widsets fue la primera herramienta gráfica para crear Widgets para móviles. Sin embargo, estos Widgets están fundamentalmente enfocados a la agregación de feeds RSS y además sólo se permiten unos pocos pasos de configuración en el proceso de creación de nuevos servicios, lo que limita en gran medida la creatividad del usuario.

Yahoo! Pipes (<http://pipes.yahoo.com>) fue la primera plataforma ligera (basada en AJAX) que permitió a la comunidad de usuarios crear y compartir sus propios servicios de información.

A pesar de su flexibilidad y relativa sencillez de uso, es bastante limitada en cuanto al tipo de servicios que se pueden crear (básicamente suscripciones RSS) al no estar diseñada para integrar servicios de operadores. Además, la interacción con el usuario se realiza únicamente a través de su portal Web.

Proyectos como Microsoft Popfly (<http://www.popfly.com/>) o Google Mashup Editor (<http://code.google.com/gme/>), son una respuesta a la iniciativa de Yahoo!.

Popfly ha puesto mucho énfasis en el interfaz de creación de servicios, que se realiza a través de una página Web que puede ser personalizada gráficamente.

Sin embargo, al igual que Yahoo! Pipes, la creación de servicios no permite la integración de servicios de operadores de telecomunicaciones (voz, videoconferencia, SMS, etc.).

Telefónica también está apostando por este nuevo paradigma con iniciativas como Open MovilForum (<http://open.movilforum.com/>). En breve el sitio para acceder será <https://bluevia.com/> y en el cual se encontrarán APIs para utilizar con funcionalidades tales como enviar y recibir SMS, MMS, realizar publicidad y también una API dedicada al manejo del contexto de usuario (User Context).

Por el momento es bastante nuevo la creación de Mashup para dispositivos móviles y estamos lejos comparado con los Mashups tradicionales, pero como vimos ya hay varias empresas interesadas en el tema y ya están empezando a desarrollar sus plataformas para la creación de nuevas aplicaciones para los dispositivos móviles. Esto también es un reto para los próximos años.

Capítulo IV

Uniando Criterios

1. Mashups sensibles al Contexto

Como mencionamos anteriormente, estamos en la era de la Web 2.0 donde cada vez hay más servicios online que ofrecen datos y cada vez hay más mashups ofreciendo todo tipo de información.

Por otro lado, los dispositivos móviles cada vez son más potentes, tienen acceso a internet casi en todos lados y tienen más sensores, como acelerómetros, sensores de proximidad, GPS, etc. por lo tanto las aplicaciones para dispositivos móviles tienen más recursos para poder adaptarse al contexto y ofrecer así una mejor experiencia al usuario.

Con dispositivos móviles más potentes se espera que pronto ejecuten las mismas aplicaciones que las de escritorio y además con todos los sensores que disponen y la cantidad de Servicios Web que se encuentran en la Web se realizaran mas aplicaciones que se adapten al contexto.

Ya se están adaptando varios lenguajes de programación para que funcionen correctamente en los dispositivos móviles, adecuándose a los tamanos de pantalla y los navegadores que utilizan, por ejemplo, bibliotecas JavaScript como JQuery con JQuery Mobil, ExtJS con Sencha Touch o HTML5 que tiene mas recursos para estandarizar el desarrollo Web tanto de escritorio como móvil.

También se encuentran lenguajes como Python que realizo una adaptacion para la serie S60 del Sistema Operativo Symbian, llamado PyS60. O el lenguaje Java con J2ME y el SDK para Android cuando se quiere desarrollar software que no sea Web.

Respecto de los Mashups se encuentran sitios que permiten al usuario comun realizar sus propios mashups, por ejemplo, mash-o-matic, openkpow, yahoo pipes, microsoft popfly, ibm qedwiki, intell mashmaker, etc. Todas estas plataformas son eficientes para integrar varias fuentes de datos en una nueva presentación pero ninguna tiene la capacidad de utilizar sensores, por lo que todavía se requiere saber desarrollar en algún lenguaje para realizar mashups sensibles al contexto.

También hay que destacar que se podrían utilizar mashups de dos maneras, una cuando se accede por medio de un explorador a la Web que es donde se puso toda la atención estos últimos años o por medio de comunicaciones directas entre las redes móviles, una área más relegada.

En este capítulo vamos a ver algunas propuestas para crear mashups sensibles al contexto y como esta tecnología puede ayudar a tener una computación cada vez más ubicua.

2. Sensibilidad al Contexto y aplicaciones en la Web

Acceder a páginas o aplicaciones Web desde un dispositivo móvil abre las puertas a la creación de aplicaciones con nuevas funcionalidades (que no es posible su utilización cuando se accede desde una computadora de escritorio) como la obtención de datos del contexto.

Este tipo de aplicaciones se podrán adaptar a las necesidades puntuales de dicho usuario.

En el artículo "Context Aware Mashups for Mobile devices"⁴⁹ proponen un escenario el cual utiliza como dato contextual a la ubicación del dispositivo obtenido por GPS, un servicio de mapas y un listado puestos de interés cercanos de diferentes proveedores de datos.

Y citan que este tipo de aplicaciones deberían cumplir las siguientes características:

⁴⁹ ftp://129.69.211.2/pub/library/ncstrl.ustuttgart_fi/INPROC-2008-50/INPROC-2008-50.pdf

- ✧ La adaptación debería basarse en sensores que vengan con el dispositivo móvil o conectado localmente.
- ✧ El mashup debería estar centrado en el usuario, o sea, el usuario debería beneficiarse de la utilización del mashup en vez de un proveedor de servicios o persona remota.
- ✧ Debería existir una versión del mashup que “no adaptativa” en caso de que la información contextual no esté disponible. Esto permite ver los mashups en dispositivos con sensores así como también en computadoras de escritorio y hace al mashup más “utilizable”.
- ✧ Los mashups deberían permitir la integración de los datos de múltiples fuentes de datos en distintos formatos e interfaces. El usuario debería poder agregar o eliminar fuentes de datos en runtime, de acuerdo a su interés actual.

Otro ejemplo de aplicaciones web sensibles al contexto que actualmente integran información contextual lo encontramos en el sitio <http://hostip.info> que saca la ubicación de donde accede el navegador en función de la ip y utiliza un servicio web de mapas para mostrar la ubicación actual del usuario que entro. Esto se podría utilizar para mostrar publicidades u otro tipo de información sensibles a la ubicación, obviamente accediendo desde un dispositivo móvil ya que desde una computadora personal siempre va a tener la misma ubicación.

Dentro de este sitio también se encuentra una sección donde muestra cómo utilizar su API. Un ejemplo sencillo es mostrar la bandera del país del usuario que ingreso:

```
<A HREF="http://www.hostip.info">
<IMG SRC="http://api.hostip.info/flag.php" BORDER="0" ALT="IP Address Lookup">
</A>
```

Otro ejemplo es el sitio <http://plazes.com> que permite compartir información contextual como la ubicación y actividad actual y agregar personas para estar conectados así cada uno puede saber quiénes están cerca y que actividad está realizando. Este sitio esta creado para dispositivos móviles y permite integrar servicios web como los de twitter, fire eagle , MySpace, Google Maps, etc.. En el año 2008 fue adquirido por la empresa Nokia.

2.1 Características de las aplicaciones web sensibles al contexto.

Según el artículo mencionado se requieren de las siguientes características para realizar aplicaciones web sensibles al contexto:

Notificaciones Asíncronas: El mashup necesita ser notificado cuando cambia el contexto del usuario.

Mutabilidad: Como los sensores pueden activarse, volverse inactivos o se pueden conectar sensores nuevos, la aplicación debería poder adaptarse a dichos cambios.

Búsqueda: El mashup debería poder buscar que tipos de datos contextuales están disponibles. Esto necesita metadata para describir el dato contextual.

Control: Para utilizar un sensor local, el mashup debería tener algún tipo de control sobre el mismo, por ej., activar o disparar alguna medición.

Estandarización: Debería existir interfaces estandarizadas cuando se utilizan APIs en sistemas Web lo cuales son multi-plataforma y multi-proveedores.

Privacidad: El usuario debería tener control de que datos quiere mostrar. Como los mashups son aplicaciones de terceros, el sistema que maneje la información contextual debería asegurar la privacidad antes de que esa información llegue a los mashups.

2.2 Aproximaciones para suministrar información contextual

A continuación se presentan algunos protocolos en donde se puede suministrar información contextual:

- ✧ Las cabeceras HTTP pueden proveer información sobre el navegador y podrían transmitir más datos contextuales. Sin embargo carecen de notificación asíncrona, búsqueda, control y estandarización⁵⁰.
- ✧ Composite Capabilities/Preference Profiles (CC/PP)⁵¹ y User Agent Profile (UAProf)⁵² son aproximaciones basadas en perfiles pero provee solo atributos estáticos contextuales.

50 Geographics extensions for http transactions - Daviel, A., Kaegi, F.A, Kohfal, M. - Internet draft The Internet Society (Septiembre 2007)

51 Composite Capabilities/Preference Profiles: Structure and vocabularies 1.0, Klyne, G, et al. - Recommendation, W3C (Enero 2004)

52 <http://en.wikipedia.org/wiki/UAProf>

- ⤴ Delivery Context: Client Interfaces (DCCI)⁵³ es un framework basado en el cliente en el cual se pueden utilizar los sensores del dispositivo y servicios remotos. Este representa al contexto como un conjunto de propiedades organizadas jerárquicamente, utilizando el Document Object Model (DOM). El resultado de las páginas web pueden accederse mediante JavaScript. Estas propiedades pueden ser estáticas por ejemplo el tamaño de la pantalla o dinámicas por ejemplo la posición del usuario. Entre las ventajas que tiene esta aproximación es que esta creada por la W3C y no es tan antigua como las anteriores, fue creado en Junio de 2010.

Dadas las características mencionadas en el punto, DCCI parece ser el más adecuado para realizar aplicaciones mashup sensibles al contexto, a continuación se detallan algunas características importantes:

- ⤴ Utiliza el DOM event model para proveer notificaciones asíncronas.
- ⤴ Registra los eventos como el cambio de algún valor o adición de alguna propiedad y es notificado mediante el listener.
- ⤴ También es posible agregar o eliminar propiedades dinámicamente, logrando la característica de mutabilidad.
- ⤴ También las propiedades del DCCI tiene metadata y el árbol de DOM puede recorrerse.
- ⤴ DCCI no soporta control sobre sensores locales. Sin embargo, para activar o disparar un sensor se puede utilizar el modelo de eventos (event model).
- ⤴ El valor de un sensor se necesita solamente cuando algún event listener se registra a la propiedad respectiva, dicha propiedad puede (des) activar o disparar el sensor respectivo.

3. Sensibilidad al Contexto en las Telecomunicaciones.

Un área todavía no muy explotada es la de mashups directamente para dispositivos móviles. En el sitio <https://bluevia.com> perteneciente a Telefónica de Personal con colaboración de Movistar se pueden encontrar varias APIs para desarrollo en telefonía móvil, una de ellas es la User Context API la cual tiene varias funcionalidades tales como obtener información del celular, el perfil del usuario o datos personales. También se encuentra la Location API que ofrece funcionalidades para obtener la ubicación del dispositivo móvil.

53 <http://www.w3.org/TR/DPF/>

Las funcionalidades se realizan mediante peticiones HTTP/HTTPS utilizando el protocolo RESTful y mediante lenguaje XML, JSON para transportar los datos.

Por ejemplo para obtener datos del usuario se realiza un GET al siguiente URL:

<https://api.bluevia.com/services/REST/Directory/{guid}/UserInfo/UserPersonalInfo>

Donde {guid} es un código alfanumérico (token) otorgado por la empresa para poder utilizar el servicio.

Luego se tienen que agregar algunos parámetros para especificar el tipo de respuesta (JSON o XML), tipo de autorización, tipo de encriptación, y el número de versión de la API.

A continuación se ve un ejemplo completo:

1. Peticion

GET

```
https://api.bluevia.com/services/REST/Directory/alias:4bf499a1ecaac050dfaddfef87f99e3a/UserInfo/UserPersonalInfo?version=v1 HTTP/1.1 Authorization: OAuth realm="https://api.bluevia.com", oauth_consumer_key="Gjqq11099SE", oauth_token="4bf499a1ecaac050dfaddfef87f99e3a", oauth_signature_method="HMAC-SHA1", oauth_signature="%2BoMI8LXTM3iYihUqdQfDXHNHPZs%3D", oauth_timestamp="1291898799", oauth_nonce="15339077371968105486410263075" Host: api.bluevia.com:443
```

2. Respuesta

La respuesta consiste en devolver un 200 OK HTTP conteniendo los campos correspondientes a los datos del usuario, si es correcto o un código 400 Bad Request si los datos enviados son incorrectos.

Ejemplo:

HTTP/1.1 200 OK

```
Content-Type: application/xml; charset=UTF-8?<?xml version="1.0" encoding="UTF-8"?><NS1:userPersonalInfo xmlns:NS1="http://www.telefonica.com/schemas/UNICA/REST/directory/v1/"><NS1:gender>male</NS1:gender></NS1:userPersonalInfo>
```

Otro ejemplo es para obtener la ubicación del sensor:

Se realiza un GET de esta manera:

```
https://api.bluevia.com/services/REST/Location/TerminalLocation?version=v1&locatedParty=alias:access_token&version:version&alt:DataType
```

Donde también se coloca el código (token) de acceso, se indica si el resultado es en JSON o XML en el parámetro DataType, la versión y en este caso un parámetro que indica el nivel de precisión aceptable.

Cabe destacar que en la misma pagina dice “Blue Via *Beta*”, indicando que todavia esta en etapa de prueba, eso demuestra el estado en que se encuentra este tipo de aplicaciones.

Uno de los retos para los proximos años es la explotacion de este tipo de mashups.

Otro ejemplo para mostrar la tendencia hacia los dispositivos móviles y la utilización de sensores, se puede ver en Google Maps API Web Services, la cual ofrece los siguientes servicios:

1. Direction API:

Calcula la dirección entre dos ubicaciones, utiliza una petición HTTP para indicar el origen, destino que pueden ser en texto, por ej. “Buenos Aires”, o coordenadas.

2. Distance Matrix API:

Provee la distancia y tiempo que puede tomar para realizar un determinado recorrido.

3. Elevation API:

4. Este servicio provee una interfaz para consultar la elevación de una determinada dirección. Esta puede estar por arriba o debajo del nivel del mar.

5. Geocoding API:

Geocoding es el proceso de convertir direcciones de texto, por ejemplo, “Av. Directorio 1600, Buenos Aires, Argentina” a coordenadas geográficas, por ejemplo, latitud 37.423021 and longitud -122.083739

También provee un servicio que realiza el proceso inverso, o sea, convertir las coordenadas a texto.

Este servicio es muy útil cuando el usuario realiza búsquedas de direcciones y no sabe exactamente el texto para ingresar y también para obtener las coordenadas exactas de un punto marcado en el mapa.

A continuación se muestra un ejemplo de conversión de una dirección con el servicio de Geocoding:

Se realiza una petición GET:

`http://maps.googleapis.com/maps/api/geocode/json?address=1600+Directorio,+Buenos+Aires&sensor=false'`

El cual devuelve como respuesta el código HTTP 200 OK con las coordenadas en formato JSON, a continuación se muestra el resultado completo:

```
{
  "results": [
    {
      "address_components": [
        {
          "long_name": "1600",
          "short_name": "1600",
          "types": [ "street_number" ]
        },
        {
          "long_name": "Av Directorio",
          "short_name": "Av Directorio",
          "types": [ "route" ]
        },
        {
          "long_name": "Parque Chacabuco",
          "short_name": "Parque Chacabuco",
          "types": [ "neighborhood", "political" ]
        },
        {
          "long_name": "Buenos Aires",
          "short_name": "Buenos Aires",
          "types": [ "locality", "political" ]
        },
        {
          "long_name": "Ciudad Autónoma de Buenos Aires",
          "short_name": "Ciudad Autónoma de Buenos Aires",
          "types": [ "administrative_area_level_1", "political" ]
        },
        {
          "long_name": "Argentina",
          "short_name": "AR",
          "types": [ "country", "political" ]
        }
      ],
      "formatted_address": "Av Directorio 1600, Buenos Aires, Ciudad Autónoma de Buenos Aires, Argentina",
      "geometry": {
        "bounds": {
          "northeast": {
            "lat": -34.63033170,
            "lng": -58.44936289999999
          },
          "southwest": {
            "lat": -34.63035010,
            "lng": -58.4493670
          }
        }
      },
      "location": {
        "lat": -34.63035010,
        "lng": -58.44936289999999
      },
      "location_type": "RANGE_INTERPOLATED",
      "viewport": {
        "northeast": {
```

```

        "lat" : -34.62899191970850,
        "lng" : -58.44801596970848
    },
    "southwest" : {
        "lat" : -34.63168988029150,
        "lng" : -58.45071393029149
    }
}
},
"partial_match" : true,
"types" : [ "street_address" ]
}
],
"status" : "OK"
}

```

Para poder comprobar este servicio puede sencillamente copiar esta dirección “<http://maps.googleapis.com/maps/api/geocode/json?address=1600+Directorio,+Buenos+Aires&sensor=false>” y pegarla en el navegador.

Hay destacar que cuando se realizan las peticiones al Servicio Web **ahora es obligatorio especificar un parámetro que indica que el dispositivo que realizo la petición tiene o no sensor**. Esto quiere decir que este servicio está hecho más que nada para dispositivos móviles donde el dato contextual relevante es la ubicación obtenida del sensor.

4. Integración de datos en aplicaciones Mashups sensibles al contexto.

La integración de datos en un tema a solucionar para este tipo de aplicación debido a la gran cantidad de tipos de datos y formatos en los que se proveen. Se encuentran los mismos problemas que en los Mashups tradicionales como se explico en el punto (Poner el punto donde se explica).

En el proyecto TELAR⁵⁴ propone una manera simple de integrar datos, utilizando el patron Adapter (Wrapper) para enmascarar los distintos datos de los diferentes proveedores convirtiéndolo en un único formato utilizando una interfaz RESTful. En este caso al utilizar un Wrapper para integrar los datos es difícil de realizar ya que hay que programarlo, por otro lado mientras mas Wrappers se programen, se podrían reutilizar hasta el punto de no tener la necesidad de crear más.

54 <http://www.edbt.org/Proceedings/2008-Nantes/papers/N1124F.html>

También se podría utilizar el patron Factory, que cumpliría casi la misma función que el anterior, creando adaptadores dependiendo del tipo de datos.

Estas soluciones aunque solucionan el problema de integración requieren de conocimientos en programación y un gran esfuerzo para desarrollar las clases que funcionen de intérprete para los distintos tipos de datos utilizados.

Uno de los retos para estos proximos años es la definicion de un estandar o marco de trabajo para la integracion de multiples tipos de datos y desde multiples fuentes.

Conclusión

En este trabajo se presentó el concepto de “computación en todos lados” o Computación Ubicua como lo denomino su creador, Mark Weiser, un investigador y visionario.

Este marco conceptual permitió la creación de distintos dispositivos acompañado de aplicaciones más interesantes que las actuales.

Notemos que importante es la investigación y la proyección a largo plazo como ocurrió y ocurre en PARC⁵⁵, donde no solo salió dicho concepto, sino que casi todo lo que utilizamos hoy en día. La Interfaz Grafica de Usuario junto con el manejo de ventanas mediante el mouse, impresoras laser, Ethernet, Smalltalk, por nombrar algunos ejemplos.

Luego vimos “Sensibilidad al Contexto”, una rama de la informática que está siendo utilizada hoy en día para el desarrollo de aplicaciones para dispositivos móviles, ya que los sensores que tienen dichos dispositivos nos permiten obtener mucha información del entorno.

Y por ultimo vimos Mashup, algo que todavía está empezando y tiene un camino largo por recorrer y quizás sea la próxima tendencia en programación.

Aunque la palabra Mashup se utiliza para denominar un tipo de aplicación, a mi me gustaría tenerlo presente también como una técnica que permita disponer de un marco de trabajo para aplicaciones que utilicen múltiples fuentes y múltiples tipos de datos ya sea para producir una aplicación o que sea intermediaria para otras aplicaciones. La cual se podría llamar “Mashup de bajo nivel”.

Como se menciona en la introducción de la presente obra, los dispositivos móviles van a ser más usados que cualquier otro tipo de computadora, junto con esto las aplicaciones móviles van a ser cada vez más y más sofisticadas, aprovechando el contexto e Internet para ofrecer funcionalidades nunca antes provistas por una aplicación.

La interconexión entre dispositivos va a ser necesaria para varios ámbitos y no solo mediante internet, también en una LAN o directamente entre ellos. Ahí es cuando necesitamos intercambiar información de una manera más ordenada y estandarizada.

En este sentido se encontraron varios problemas con la integración de datos como se mencionó en el Capítulo II, la calidad del dato intercambiado es crítico para que la aplicación cumpla su objetivo, sin información válida para ofrecer la aplicación carece de sentido.

Cabe destacar que se encontraron varios puntos en común entre la Computación Ubicua y la Sensibilidad al Contexto, en el capítulo I se muestra una manera de

⁵⁵ http://en.wikipedia.org/wiki/PARC_%28company%29

obtener ubicuidad en un aula para enseñar idiomas y en el capítulo II una oficina en donde los dispositivos obtienen información del contexto y permanecen interrelacionados.

También los Mashups tienen puntos en común como se vio con la API de Google Maps, donde obtienen información de la ubicación y es obligatorio indicar si el dispositivo al que se accede tiene o no sensor.

Esta es una era que está comenzando, estamos viviendo cambios en la tecnología y vamos a ver aún más en los próximos años.

Y por último, como mencioné en la presente obra, estos tres temas los elegí por la relación que existe entre ellos. Tomé a la Computación Ubicua como marco conceptual, a la sensibilidad al contexto como una parte de la informática que permitirá la interrelación entre dispositivos de una manera más rica siendo consciente del entorno y a los Mashups como una manera de intercambiar información.

Trabajos Futuros

Se detectaron varios problemas en la integración de datos, no solo para la creación de Mashups, sino para todo tipo de interrelación entre dispositivos.

También se notó que en la actualidad carecemos de un marco de trabajo para la creación de aplicaciones sensibles al contexto.

Se podría utilizar la manera en que se desarrollan los Mashups para interrelacionar dispositivos no solo en Internet sino en una habitación cerrada, mediante una LAN.

Para tal efecto, se podría crear un marco de trabajo utilizando “Mashups de bajo nivel” como capa integradora para la comunicación entre los dispositivos móviles y utilizar sensibilidad al contexto para enriquecer la interrelación entre los mismos, logrando así una ubicuidad del entorno.

Bibliografía

[1] <http://www.ubiq.com/ubicomp/>

[2] The Coming age of calm technology, Mark Weiser and John Seely Brown, Xerox Parc, 5-10-1996

[3] <http://www.ubiq.com/parctab/csl9501/paper.html>

[4] An Overview of the ParcTab Ubiquitous Computing Experiment

[5] <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>

[6] Upgrade Novatica, Revista de la asociacion de tecnicos de Informatica, Septiembre – Octubre 2001 Nro 153.

[5] <http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm>

[6] Robertson, G.G. MacKinlay, J.D. and Card, S.K. "Cone trees: Animated 3D visualizations of hierarchical information." In HCI 91, pages 189-194, 1991.

[7] Transition to the Information Highway Era in 1995-96 Information Industry and Technology Update.p. 2.

[8] Brown, J.S. and Duguid, P. Keeping It Simple: Investigating Resources in the Periphery Solving the Software Puzzle. Ed. T. Winograd, Stanford University.

[9] Robertson, G.G. MacKinlay, J.D. and Card, S.K. "Cone trees: Animated 3D visualizations of hierarchical information." In HCI 91, pages 189-194, 1991.

[10] A conceptual Framework and Toolkit for Supporting the Rapid Prototyping Context-Aware Applications, Anind Dey, Gregory Abowd.

[11] A Survey of Context-Aware Mobile Computing Research, Guanling Chen and David Kotz.

[12] IBM Research Report: Building Context-Aware Applications with Context Weaver, IBM Research Division, Norman Cohen, et al. 22-10-2004.

[13] Context-Aware Computing Aplicaciones, Bill Schilit, Norman Adams and Roy Want, Xerox Parc.

- [14] A Survey of Context-Aware Mobile Computing Research, Guanling Chen and David Kotz.
- [15] [SAW94] Context-Aware Computing Applications
- [16][JB1] Adding Generic Contextual Capabilities to Wearable Computers
- [17] Bill N. Schilit and Daniel Duchamp. Adaptive remote paging for mobile computers. Technical Report CUCS-004-91, Columbia Univ. Computer Science Dept., February 1991.
- [18] <http://www.sg.com.mx/content/view/338>
- [19] Bentallah_CTDS09_Mashups and SaaS (PDF- Boualem Benatallah (University of New South Wales, Australia/University of Blaise Pascal, France))
- [20] Services Mashups, The New Generation of Web Applications , IEEE Computer Society – Octubre 2008.
- [21] Mashups: The new breed of Web app An introduction to mashups, IBM , Duane Merrill , 24/07/2009.
- [22] [http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid))
- [23] Holmes, Josh. "Enterprise Mashups" (<http://msdn.microsoft.com/en-us/architecture/bb906060.aspx>). MSDN
- [24] Enterprise Mashups: The New Face of Your SOA" (<http://soa.sys-con.com/node/719917>). <http://soa.sys-con.com/> : SOA WORLD MAGAZINE.
- [25] A Context-Aware Mobile Mashup for Ubiquitous Web, Diego Lopez Ipina, Inaki Vazquez y Joseba Abaitua, pag. 3
- [26] <http://www.edbt.org/Proceedings/2008-Nantes/papers/N1124F.html>
- [27] Geographics extensions for http transactions - Daviel, A., Kaegi, F.A, Kohfal, M. - Internet draft The Internet Society (Septiembre 2007)
- [28] Composite Capabilities/Preference Profiles: Structure and vocabularies 1.0, Klyne, G, et al. - Recommendation, W3C (Enero 2004)
- [29] <http://en.wikipedia.org/wiki/UAProf>
- [30] <http://www.w3.org/TR/DPF/>

[31] <http://www.edbt.org/Proceedings/2008-Nantes/papers/N1124F.html>

[32] <http://www.ubiq.com/want/papers/parctab-wmc-dec94.pdf>

[33] The Anatomy of a Context-Aware Application - Andy Harter, Andy Hopper, Pete Steggles, AndyWard, Paul Webster - *AT&T Laboratories Cambridg.*

[34] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, Walter Van de Velde. "Advanced Interaction in Context. In: 1th International Symposium on Handheld and Ubiquitous Computing" (HUC99), Karlsruhe, Germany, 1999 & Lecture notes in computer science; Vol 1707, ISBN 3-540-66550-1; Springer, 1999, pp 89-101.