



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Escuela Académico Profesional de Ingeniería de Sistemas

**Implementación de un sistema de información personal
para dispositivos móviles utilizando agentes móviles**

TESINA

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Rosa Isabel RUIZ VÁSQUEZ

ASESOR

Jorge Leoncio GUERRA GUERRA

Lima, Perú

2007



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Ruiz, R. (2007). *Implementación de un sistema de información personal para dispositivos móviles utilizando agentes móviles*. Tesina para optar el título de Ingeniero de Sistemas. Escuela Académico Profesional de Ingeniería de Sistemas, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

DEDICATORIA: Con mucho cariño a mis queridos padres que me dieron la vida y han estado conmigo en todo momento; a mi asesor, por compartir sus conocimientos y experiencia, y a todas las personas que me han apoyado, les agradezco de todo corazón.

RESUMEN

IMPLEMENTACION DE UN SISTEMA DE INFORMACIÓN PERSONAL PARA DISPOSITIVOS MÓVILES UTILIZANDO AGENTES MÓVILES

RUIZ VASQUEZ, ROSA ISABEL

Agosto 2007

Asesor : Jorge Guerra Guerra

Grado : Lic. en Computacion

Hoy en día, uno de los campos más relevantes de la orientación del agente se relaciona con los agentes móviles. Es decir, un software autónomo el cual puede detener su ejecución, moverse a otro dispositivo y recomenzar su funcionamiento en el mismo punto. Esta capacidad, en común con las cualidades características de agentes hace que los agentes móviles sean convenientes para trabajar en una red inalámbrica. En general, la solución de problemas complejos no se lleva a cabo por un único agente, sino que es necesario un conjunto de agentes que interactúan entre ellos para la consecución de los objetivos del sistema. La cantidad de información, que fluye en la red, se puede reducir usando agentes móviles, porque los datos se procesan en el mismo dispositivo donde se almacenan.

En la presente tesina describiremos una arquitectura multi-agente para dispositivos móviles, al que hemos denominado SICIP (Sistema de Información para la Consulta de Información Personal) el cual muestra un uso práctico de un sistema de información personal, el cual permite al usuario buscar en Internet y acceder a sus cuentas de correo electrónico en entornos de conectividad limitada. El sistema ha sido desarrollado usando la metodología INGENIAS y para ser implementada en la plataforma Netbeans para agentes móviles.

Palabras clave:

INGENIAS

Sistemas de Información Personal

Gestión de Sistemas de Información Personal

ABSTRACT

IMPLEMENTATION OF A PERSONAL INFORMATION SYSTEM FOR MOBILE DEVICES USING MOBILE AGENTS

RUIZ VASQUEZ, ROSA ISABEL

August 2007

Advisor : Jorge Guerra
Degree : Lic. en Computacion

Nowadays, one of the most relevant fields of agent orientation is related to mobile agents. That is, autonomous software that can stop execution, move to other device and restart running in the same point. This ability, jointly with the characteristic attributes of agents makes mobile agents suitable for working in a wireless network. In general, the solution of complex problems is not carried out by an only agent, but that is necessary a set of agents who interact among them for the attainment of the objectives of the system. The amount of information, which flows in the net, can be reduced using mobile agents, because the data is processed in the same device where it is stored.

In the present research we will describe to an architecture multi-agent for mobile devices, to which we have denominated SICIP (Information system for the consultation of personal information) which shows a practical use of a personal information system, which allows the user to look for in Internet and to access to its accounts of electronic mail in limited connectivity environments. The system has been developed using INGENIAS methodology and will be implemented in the platform netbeans for mobile agents.

Palabras clave:

INGENIAS

Personal System Information

Personal Information Management

ÍNDICE

RESUMEN	3
ABSTRACT	5
INDICE	7
INTRODUCCIÓN.....	11

CAPITULO I : PLANTEAMIENTO DEL PROBLEMA

1.1. Planteamiento del problema.....	14
1.2. Objetivo general.....	15
1.3. Objetivos específicos.....	15
1.4. Metodología.....	17

CAPITULO II : CONCIENCIA DEL PROBLEMA

2.1 Marco teorico.....	19
2.1.1 Definición de PIM (Personal Information Management).....	19
2.1.2 ElCorreo Electrónico en el Sistema de Información Personal.....	22
2.1.3. Principales problemas del E mail.....	24
2.1.4 Funciones Principales del PIM.....	25
2.1.5 Administración de la Información Personal.....	29
2.1.6 De PIM hacia GIM.....	31
2.1.7 Evaluando los Comportamientos y herramientas de la Gerencia de Información Personal.....	36
2.1.8 Acercamientos a estudiar PIM.....	38
2.1.9. Definición de Middleware.....	42
2.1.10. Definición de Web Services.....	49
2.1.11. Definición de Agente.....	52

2.1.12	Albedrío, inteligencia y movilidad.	55
2.1.13	Clasificación De Agentes.	55
2.1.14	Evolución de los agentes.	58
2.1.15	Agentes Móviles	63
2.1.16	El paradigma de Agentes móviles	64
2.1.17	La tecnología de agentes móviles	66
2.2	Estado del arte	70
2.2.1	Los Agentes Móviles y Otras Tecnologías.	70
2.2.2	Comparación Y Tendencias De Los Sistemas De Agentes Móviles.	72
2.2.3	Aplicaciones de los Agentes Móviles.	75
2.2.4	Ventajas de los Agentes Móviles.	79
2.2.5	Desventajas de los Agentes Móviles.	83
2.2.6	Caso de Exito de las Tecnologías de Solución Existentes.	86
2.2.6.1	Middleware	86
2.2.6.2	Web Services.	97
2.2.6.3	Sistema Multiagentes.	114
2.3	Aplicaciones de PIM	119
2.4	Importancia o Justificación del Problema.	129
2.5	Problemática Actual.	132

CAPITULO III : LA PROPUESTA

3.1	La Propuesta.	135
3.2.	Evaluación de las Tecnologías de Solución Existentes.	139
3.3.	Elección de la Tecnología a utilizar.	141
3.4.	Evaluación de las Metodologías de Solución Existentes.	143
3.5.	Elección de la Metodología a utilizar.	150

CAPITULO IV : EL DESARROLLO

4.1 Implementación.....	152
4.1.1 Agente Interfaz.....	152
4.1.2 Agentes Cartero y Buscador Documentos.....	153
4.2 Herramientas y tecnologías utilizadas.....	154
4.3 Mecanismos de Seguridad.....	155
4.4 Arquitectura de SICIP.....	156
4.5 Análisis.....	159
4.6 Modelos Establecidos por la Metodología.....	165

CAPITULO V : EVALUACION

5.1 Factibilidad economica.....	173
---------------------------------	-----

CONCLUSIONES.....	181
--------------------------	------------

RECOMENDACIONES.....	183
-----------------------------	------------

REFERENCIAS BIBLIOGRÁFICAS.....	184
--	------------

Lista de Figuras.....	189
-----------------------	-----

Lista de Tablas.....	190
----------------------	-----

INTRODUCCION

Los avances de las tecnologías de la información y las comunicaciones, y el auge actual de las comunicaciones inalámbricas ha dado lugar a la utilización cada vez más de dispositivos móviles con acceso a redes tales como PDA, Pocket Pc, Ordenadores Portátiles, Teléfonos Móviles, etc.). Estos dispositivos proporcionan al usuario la posibilidad de acceder a la intranet de su empresa, realizar pedidos o navegar por Internet (por ejemplo, para buscar documentos de descripción de productos o para realizar consultas a las bases de datos de clientes) haciendo uso de una red inalámbrica.

El usuario de estos sistemas, mientras realiza el proceso de consulta suele recibir más información de la que desea debido a que los procesos de consulta o navegación por Internet incluyen información prescindible (por ejemplo, banners, enlaces a documentos que ya no existen, consultas de clientes demasiado genéricas, etc.), incrementado así el tráfico necesario para realizar la operación. Esto conlleva un incremento del coste de la comunicación (puesto que este tipo de tecnologías tienen un coste en función del tráfico consumido) que no puede ser controlado por el usuario.

Además del incremento del coste por información innecesaria, otro inconveniente muy habitual en este tipo de redes es que la conexión se vea interrumpida por diferentes motivos (por ejemplo, falta de cobertura) lo que provoca la pérdida de ciertas operaciones realizadas hasta el momento (por ejemplo, la descarga de un documento adjunto, o la petición de información en una página web). Es preciso destacar que estos dispositivos móviles frecuentemente disponen de una capacidad de cálculo limitada y no pueden manejar la gran cantidad de datos que reciben.

Por otro lado, en los últimos años el desarrollo Orientado a Agentes se ha convertido en un nuevo paradigma de Ingeniería del Software [1, 2, 3]. El concepto de agente constituye una potente herramienta de abstracción en el desarrollo de software, que facilita la construcción de sistemas distribuidos, inteligentes y robustos [4].

Habitualmente el agente se define en función de una serie de atributos que debe poseer. Dichas características incluyen la autonomía, reactividad, sociabilidad y proactividad, además de otros aspectos como: racionalidad, movilidad, etc. [5]. En general, la solución de problemas complejos no se lleva a cabo por un único agente, sino que es necesario un conjunto de agentes que interactúan entre ellos para la consecución de los objetivos del sistema. La utilización de varios agentes representa de forma natural la descentralización del sistema y enfatiza la comunicación para gestionar las dependencias entre ellos.

Actualmente dentro del paradigma de la programación orientada a agentes están cobrando especial relevancia los agentes móviles [6], esto es, programas que pueden parar en cualquier momento, trasladarse a otro dispositivo y continuar su ejecución en el punto en que se suspendió lo cual, unido a las características propias de los agentes, permite llevar los cálculos al dispositivo en el que se encuentran los datos disminuyendo así el tránsito de información por una red.

El uso de la programación orientada a agentes móviles podría resolver gran parte de los problemas que se han comentado anteriormente, permitiendo crear aplicaciones, de monitorización de redes o de filtrado y búsqueda de

información. Estas aplicaciones cobran especial relevancia al ser usadas en dispositivos móviles con conexión a redes inalámbricas.

En la presente tesina se presenta un modelo utilizando la metodología INGENIAS para desarrollo de un sistema de información personal, implementado con agentes móviles, que permite a un usuario realizar búsquedas y consultar su correo desde entornos con conectividad limitada.

En el Capítulo I se realiza el planteamiento del problema identificado, así como la definición del objetivo general y los objetivos específicos y la metodología a utilizar. En el Capítulo II, se desarrolla el Marco Teórico, es decir todos los conceptos asociados a los sistemas de información personal. El Estado del Arte, en el cual vemos cual ha sido el alcance de las aplicaciones que se han desarrollado hasta el momento así como el planteamiento de las diferentes posibles soluciones, de las cuales mediante un sistema de indicadores, se elige la mejor, la Justificación o importancia del problema y la problemática actual. En el Capítulo III, se describe de manera general la solución o propuesta, y los criterios de decisión para escoger la tecnología, así como la metodología de solución adecuadas. En el Capítulo IV, se describe de manera más detallada la solución, en cuanto a los requerimientos de la arquitectura, el análisis, diseño, así como el desarrollo del modelo con la metodología escogida. En el Capítulo V, se realiza un estudio de Factibilidad Económica y finalmente se presentan las Conclusiones y Recomendaciones de la presente tesina.

CAPITULO I

1.1 Planteamiento del problema

El rápido desarrollo y grado de penetración de las tecnologías móviles (teléfonos y PDAs), esta proporcionando los mecanismos necesarios para que los usuarios puedan acceder en cualquier momento a todo tipo de información desde cualquier punto, que permite a los usuarios acceder, desde un dispositivo móvil, a información contextual dependiente del lugar desde el que se solicite dicha información.

Los últimos datos vienen de la compañía británica The Mobile World, que asegura que durante este mes de julio, se superarán en el mundo los 3.000 millones de personas con móvil. Para finales de año, se prevé que la cifra llegue a los 3.250 millones (más de la mitad de la población del planeta). La seguridad es básica para los usuarios de móvil del ámbito residencial, así como se vuelve absolutamente primordial para usuarios de redes corporativas¹.

¹ Se llama "red corporativa" al entramado de comunicaciones que permite organizar el acceso de los usuarios, tanto internos como externos, a los recursos informáticos de la empresa. Este entramado incluye tanto la comunicación de datos como la transmisión de voz y vídeo, y se caracteriza esencialmente por el diseño flexible y seguro.

En un entorno donde la información crece de manera exponencial, y en donde la información personal se encuentra dispersa, a pesar de los grandes avances en tecnología móvil, no hay algún sistema que nos permita el acceso rápido y eficaz a nuestra información personal, de tal manera que pueda acceder a información relevante desde cualquier lugar en el momento que se necesite, entonces tenemos la pregunta que nos lleva al desarrollo del siguiente trabajo y que nos hace meditar sobre **¿De qué manera se puede mejorar el acceso a sistemas de información personal desde dispositivos móviles?**.

1.2 Objetivo general

El objetivo principal de este trabajo es proponer una metodología y una arquitectura para el desarrollo de un sistema de información personal (SICIP) para telefonía móvil usando la tecnología de agentes móviles.

1.3 Objetivos específicos

- La implementación de la Arquitectura planteada permitirá mejorar el acceso a la información personal del usuario de manera rápida, simple y eficaz.
- Asimismo la implementación de la arquitectura planteada permitirá el acceso a la información más relevante para el usuario.
- Conocer la infraestructura tecnológica que soporta las aplicaciones móviles, como las redes inalámbricas y dispositivos.

- Conocer los principales ambientes de desarrollo de aplicaciones móviles.
- Conocer y aprender a desarrollar aplicaciones móviles basadas en redes inalámbricas como Telefonía celular².

² Es un sistema de comunicación de alta tecnología telefónica, que se transmite de manera inalámbrica, utilizando ondas electromagnéticas que viajan por aire.

1.4 Metodología

El método que utilizaremos toma como referencia el método del ciclo de vida para el desarrollo de sistemas con la finalidad de que pueda ser desarrollado e implantado posteriormente. El método planteado para la solución del problema planteado consta de 6 fases principalmente y son:

➤ **Detección de la Necesidad.**

Consiste en determinar las necesidades o requerimientos de los procesos, equipos, personas, etc, se requiere de uno no existente de acuerdo al nivel de importancia que manifieste la necesidad.

➤ **Definición del Problema.**

Consiste en delimitar las fronteras y el alcance de las necesidades que se desean atender y sobre la cuales existen posibilidades de definir un proyecto.

➤ **Definición de Factibilidad.**

Consiste en definir el nivel de factibilidad (posibilidades de éxito) para conseguir la solución de las necesidades. Se manejarán 4 niveles de factibilidad que servirán para determinar si un proyecto puede ser exitoso o no, estos niveles son:

- Operacional.
- Técnico.
- Económico.

- Calendarización.

➤ **Planeación del Proyecto.**

Consiste en explicar como será la delimitación del problema, justificando el planteamiento de los objetivos desarrollados inicialmente. En esta etapa se definen los niveles o etapas del desarrollo del proyecto, además de las técnicas y el control que se llevará a cabo.

➤ **Elaboración del Proyecto.**

Consiste en definir el diseño, la elaboración de módulos y la integración de todos los elementos. Se deben de dar a conocer en esta etapa todos los distintos tipos de pruebas y técnicas de análisis de resultados para determinar una posible evaluación al final del proyecto.

➤ **Documentación.**

Consiste en explicar como están compuestos los manuales técnicos y de usuario del proyecto.

CAPITULO II

LA CONCIENCIA DE PROBLEMA:

2.1. Marco Teórico

2.1.1 Definición de PIM (Personal Information Management)³

La Administración de la Información Personal (PIM) es requerida para apoyar las actividades que nosotros, como individuos, realizamos para ordenar nuestras vidas diarias como la adquisición, la organización, el mantenimiento, la recuperación, etc. de la información.

Aunque cada uno tiene que aplicar técnicas PIM en sus vidas diarias, el interés popular a tecnologías PIM ha aparecido recientemente.

La frase "Administración de la Información Personal" primero fue usada en los años 1980 en medio del entusiasmo popular sobre el potencial del ordenador personal para mejorar enormemente nuestra capacidad humana de tratar y manejar la información. El interés a PIM ha aumentado en años recientes, no sólo como un asunto de tecnología, sino con una serie de interrogantes, enfocando mejor el trabajo de un juego diverso de disciplinas científicas y de ingeniería, incluyendo la psicología cognoscitiva, la interacción hombre-máquina, la gestión de datos, la recuperación de documentos y la ciencia de la información.

³ Tomado de la referencia [7]

El almacenaje digital es barato y abundante. El soporte de búsqueda mejorado hace fácil señalar la información que necesitamos, incluso funciona en grandes bases de datos de información no relacionada. La presencia de la informática, las comunicaciones y la miniaturización de dispositivos de procesamiento hace posible para nosotros tomar nuestra información en cualquier parte donde nosotros vayamos.

Pero el interés de la comunidad de investigación de PIM también surge de la conciencia creciente de los problemas que estas nuevas tecnologías a veces crean. La información que antes se luchaba para ordenar en papel ahora es dispersada en múltiples versiones entre el papel y copias digitales y aisladas en usos separados y dispositivos en el mundo entero. Incluso una acción aparentemente simple como el responder a una petición de correo electrónico puede llamar a una tarea que llevaría mucho tiempo, predispuesta al error que requiere la información que se integra de varias colecciones distribuidas de documentos de papel y electrónicos, correo electrónico, páginas de Web, y más. El resultado puede ser que no podemos encontrar lo que buscamos, incluso cuando estamos seguros que esto es parte de nuestra propia información.

La administración de la información personal nos permite reflejar las oportunidades y los desafíos de nuevas tecnologías PIM por dos motivos principales: el deseo de aplicar mejoras de tecnologías digitales al desafío de PIM y el deseo de asegurar que todas las preocupaciones de PIM no se han perdido en la prisa por el desarrollo de software y productos de hardware que exploten las ventajas de la tecnología digital.

El PIM se puede aplicar incluso a la gestión de la información médica en donde cada vez más todos los pacientes llevan una responsabilidad personal de recoger y manejar la información que se relaciona con sus condiciones médicas. La supervivencia incluso personal puede depender de su capacidad de manejarlo.

Sin embargo, esto también genera preguntas importantes acerca de la privacidad y seguridad no sólo para ellos y para los profesionales médicos que los tratan, sino también para clientes, estudiantes, ciudadanos, y empleados por todas partes. Clara - Marie Karat et al. proporciona una descripción profunda de esfuerzos para mejorar la capacidad de la gente de controlar quienes tendrían acceso a su información personal.

Los usos del correo electrónico juegan un papel central en muchas de nuestras vidas. Por ejemplo, muchos de nosotros viven en nuestro correo electrónico durante el día (y en casa de noche), usándolo no sólo para la comunicación, sino también para la ejecución de tareas, la gestión del tiempo, y la administración de documentos. Aún nosotros también a menudo somos confundidos y distraídos por la complejidad creciente de nuestros usos de correo electrónico. Steve Whittaker explora el futuro del correo electrónico en apoyo hacia el PIM, acentuando su papel en la administración de PIM y la Administración de Información de Grupo (GIM). Thomas Erickson explora más lejos la práctica de PIM en un contexto más grande conectado a una red GIM definido por varios papeles de una persona en la vida como, digamos, el padre, el esposo, el amigo, el empleado, y el miembro de equipo.

Los problemas de la fragmentación de la información surgen como un tema importante. David Karger y William Jones estudian varias soluciones prometedoras hacia la integración y la unificación de la información personal y el apoyo de PIM. Pero ¿Los instrumentos existentes y artefactos han aumentado la fragmentación de la información, cómo sabemos que nuevos accesos realmente nos ayudan a manejar nuestra información con el tiempo y a través de situaciones? Diane Kelly explora los desafíos especiales de medida y evaluación de comportamientos PIM e instrumentos.

La mejor gestión de PIM quiere decir que hacemos el mejor empleo de nuestro tiempo cada vez más precioso. Nosotros podríamos gastar menos de nuestro tiempo con las actividades pesadas y predispuestas al error y tomar más tiempo haciendo el empleo creativo, inteligente de la información para conseguir cosas hechas. Como Benjamín Franklin dijo, " No malgasten el tiempo; ya que esto es la materia de la que la vida está hecha. "

2.1.2 El Correo Electrónico en el Sistema de Información Personal⁴

El uso del Correo Electrónico involucra la gestión de la Bandeja de Entrada, folders, búsqueda y son usados para soportar las funciones de la Administración de las tareas del PIM, Archivamiento personal y Administración de Contactos.

⁴ Tomado de la referencia [8]

Tendemos a usar de manera extensa nuestro email, según lo reflejado en la cantidad de tiempo que pasamos usándolo y nuestra evaluación de su importancia en el trabajo diario. El papel del email conduce naturalmente a que esta siendo utilizado para tres principales funciones en la gestión de la información personal (PIM): *la Administración de Tareas, el Archivamiento Personal, y la Administración de Contactos.*

La administración de tareas implica el recordar tareas actuales, el seguimiento del estado de la tarea, y el mantenimiento de la información relevante. La función del email conduce a muchos de nosotros a explotar las funciones de nuestra Bandeja de Entrada para la Administración de Tareas. Dejamos la información sobre tareas actuales allí, sabemos que cuando la abrimos y exploramos su contenido, nos recordarán sobre las tareas principales. Incluso nos enviamos correos para poner mensajes en nuestro Inbox como recordatorios y quizás como enlaces a la información útil. Algunos de nosotros también organizan correos referentes a tareas actuales en las carpetas activas, accediendo a ellos según lo necesitado.

Nosotros también utilizamos el Correo Electrónico para el Archivamiento personal. La información de referencia entregada a través del correo o la información sobre tareas terminadas termina a menudo en las carpetas de correo para el uso futuro. Y para muchos de nosotros, porque el email es nuestra herramienta primaria de trabajo, es también natural que lo utilizemos para almacenar la información de algún contacto.

Los usos originales del email que surgieron hace unos 20 años, no fueron diseñados para el PIM, conduciendo a una *Carencia de la Ayuda Directa para las funciones de PIM* cuando utilizamos el email para las actividades de PIM. Por ejemplo, podemos programar reuniones y citas usando el email, pero el email en sí mismo no proporciona la ayuda dedicada para las funciones que hacen calendarios.

2.1.3 Principales problemas del Email

En el email se presentan 2 problemas de acceso a la información: *el problema de Fragmentación y la Carencia de la Ayuda Directa para las funciones de PIM.*

La fragmentación resulta cuando la información entregada a través del email es dejada allí, y se vuelve a tomar para los usos dedicados de PIM (tales como administradores de contactos, calendarios, listas de actividades, y sistema de ficheros personal del usuario). La información se puede dejar en el email debido al esfuerzo implicado en volverlo a utilizar en una aplicación aparte o a la sensación que es más significativa y accesible en el email.

Carencia de la ayuda directa para las funciones de PIM

Dos arquitecturas y las técnicas de procesamiento de textos – **centralización y extracción de información** - han sido propuestas por los investigadores de la universidad de Sheffield y por los investigadores de la recuperación de datos en Microsoft para tratar estos problemas.

La centralización trata la fragmentación localizándola en el PIM a través de la gestión del email y proporciona la ayuda directa de PIM explícitamente en funciones de construcción del PIM del email. Por ejemplo, *Microsoft Outlook* aplica este acercamiento para proporcionar a la *administración de tareas, administración de contactos, y hacer calendarios con una sola aplicación*. **La extracción de la información** toma la visión opuesta, esperando migrar las funciones de PIM e información del email en aplicaciones dedicadas para proporcionar la ayuda directa de PIM. Trata la fragmentación haciendo los datos del email accesibles a aquellas aplicaciones.

2.1.4 Principales funciones del PIM

Una debilidad del enfoque de la centralización es que los clientes del correo electrónico actuales no manejan bien todas las funciones de PIM ¿Entonces cómo debería ser modificado una aplicación del correo electrónico para apoyar explícitamente el núcleo de las funciones PIM de la Administración de tareas, archivamiento personal, ¿y de la administración de contactos?

La Administración de tareas. Aunque muchos de nosotros dejamos la información relacionada con tareas en nuestras bandejas de entrada como una manera de manejar las tareas, este enfoque no es adecuado cuando nosotros recibimos muchos mensajes. La colección desorganizada de mensajes que se acumulan en la bandeja de entrada disminuye la salida y accesibilidad de mensajes individuales, los cuales son a menudo empujados fuera de vista por la entrada de otros mensajes. *La estrategia alternativa de poner los mensajes en carpetas*

activas tiene la ventaja de agrupar los mensajes de tal forma que ellos pueden trabajarse juntos más eficazmente y más coherentemente. Pero sólo funciona si nosotros desarrollamos el hábito de volver a inspeccionar nuestras carpetas, como la mayoría de nosotros hace con nuestra bandeja de entrada.

*Otra manera de apoyar la administración de tareas es clasificar los mensajes por tarea en la misma bandeja de entrada. Clasificar mensajes de entrada hace más fácil procesar las tareas; pueden contraerse los artículos relacionados en una sola lista de artículos vistos cada vez que nosotros accedemos a la bandeja de entrada. Esto reduce el desorden de la bandeja de entrada total, incrementa la salida de tareas y mejora cómo la aplicación nos recuerda acerca de las tareas corrientes. Varias aplicaciones de visualización se han desarrollado para representar tareas de entrada, incluyendo las representaciones de árbol y representaciones planas de información relacionada a las tareas específicas. Sin embargo, una limitación notable de estos enfoques es su dependencia en los hilos para determinar si los mensajes se relacionan a una tarea común. *Los hilos son conocidos como un indicador débil de tareas debido a la tendencia del tema y las prácticas de respuesta del correo electrónico.* Por esta razón, los autores de [desarrollaron la idea de “thrasks” (tareas e hilos) o colecciones usuario-personalizada basados en hilos. Los usuarios pueden agregar artículos deshilados (o no hilados) a la colección o puede removerlos, así que un thrask representa una colección de tarea más que solo una serie de mensajes.*

Aunque la búsqueda se ha propuesto como una solución a PIM [4], representa sólo una solución parcial para el trabajo de administración.

La búsqueda puede ser eficaz para la información que ingresa ya identificada como pertinente a una tarea dada, nosotros necesitamos nuevos métodos automáticos construidos en el correo electrónico para descubrir y resaltar tareas críticas.

Archivamiento personal. El correo electrónico es un depósito de información importante para la información personal, pero su gestión puede ser problemática. Los usuarios aplican tres estrategias en base al uso de folders, principales (conteniendo mensajes clasificados manualmente), búsqueda y ordenamiento - para acceder a la información archivada. La clasificación manual en las carpetas intenta principalmente organizar la información para hacerlo más accesible después. Pero la clasificación manual es una tarea cognoscitivamente difícil que exige a los usuarios el uso de contextos que puedan predecir el futuro. Como resultado, los usuarios son a menudo incoherentes en sus clasificaciones y también pueden olvidar la existencia de sus propias carpetas a largo plazo. Una carpeta dada puede terminar conteniendo mensajes muy diferentes, o carpetas duplicadas pueden terminar conteniendo materiales muy similares. Esta situación se exagera cuando los usuarios ponen al día sus definiciones de carpeta o agregan nuevas carpetas en respuesta a los cambios en las responsabilidades de su trabajo.

Una solución propuesta es la clasificación asistida en que la máquina aprende las técnicas de análisis de los encabezados y contenido de los mensajes, deriva en definiciones de carpetas, y hace las recomendaciones a los usuarios sobre cómo ellos podrían categorizar su bandeja de entrada.

Aunque la clasificación asistida ha mostrado ser eficaz en las pruebas del usuario, funciona sólo si los usuarios ya han creado las carpetas, y no todos los usuarios lo hacen. Es más, para aquéllos quiénes crean carpetas, la clasificación asistida no puede identificar y crear las nuevas carpetas.

Estas dificultades han llevado a algunos usuarios a dejar de crear carpetas. En cambio, intentan afinar dicha técnica con el problema confiando en la búsqueda u ordenamiento, usando cabeceras de mensaje para acceder a los artículos del correo electrónico guardados a largo plazo.

Pero la búsqueda y ordenamiento tienen limitaciones también. Ordenando por remitente o fecha se aprovecha la habilidad del usuario de recordar información parcial sobre un mensaje, pero acceder a través de un ordenamiento es una manera indirecta de encontrar la información.

No obstante, definir una pregunta de búsqueda puede ser tan difícil como clasificar la información en las carpetas. Y no crear carpetas, ordenar y buscar dirige al desorden de la bandeja de entrada, reduciendo la efectividad de la Administración de Tareas.

Administración de contactos. La gestión de los nombres y direcciones asociadas con los contactos importantes es otra tarea importante del PIM. Mientras más sistemas de correo electrónico puedan ser personalizados para extraer las direcciones de correo electrónico automáticamente en la libreta de direcciones, otra información (como los números de teléfono y las direcciones físicas) deben ser extraídas a mano del mensaje - un proceso tedioso y propenso al error. Pero la mayor parte de la información puede ser automáticamente extraída del correo electrónico; por ejemplo, pueden identificarse los contactos importantes automáticamente a través de la información del mensaje-título (como la frecuencia, la fecha de la comunicación, y probabilidad de respuesta). Teniendo identificado los contactos, debería ser posible para la máquina aprender a extraer automáticamente la información adicional de, digamos, archivo de firma y páginas Web que podrían usarse para llenar los campos de dirección del contacto.

2.1.5 Administración de la Información Interpersonal

Además de ocupar un lugar crítico para PIM, el correo electrónico presenta problemas más complejos que otras aplicaciones de PIM. Una diferencia importante entre el correo electrónico y otros aspectos de PIM es que el correo *electrónico es interpersonal*, sirviendo como una canalización para tareas que involucran dos o más personas. El correo electrónico involucra la administración de la información. En contraste, otras tareas de PIM (como la búsqueda y archivo de información) involucra el manejo de la información auto-generada o auto-descubierta

que normalmente no requiere una contestación. La información del correo electrónico es más compleja y consume tiempo para procesarla por tres razones principales:

Afecta a otros. Las decisiones de proceso de correo electrónico tienen implicaciones directas para el trabajo de otras personas. El correo electrónico es una canalización del trabajo, de manera que el fracaso para responder apropiadamente a un mensaje puede poner en peligro el trabajo de alguien en alguna parte. Recíprocamente, las tareas interdependientes están a menudo sujetas a retrasos debido a la espera por una contestación de otras personas con prioridades diferentes. Tales retrasos pueden dejar mensajes que esperan en la bandeja de entrada (o, a menudo, en las carpetas procesables) para períodos extendidos, flotando a menudo fuera de vista y mente. Los usuarios deben rastrear obligaciones y el estado del mensaje para la información del correo electrónico;

El proceso constante. El fracaso para responder rápidamente los mensajes de los colegas podría comprometer su trabajo. Y el volumen enorme de mensajes que nosotros recibimos de las actividades de trabajo cada día representa el fracaso para tratar los mensajes entrantes que pueden llevar a un atraso de mensajes sin abrir y sin responder en la la bandeja de entrada, comprometiendo así la función de administración de tareas.

La presión es tenaz, pues los nuevos mensajes constantemente demandan atención personal y proceso. En contraste, archivando personalmente los archivos digitales generados, las direcciones del

contacto, y los recursos Web descubiertos tiende a ser la discreción propia del usuario, con menos retrasos externamente impuestos o fechas tope; y

Carencia de contexto. A la información del correo electrónico le puede faltar adecuado contexto, haciéndolo más difícil de procesar.

La mayor parte de la información personal se auto-genera o se auto-descubre, surgiendo en el contexto de metas del usuario específicas e intereses. En contraste, los mensajes de correo electrónico no pueden directamente relacionarse a las metas de un usuario particular o intereses, generándose otros con sus propios objetivos.

Carencia de contexto hace más difícil reaccionar apropiadamente a un mensaje, juzgar su valor, o ponerlo en la categoría correcta.

2.1.6 De PIM hacia GIM:

Gestión de información personal en Contextos de grupos⁵

“Compartir la información personal con redes de grupos e instituciones aumenta las preguntas de los acerca de la privacidad y el mejor control dirigido hacia un nuevo campo de la investigación: El Grupo de Gerencia de información”.

Más de nuestra información personal que se almacena en forma digital, es fácil de compartir. La información personal se puede compartir intencionalmente para facilitar un fin individual. Puede emerger como

⁵ Tomado de la referencia [9]

entrada a un proceso de organización rutinario. Puede salir accidentalmente debido a la desatención o a la ignorancia. A medida que nuestra transición a una sociedad digital continúa y la información personal llega a estar cada vez más disponible, los investigadores y los desarrolladores deben prestar mucha atención a la interacción entre la información y los contextos del grupo.

La gestión de la información personal (PIM) desempeña papeles instrumentales y simbólicos en nuestras vidas. Los artefactos de PIM también desempeñan un papel simbólico formando las impresiones que otros forman de sus usuarios. Por ejemplo: puede ayudar a su usuario a aparecer ser productivo, un profesional bien organizado. Y si los artefactos del PIM diestramente se despliegan y alcanzan sus fines, el desempeño del usuario realza esa impresión.

Aunque PIM es generalmente una actividad privada, creamos a menudo información personal con la idea de compartirla. Un estudiante puede llevar notas y compartirla con un amigo ausente. Un compañero de trabajo puede compartir su calendario más fácilmente a los horarios de las reuniones. Pero el compartir puede conducir a problemas; por ejemplo, las notas del estudiante pueden ser desordenadas y mal organizadas, y el calendario del trabajador puede revelar fechas no importantes como aquellas extensas de algún almuerzo con algún compañero. Cuando se comparte información personal, produce tensiones entre los fines para los cuales se comparte y los fines que no necesariamente puede apoyar.

Las tensiones que ocurren naturalmente mientras que se comparte la información personal son complejas. Pueden retroactuar y afectar las normas que tienen que ver con la estructura y el contenido que es compartido; por ejemplo, el estudiante puede ser motivado para tomar mejores notas, y el compañero de trabajo puede crear entradas falsificadas del calendario para obscurecer sus actividades. Más positivamente, un grupo puede estructurar información en un repositorio compartido, haciéndola útil a todos. Es este ciclo de retroalimentación por el que compartir la información crea las tensiones que forman de cómo la información es compartida, marcan esta área como un foco importante para la atención de la investigación.

Se utiliza la Gestión de la Información del Grupo (GIM) para referir a PIM de cómo funciona en esferas más públicas. Más específicamente, GIM tiene que ver cómo la información personal que se comparte con un grupo, clasificar las normas que son la base de eso que es compartido, tan bien como los participantes negocian las maneras de estas normas en respuesta a las tensiones que son inevitablemente compartidas.

Las aplicaciones de GIM incluyen el email, páginas Web, y los wikis. Un ejemplo de muchos años de GIM es hacer calendarios en línea; cuando primero fué introducido a mediados de los años ochenta, encontró una cierta resistencia de los usuarios, quienes realizaron entradas privadas al calendario se podrían utilizar antes para otros fines con excepción de programar reuniones. Relativamente un nuevo género de GIM son los servicios sociales del establecimiento de una red, que permiten que los

usuarios fijen perfiles personales y que creen enlaces a la gente para mostrar lazos profesionales o sociales; alternadamente se utilizan para los propósitos que se extienden de establecimiento de una red profesional a fechar. También plantean preguntas sobre lo que eligen los usuarios revelar u ocultar, cómo su acceso a la información personal se relaciona con los fines que esperan alcanzar.

Otro ejemplo de GIM son los expedientes médicos de pacientes que consisten en información generada por muchas personas (y dispositivos) y es accesada por gente de diversas instituciones para los propósitos que se extienden de la coordinación desde el cuidado hasta la facturación del seguro. Este uso plantea preguntas éticas, legales, y prácticas complejas referentes a la privacidad, el acceso, y a la propiedad.

Para ganar una comprensión más amplia de estos usos, considere un modelo simplificado de GIM que implica a una persona que genere la información que debe ser compartida con un grupo en ayuda de una tarea:

La persona genera la información. ¿Cómo debe ser compartida la información a ser creada? ¿Qué información la gente podría elegir compartir? ¿Por qué ellos desean compartirla? ¿Cómo ellos podrían estructurarla? ¿Qué normas deben tomar en cuenta al compartir? ¿Cómo las normas varían según la forma, contenido, y tarea? Y cómo ellos lo negociaron?;

Con respecto al grupo. ¿Quién compartirá la información? ¿Cómo es la gente que la seleccionará? ¿Cómo esta audiencia imaginada afecta la naturaleza de la información que se compartirá? ¿Cómo es la información estructurada permitida del grupo para ser utilizada, y qué proceso debe ser utilizado para negociar una estructura compartida? Cuáles son las consecuencias de la información "que se escapa" más allá de las ¿audiencias previstas? Y cómo los sistemas GIM se diseñan para apoyar a usuarios que se ocupan de estos usos?; y

Apoyar una tarea. ¿Qué sucede cuando la información compartida resulta ser útil para otras tareas que no estén en los intereses del dueño? ¿En qué medida es posible dar a los usuarios control sobre las aplicaciones de su información personal? ¿Qué clases de control-conocimiento del uso, corrección de errores, información del estado de la finalización de la tarea es factible para proporcionar al dueño de la información?

Estos usos se presentan de un modelo de información compartida que es muy compleja. Contraria a sus implicaciones, la información personal no es producido siempre voluntariamente (las Cuentas de Crédito por ejemplo) o por una sola persona (expedientes médicos). Los grupos con los cuales se comparte la información son raramente estáticos; es nueva gente que construye, incluso mientras que los miembros establecidos salen de los grupos. La información no se comparte necesariamente con los individuos identificables, responsables; puede en lugar de otro ser

compartida con una organización en la cual tiene gente que cumple los mismos roles. Ni es la información personal usada necesariamente para una sola tarea. Y la naturaleza que diferencia de tareas (tales como usar los mismos expedientes médicos para coordinar el cuidado y para la facturación del seguro) puede introducir las tensiones que forman el contenido y la estructura de información, así de cómo esta es categorizada, accesada, controlada, y utilizada.

2.1.7. Evaluando los Comportamientos y herramientas de la Gestión de la Información Personal⁶

Cómo estudiar y entender la naturaleza del comportamiento del PIM que exhibimos naturalmente en nuestro hogar, en el trabajo, y los intereses de las herramientas futuras de PIM.

La naturaleza de la gerencia de información (PIM) implica acercamientos personalizados a los comportamientos y a evaluar el PIM de la gente que entiende las herramientas diseñadas para apoyarlas. La gente crea y tiene acceso a colecciones de su información personal por períodos largos de tiempo, ejecutando una variedad de tareas de gestión de información y exhibiendo una gama de comportamientos que son a menudo únicos a sus colecciones, herramientas, ambientes, preferencias, y contextos. Camine por cualquier vestíbulo de la oficina y usted verá tantas variaciones de PIM como hay gente en las oficinas.

⁶ Tomado de la referencia [10]

Algunos tienen apilados papel que cubren sus escritorios, gabinetes, y estantes para libros; otros tienen tableros de escritorio casi vacíos, quizás, con un solo apilado de documentos y de carpetas cuidadosamente pedidos del archivo; otros documentos se archivan lejos en carpetas que cuelgan, y los libros son organizados alfabéticamente por el nombre del autor en estantes, como si estuvieran en una biblioteca. Pida que esta gente recupere cualesquiera de estos artículos, y sus comportamientos serán probablemente tan diferentes como sus estrategias de organización. ¿Cómo pueden tener tales diversos comportamientos en circunstancias similares? ¿Son las prácticas de PIM entre dos personas semejantes? ¿Qué podemos aprender de esta variación para diseñar el software futuro de PIM?

Mientras que podemos analizar un cierto comportamiento de alto nivel de PIM (tal como organización y recuperación), en el nivel operacional PIM parece ser tan único como somos. Así, usando métodos de evaluación y herramientas no adecuadas, es probable que sea una estrategia menos que ideal para estudiar algo tan complejo como el PIM. Varios desafíos metodológicos se deben tratar cuando se diseñan los estudios del proyecto de los comportamientos y de las herramientas de PIM. PIM es una actividad en curso hecha a menudo con anticipación de las acciones futuras o de los usos esperados (tales como compartir objetos de información). Porque los acontecimientos básicos de PIM ocurren a menudo en las horas imprevisibles, es difícil programar ellos como parte de un protocolo experimental. PIM abarca una gama de

actividades y de herramientas; entender apenas una de ellas proporciona solamente un cuadro parcial de lo que desean los usuarios lograr y de cómo puede ser que lo logren. La gente debe ser observada en sus ambientes naturales; en su hogar, en el trabajo, y mientras tanto ellos son estudiados por el PIM en tiempo real, registrando el proceso y las consecuencias del comportamiento. Estos desafíos también se hacen necesario para estudios de laboratorio de comportamientos y de herramientas para entender más sobre el comportamiento general de PIM. Solo el análisis detallado nos permitirá entender como es el comportamiento de PIM, entonces construir las herramientas para apoyarlo y realizar evaluaciones válidas y confiables.

2.1.8 Acercamientos a estudiar PIM⁷

Una variedad de acercamientos está disponible para estudiar PIM así como los comportamientos de la gente en la vida real. Los acercamientos naturalmente, incluyendo el trabajo en el terreno y la etnografía, son apropiados, puesto que permiten que la gente realice comportamientos de PIM en ambientes familiares, donde se podrán utilizar las herramientas de análisis adecuadas. En los casos de estudios, la atención que se enfoca en un usuario particular o en un número pequeño de usuarios, los cuales son valorados para estudiar el comportamiento de PIM. Los resultados pueden motivar otros estudios donde las hipótesis específicas pueden ser planteadas.

⁷ Tomado de la referencia [11]

Dichos acercamientos nos permiten capturar los datos por un periodo largo y tomar medidas de su comportamiento en el tiempo; ellos pueden ser combinados con investigaciones y casos de estudio. Otro desafío importante en los acercamientos al estudio de PIM está determinando por un intervalo apropiado para tomar medidas y la longitud del tiempo que conduce el estudio. Cuando usted sepa cuán importante es qué usted mida, así como el comportamiento de PIM varía basado en lo que la gente realizará para intentar lograrlo en el momento dado. Por ejemplo, los acontecimientos externos (tales como días de fiesta) interrumpen o cambian a menudo a las actividades de la persona y comportamiento, afectando posiblemente las clases de PIM que se realicen.

Mientras que la calidad y la cantidad de datos que recopilamos con estos acercamientos pueden ser ricas, extensas, y variadas, tales como intensivas y los acercamientos personalizados a la información personal no tendrán límites. Los estudios profundizados de sistemas pequeños en un cierto plazo, los participantes son costosos, y los resultados no se pueden generalizar a poblaciones más grandes. Los acercamientos personalizados a la evaluación implican otros costes también, algunos aparecen durante el análisis de datos.

Dependiendo del número de participantes, puede ser que terminemos de modificar diversos instrumentos de análisis de requisitos particulares,

conduciendo diversas pruebas estadísticas, o crear diversos modelos de comportamientos. Los resultados pueden ser muy complejos y difíciles de describir de manera sucinta. Así, es importante seguir estos acercamientos con estudios de laboratorio para refinar, explorar, y ampliarse en los resultados y para contribuir al desarrollo de teorías generales del comportamiento de PIM.

La investigación de PIM puede contener una colección de objetos de información recogida en tiempo real que describe usuarios originales, tareas, situaciones, y comportamientos. Mientras que la construcción buenas estructuras de información es difícil, con trabajo intensivo y alto consumo de tiempo, tiene el potencial de facilitar el progreso en diseñar la tecnología de PIM. Por ejemplo, el estudio del PIM permite modos múltiples de investigación (tales como comparar técnicas de PIM, hipótesis alternativas que examinan, y la replicación de resultados anteriores).

Desarrollar un sistema común de tareas compartibles tiene el potencial de facilitar la investigación de PIM. Crear y usar las tareas compartibles para la investigación que conduce en el campo de la interacción entre el hombre y el computador pueden conducir a una investigación más holística, más incremental, y generalizable. Identificar las tareas para la investigación de PIM es intrínsecamente difícil porque PIM posee las tareas que se pueden definir en muchos niveles de la especificidad. Por ejemplo, al "hacer un email" se puede subdividir en por lo menos en

cuatro tareas separadas: buscar para un dato específico del email; su administración y gestión de archivos; su configuración y acceso a un directorio; así como su lectura. Es difícil crear tareas bien definidas cada vez que uno está interesado en conducir un nuevo estudio de un ambiente particular de la población o del usuario. Por otra parte, sin una cierta semejanza en tareas a través de estudios, es difícil comparar resultados y desarrollar teorías del comportamiento de PIM.

Desarrollar la métrica válida y confiable para el estudio del comportamiento de PIM y la evaluación de las herramientas de PIM es una parte importante que necesita mucha atención. Procedimientos científicos rigurosos donde se proporcionan y se justifican las definiciones conceptuales y operacionales, en donde una variedad de técnicas se utilizan para establecer la validez y la confiabilidad de medida- se utilizan raramente. Por lo tanto, la investigación y la teoría referente a comportamiento de PIM y las herramientas han sido desestimadas, puesto que es difícil acumular, comparar, e integrar resultados a través de estos estudios.

Para progresar en la investigación de PIM se debe emplear una variedad de acercamientos y esforzarse para apoyar progresos teóricos, experimentales, y prácticos. Junto con la proliferación de las aplicaciones de PIM, debemos desarrollar los métodos y las métricas de evaluación que producen conocimiento válido, generalizable,

compartible sobre cómo los usuarios van sobre las actividades y las interacciones de PIM en sus vidas diarias.

2.1.9. Definición de Middleware⁸

El middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores.

El esquema tradicional cliente/servidor permite que diferentes módulos de aplicación se comuniquen directamente sin una capa intermediaria.

Se presenta problemas en sistemas complejos con componentes de diversos proveedores la comunicación resulta poco flexible e inoperante.

Middleware es un término, como muchos otros en este campo, ampliamente usado y posee muchas definiciones. De manera informal se le llama *plumbing* porque conecta partes de aplicaciones distribuidas con *pipes*⁹ de datos y favorece el intercambio de datos entre ellas. También se llama tecnología del *glue*, porque se utiliza para integrar componentes heredados.

Middleware es la capa de software que se coloca entre el usuario y el entorno distribuido, abstrayendo al usuario de la complejidad y la heterogeneidad de las arquitecturas, protocolos, sistemas operativos, lenguajes de programación; es decir, otorga la posibilidad de intercomunicar aplicaciones desarrolladas en distintos lenguajes de programación, sistemas operativos y plataformas. En otras palabras, engloba los elementos que permiten la comunicación entre los sistemas.

⁸ Tomado de la referencia [12]

⁹ Pipes.- consiste en una cadena de procesos conectados de forma tal que la salida de cada elemento de la cadena es la entrada del proximo. Es comun el uso de datos entre elementos consecutivos(Wikipedia)

Orientación en la Selección de Tecnologías Middleware

En este punto se presentan algunas consideraciones útiles para orientar al ingeniero de software en la formulación del Portafolio de Proyectos de Integración tomando en cuenta los Niveles de Integración y los tipos de tecnologías antes descritos. Con el propósito de visualizar mejor su aplicabilidad se identificaron las ventajas y las desventajas de estas tecnologías Middleware, las cuales se resumen en la Tabla 1:

MIDDLEWARE	VENTAJAS	DESVENTAJAS
DT	<ul style="list-style-type: none"> • Permite compartir datos entre aplicaciones, con poco impacto en el rendimiento. • Provee un modelo poderoso para manejar sistemas de objetos distribuidos 	<ul style="list-style-type: none"> • Sólo es recomendable cuando los datos no se actualizan con mucha frecuencia.
RPC	<ul style="list-style-type: none"> • Por ser el primer tipo de Middleware es el menos complicado de usar y comprender. • Por ser síncrono, provee mayor integridad de datos. 	<ul style="list-style-type: none"> • Exige alto nivel de procesamiento. • Tiene bajo rendimiento.
MOM	<ul style="list-style-type: none"> • Por ser asíncrono, un proceso puede enviar una solicitud y continuar ejecutándose sin recibir respuesta de culminación de esta. • Es una buena elección cuando se tiene poco ancho de banda o cuando la red es inestable. 	<ul style="list-style-type: none"> • No aplican cuando se requiere la respuesta de una solicitud para continuar el proceso. • No soporta procesos de negocio.
DOM	<ul style="list-style-type: none"> • Aplican en ambientes orientados a objetos. 	
CORBA	<ul style="list-style-type: none"> • Útil para desarrollar sistemas grandes, soporta multilinguaje de programación. • Soporta plataformas y sistemas operativos heterogéneos. 	<ul style="list-style-type: none"> • Tiene menor desempeño que RMI. • Al ser tan amplio, es más complejo que RMI y DCOM.
RMI	<ul style="list-style-type: none"> • Gracias a la máquina virtual Java soporta • multiplataforma. 	<ul style="list-style-type: none"> • Sólo para objetos Java.
DCOM	<ul style="list-style-type: none"> • Soporta multilinguaje 	<ul style="list-style-type: none"> • Especialmente diseñado para trabajar sobre sistemas operativos de Microsoft®

MIDDLEWARE	VENTAJAS	DESVENTAJAS
TP Monitors	<ul style="list-style-type: none"> • Útiles cuando se necesita tener control de las transacciones que se procesan • Las aplicaciones que ejecutan las transacciones generalmente acceden bases de datos y sistemas de comunicaciones. • La asignación de recursos no es de forma permanente • Comparten y reutilizan recursos 	<ul style="list-style-type: none"> • No son útiles para el intercambio de información.
DBAT	<ul style="list-style-type: none"> • Es un estándar muy utilizado. • Encapsula la complejidad de la conexión con la base de datos. 	<ul style="list-style-type: none"> • Puede resultar mas lento, el intercambio entre la aplicación y a base de datos, que con el uso de otras tecnologías.
COF	<ul style="list-style-type: none"> • Proporciona lineamientos para el uso apropiado de DOM. 	<ul style="list-style-type: none"> • Puede resultar complejo en ambientes altamente distribuidos.
DS	<ul style="list-style-type: none"> • Más sencillo de manejar que una BD. • Se utiliza para el almacenamiento de: <ul style="list-style-type: none"> ○ Información de usuarios (autenticación y autorización). ○ Información del hardware de la red. ○ Agendas de direcciones. ○ Configuraciones de programas. 	<ul style="list-style-type: none"> • Su uso es específico para entidades de red como aplicaciones, archivos, impresoras y usuarios.
AS	<ul style="list-style-type: none"> • Provee servicios para la administración de procesos (tal como desarrollo, monitoreo y alimentación de procesos) que son compartidos por múltiples aplicaciones. • Centraliza la lógica de las aplicaciones, • haciendo que la administración de cambios sea más sencilla. 	<ul style="list-style-type: none"> • Se utilizan sólo en ambientes multicapas (actualmente, es el caso de la mayoría de los ambientes)

Tabla 1. Ventajas y Desventajas de los Middleware.

Al analizar la Tabla 1 y las definiciones antes mencionadas, se puede decir que no es posible hacer una comparación entre estos tipos de Middleware, debido a que los mecanismos que las soportan están diseñados para aplicarse en ambientes y situaciones diferentes (escenarios). Por lo tanto, estos tipos de Middleware pueden coexistir; es decir, no son excluyentes. Aún más, existen tipos que están estrechamente relacionados, sobre todo si pertenecen al mismo

fabricante. Sin embargo, las comparaciones son posibles entre tipos de tecnologías Middleware que se basan en el mismo paradigma.

Con base en la reflexión anterior, los niveles de integración definidos por Schmidt y tomando en cuenta la información de la Tabla 1 con algunas consideraciones planteadas por los autores antes mencionados, se describen varios escenarios y las orientaciones propuestas para decidir sobre la tecnología Middleware apropiada. Cabe señalar que tal como se mencionó anteriormente, el paso de un nivel a otro puede representar una evolución tecnológica por parte de la empresa; por lo tanto, algunas tecnologías Middleware que propician en mayor medida el nivel 2, por ejemplo, pueden estar presentes en niveles superiores, aún cuando no apoyen dicho nivel directamente, sino más bien a alguna otra tecnología que sí soporta directamente a dicho nivel.

- □ El nivel 1 no contempla la presencia de tecnología Middleware; sin embargo, en muchas aplicaciones es frecuente el uso de DBAT para comunicarse con las bases de datos.
- □ El Middleware DT es un framework el cual permite "publicar" información en un Tuple Space (TS) que a su vez permite a quienes se "suscriban" utilizar dicha información. Proporciona integración sólo a través del compartimiento de datos. Mantiene duplicidad de la data, con lo cual sacrifica espacio en disco y posible falta de integridad, pero puede resultar beneficioso para el rendimiento tanto de la aplicación que publica como para los suscriptores de la información, debido a que el proceso de una aplicación no interfiere con el de la otra. Es

recomendable sólo si el suscriptor puede trabajar con información que pudiera no estar actualizada al momento.

- Los DT pueden estar en los niveles 2, 3 y 4, dependiendo como se combine con otras tecnologías, debido a que por si solo se ubica en el Nivel 2.
- Si entre los sistemas que se desean integrar existe alguno heredado del cual no se posee código fuente, la posibilidad de integración que se tiene es a través de datos, puesto que actuarían como caja negra y sólo se tendrá acceso a la salida de datos que genera. Es decir, se puede manejar un DT o vía API para comunicarse desde una aplicación a la base de datos de software heredado. Nuevamente, lo anterior implica que la empresa estará en el Nivel 2.
- Para alcanzar el Nivel 3 se debería obtener una integración que involucre los procesos del negocio, lo cual es posible si se utilizan tecnologías Middleware como DOM en aplicaciones donde probablemente se tiene el código fuente y se pueden crear lazos entre las mismas para que trabajen en equipo y colaboren entre si. En este caso, lo recomendable es un estudio de la plataforma presente en la empresa, de manera que la primera sugerencia es trabajar usando tecnología del mismo fabricante. Además, se debe considerar si los sistemas de software involucrados permiten o no orientación a objetos. De no permitirlo, la alternativa es el uso de RPC, pero si lo permite, lo apropiado es DOM, lo cual viene resaltar la sugerencia anterior, de modo que si la plataforma es Windows®, por ejemplo, es recomendable un desarrollo basado en DCOM, el framework deberá ser COM+ (con

.NET), pero si además lo que se quiere es comunicar aplicaciones Web, el servidor apropiado es IIS, y si se requiere comunicación con alguna base de datos, entonces se deberá usar el API de comunicación ODBC.

- Por su parte, el Nivel 4 requiere integración externa. En este caso, conviene evitar el uso de RPC, debido al alto consumo de procesamiento que requiere, lo que baja notoriamente el rendimiento. Bajo las mismas recomendaciones hechas anteriormente, es posible el uso de DT y/o DOM (todo dependerá de la necesidad de integración: datos, procesos o ambos). Sin embargo, actualmente la tendencia en aplicaciones e-commerce, es usar MOM debido a que permite una comunicación asíncrona entre aplicaciones de software. Por ejemplo, al efectuar una compra por Internet, el comprador introduce los datos de la compra y de la forma de pago, los envía y posteriormente él recibe información que especifica si sus datos están o no en orden; es decir, hubo un envío de “mensaje” a una aplicación que se encarga de validar la transacción bancaria y una vez hecho esto se envían “mensajes” notificando el estado de la compra (como se aprecia, la compra realmente no se hace en línea puesto que los mensajes son asíncronos). Los mensajes suelen usar Extensible Markup Language (XML) debido a la flexibilidad y uniformidad con que puede ser intercambiada Información.
- Las tecnologías como TPM y para manejo de base de datos, son posibles en los niveles 3 y 4. Sin embargo, TPM propicia en mayor medida el nivel 3 (debido a las funciones que realiza y su alto costo). Los TPM son especialmente empleados en los sistemas bancarios, donde es

de vital importancia el monitoreo de los movimientos bancarios, en cuyo caso es muy importante conocer el éxito o el fracaso de la transacción, de manera que de existir un fallo (tal vez de luz o conexión) sea posible tomar acciones, las cuales generalmente revierten la transacción.

- Los grandes sistemas (por ejemplo bancarios) se encuentran en plataforma UNIX, lo que imposibilita el uso de componentes Microsoft®, debido a que no son multiplataforma.
- □ Los API de base datos se pueden incorporar en cualquier Nivel de Integración para comunicar las aplicaciones con la base de datos, en este caso la recomendación sigue siendo mantener la plataforma utilizada en la empresa.
- Debido a la flexibilidad que ofrecen las tecnologías Middleware como DOM, ellas pueden propiciar además del nivel 3 (como se mencionó anteriormente) los niveles 2 y 4.
- □ Las consideraciones de uso de COF dependen nuevamente de la plataforma. Ahora bien, en sistemas altamente distribuidos (sobre todo si estamos en nivel 4) es posible que se presente la necesidad de combinar tecnologías. Para lo cual se tienen alternativas como integración entre COM y CORBA desarrollada por IONA Technologies o integración entre CORBA y RMI desarrollada en conjunto por OMG y SUN Microsystems. Por estar esta tecnología estrechamente relacionada con DOM, de la misma manera puede apoyar los niveles 2, 3 y 4.
- Los Middleware tipo DS son fundamentales en una red moderna, debido a que los servicios que proporcionan permiten identificar y administrar

las relaciones entre todos los elementos de la red. En ese sentido, y aunque su funcionalidad puede extenderse, es posible encontrarlo a partir del nivel 2.

- Los Middleware AS se utilizan principalmente en aplicaciones WEB, las cuales se ubican regularmente en los Niveles 3 y 4.

2.1.10. Definición de Web Services:¹⁰

El avance de Internet y la rápida aparición de nuevas tecnologías asociadas al web están facilitando que los sistemas de información se universalicen, permitiendo su acceso a cualquier usuario potencial conectado a internet. Las características novedosas que plantea el desarrollo de aplicaciones en este contexto hacen que tenga sentido introducir el término de “Aplicación Web”, referido a la nueva familia de aplicaciones informáticas especialmente modeladas y diseñadas para ser ejecutadas en la web. Análogamente, empieza a utilizarse el término “Ingeniería Web” para referirse al conjunto de métodos, técnicas y herramientas que deben de ser utilizadas para abordar el desarrollo de tales aplicaciones. En este contexto, se hace imprescindible disponer de métodos de desarrollo que proporcionen soluciones al problema de informatizar sistemas en la web. Actualmente se están imponiendo las arquitecturas basadas en Web Services XML. Los Web Services facilitan el acceso a la funcionalidad de las aplicaciones a través de Internet, permitiendo acceder a servicios desde cualquier punto de la red simplificando su utilización y facilitando la interoperabilidad entre

¹⁰ Tomado de la referencia [13]

servicios y entre aplicaciones lo que permite integrar la funcionalidad de distintas aplicaciones empresariales. Esta tecnología facilita el paso de arquitecturas de 2 niveles a arquitecturas de n-niveles, suponiendo además un gran avance respecto a la tecnología actual de componentes software porque permite realizar alquiler de servicios externos frente a la compra o desarrollo de componentes. Su ámbito actual de aplicación es el comercio electrónico, aunque se está extendiendo a todos los dominios de aplicación. Durante los próximos años uno de los temas que va a ser objeto de un intenso trabajo de investigación es el desarrollo de métodos de producción de software que permitan construir aplicaciones web complejas con comportamiento dinámico, que sean compatibles con los estándares metodológicos y notacionales más extendidos en la actualidad (en particular UML) y que establezcan cómo diseñar y desarrollar aplicaciones web basadas en Web Services XML para integrar aplicaciones existentes, realizar comercio electrónico (B2B, C2B) y ofrecer funcionalidad a través del web. En este trabajo se presenta una propuesta que constituye una extensión de un método de modelado de aplicaciones web llamado OOWS, que pretende dar solución a estas necesidades metodológicas.

Ventajas:

- Interoperatividad, pues puede usar XML como formato de intercambio de información y un conjunto de protocolos estándares abiertos para transmitir por ello se pueden comunicar componentes escritos en diferentes lenguajes y distintas plataformas.

- Fácil de implementar. No es costoso: se usa una infraestructura existente.
- La mayoría de las aplicaciones pueden re-empaquetarse como Web Service.
- Reusabilidad del software

Desventajas:

- Dependencia de la disponibilidad de servidores y comunicaciones.
- Dependencia del Ancho de Banda y tráfico de red.

2.1.11 Definición de Agente¹¹

La primera definición más general dada por W. Lecky-Thompson es: *‘Un agente es una pieza de software que ejecuta una tarea dada usando información obtenida de su entorno para actuar de forma apropiada para completar la tarea exitosamente. El software debe ser capaz de adaptarse a sí mismo en base a los cambios que ocurren en su entorno, para que un cambio en circunstancias le permita aún obtener el resultado deseado’.*

Un segundo intento más técnico orientado a la tecnología de objetos y agentes móviles es el dado por Fritz Hohl: *“Un agente móvil es un objeto especial que tiene un estado de datos (otros objetos no agentes, estructuras y bases de datos), un estado de código (las clases del agente y otras referencias a objetos) y un estado de ejecución (el control de procesos que se ejecutan en el agente)”*

Como ninguna de estas definiciones resulta completa en vez de dar una definición formal, proporcionaremos la lista de características que se espera que un agente deba tener, para poder tener una idea de lo que un agente puede ser. Las características siguientes suelen tenerlas los agentes móviles que serán el tipo de agentes a las que esta dedicado este trabajo.

¹¹ Tomado de la referencia [14]

La forma más general en que es usado el término agente es para denotar un sistema de software basado en agentes, en este ámbito un agente debe tener las siguientes propiedades:

- **Autonomía:** un agente opera sin la intervención directa de humanos y debe tener una cierta clase de control sobre sus acciones y su estado interno.
- **Habilidad social:** los agentes interactúan con otros agentes y (posiblemente) con humanos.
- **Reactividad:** los agentes perciben su entorno y responden en un tiempo razonable a los cambios que ocurren en él. El agente puede estar en estado pasivo la mayor parte del tiempo y despertar al momento de que detecte ciertos cambios.
- **Proactividad:** los agentes no solo responden a cambios sino que pueden tener un comportamiento con una iniciativa propia hacia una meta dirigida.
- **Continuidad temporal:** los agentes están constantemente ejecutando procesos ya sea en forma activa o pasiva.
- **Orientación hacia el objetivo Final:** El agente es capaz de desarrollar una tarea compleja. Para lograrla es necesario subdividir esta tarea en pequeñas subtareas, el agente debe decidir por si mismo la mejor manera y orden de ejecutarlas para lograr el objetivo final.

- **Movilidad:** El agente debe ser capaz de suspender su ejecución en un servidor y reanudarla en otro servidor una vez que se haya desplazado a este. Este concepto se ha introducido en los últimos años.

Para algunos investigadores, especialmente para los que trabajan en JA, el termino agente tiene un significado más fuerte y específico. Estos investigadores normalmente pretenden que un agente sea como un sistema informático que además de tener las características mencionadas previamente posean conceptos aplicados normalmente a los humanos, así pues podremos añadir nuevas características a los agentes para el ámbito de IA:

- **Benevolencia:** es la suposición de que los agentes no tienen metas conflictivas y que cada agente por consiguiente tratara de hacer la tarea que le fue encomendada.
- **Racionalidad:** es la suposición de que el agente siempre actuara para lograr sus metas y de que no actuara de una forma que evite la consecución de las mismas.
- **Adaptabilidad:** un agente debe ser capaz de ajustarse a sí mismo a los hábitos, métodos de trabajo y preferencias de su usuario.
- **Colaboración:** un agente debe tomar en cuenta que los usuarios humanos cometen errores, por ejemplo pueden omitir información o proporcionarla de una forma ambigua. Un agente debe ser capaz inclusive de negarse a ejecutar ciertas tareas si éstas implican una

sobrecarga para los recursos de la red o si ocasionase daño a los usuarios.

Aunque aún no hay un solo agente que posea todas estas habilidades, existen sistemas de agentes prototipo que poseen muchas de ellas. Además nos proporcionan una buena idea de lo que un agente puede hacer, así como que características debemos de evaluar al examinar la calidad de un sistema de agentes en específico.

2.1.12 Albedrío, inteligencia y movilidad.

El grado de albedrío (de la lengua inglesa *agency*), inteligencia y movilidad que puedan tener los agentes de un sistema son los factores principales a medir para evaluar la calidad de un sistema de agentes, estas características son las que fundamentalmente les proporcionarán a los agentes un mayor grado de versatilidad para la consecución de cualquier tarea.

2.1.13. Clasificación de agentes:

Los agentes pueden ser clasificados en una variedad de formas: por sus características, por la tecnología que los soporta, por la función que desarrollan (o lo que hacen), o por una mezcla de ellos. La clasificación de agentes no es única, así como algunas de las características que de los agentes usadas para su clasificación no son mutuamente exclusivas.

Los agentes en base a su movilidad.

Basados en su movilidad los agentes pueden ser clasificados como:

- **Agentes estacionarios o estáticos:** un agente estacionario reside en un sistema informático y ejecuta sus funciones y o tareas, interacciona con el mundo exterior intercambiando mensajes. De forma parecida al software convencional.

- **Agentes móviles:** son aquellos que ejecutan las tareas designadas en distintos ordenadores moviéndose de uno a otro para recopilar la información requerida y posteriormente regresar a su sitio de origen.

Los agentes en cuanto a su característica fundamental.

Como habíamos mencionado anteriormente las características fundamentales que hacen valer más o menos a un agente son: autonomía, inteligencia y movilidad. Estas tres características dan origen a otra clasificación que se ha hecho sobre los agentes que es:

- **Agentes inteligentes:** estos están diseñados para razonar interactivamente la información que se les va proveyendo para seleccionar la mejor opción. Son capaces de realizar tareas complicadas y de aprender de su entorno.

- **Agentes autónomos:** aquellos que son capaces de actuar sin intervención humana y que son capaces de tomar decisiones importantes por sí solos. Desde el punto de vista programacional éstos agentes poseen un hilo (thread) propio.

- **Agentes móviles:** aquellos que son capaces de trasladarse de un servidor a otro para reanudar su ejecución en un nuevo sitio destino.

Como se puede apreciar las clasificaciones tienen una gran relación, es decir los agentes autónomos normalmente para lograr un buen grado de autonomía tienen que ser inteligentes. A su vez los agentes inteligentes la mayoría de las veces son autónomos aunque habrá ocasiones en que su inteligencia sólo la denoten al procesar la información que reciben de un usuario interactivamente.

Por último los agentes móviles suelen ser autónomos y en función de su grado de inteligencia esta la complejidad de las tareas que pueden desempeñar. Sin embargo hay muchos agentes móviles que realizan tareas rutinarias como la recopilación de datos en donde su grado de inteligencia es casi nulo.

Diversos tipos de agentes

Debido a la sobrecarga del término que ha sido usada para todo, podremos citar diversos tipos de agentes que mezclan características mencionadas en las otras clasificaciones:

- **Agentes de interfaz:** éstos asisten a los usuarios y proveen interfaces de usuario “inteligentes” que pueden detectar cuando un usuario está teniendo dificultades y asesoran al usuario acerca del problema. Observan y aprenden las preferencias y hábitos del usuario para automatizar acciones que son ejecutadas cotidianamente y sugieren cursos de acción. Finalmente reducen la complejidad de la interfaz favoreciendo una mayor funcionalidad.

- **Agentes colaboradores y sistemas multiagente:** un agente colaborador interactúa y coopera con otros agentes en nombre de un usuario para ejecutar una tarea. Un sistema multiagente consiste en dos o más agentes semi-autónomos que interactúan, colaboran y trabajan juntos para ejecutar un conjunto de tareas o lograr un grupo de objetivos.
- **Agentes de Internet Información:** los agentes de Internet pueden filtrar una gran cantidad de información disponible en la red, pasándole al usuario sólo aquella información en la que está interesado. Ellos automatizan la recuperación y el proceso de la información obtenida de la red.
- **Agentes de aprendizaje:** su principal función es asistir a los usuarios una vez que hayan aprendido de observar e imitar sus acciones y de la retroalimentación conectiva que reciben por parte de los usuarios.

2.1.14 Evolución de los agentes:

Debido a la gran cantidad de agentes que existen su evolución ha sido muy variada ya que han tomado giros distintos en función del ámbito en el que se desarrollan. Definitivamente como lo mencionamos en la introducción, el estudio de agentes comenzó en aplicaciones de inteligencia artificial, sin embargo con la explosión del Web e Internet han nacido nuevas vertientes y tendencias por lo que hoy en día los agentes gozan de una gran relación con Internet.

Diversas etapas

En sus principios una de las primeras compañías que invirtió en la tecnología de agentes fue General Magic que fue fundada en 1990 para crear una tecnología basada en agentes para dispositivos de computación portables. Esta tecnología llamada Telescrip fue desarrollada por AT&T, Motorola y algunos otros.

Posteriormente adaptaron Telescrip para trabajar con servidores en Internet y crearon *Tabriz AgentWare* el cual ejecutaba y gestionaba aplicaciones basadas en agentes que estaban en servidores.

En el campo de inteligencia artificial el estudio de agentes ha seguido avanzando, sobre todo al tratar de crear nuevos agentes que se asemejen más cada día al comportamiento humano y con un alto nivel de raciocinio e inteligencia que son las dos características fundamentales de estos agentes. Sin embargo al paso del tiempo con la explosión de Internet el desarrollo de estos agentes ha sido menguado por la rápida evolución de los agentes de Internet y agentes móviles.

Mientras *Telescrip* era precursora en el campo de agentes móviles se creaba el *Lenguaje de manipulación y consultas de conocimiento (KQML)* que hoy en día sigue vigente.

De igual forma se implementaba el lenguaje de *Agent TCL* en Dartmouth College; *Agents for remote action (ARA)* en la universidad de Kaiserslautern, Alemania y *Penguin*.

Todos estos sistemas que son sistemas de agentes móviles, salvo KQML, siguen vigentes hoy en día y cada vez se realizan nuevas versiones, sin embargo la carencia de éxito en algunos de ellos radica en que son basados en lenguajes propietarios o poco difundidos en la mayor parte de las veces, en donde para poder realizar una aplicación basada en agentes móviles es necesario aprender el propio lenguaje de codificación lo cual no resulta una tarea fácil. Peor aún por ser propietarios una aplicación hecha en telescript no podía interaccionar con otra hecha en ARA por ejemplo. Debido a esto el avance no fue significativo. En 1995 con la aparición de windows 95 el mundo de las interfaces gráficas dominó el mercado completamente y dio lugar a la proliferación de los agentes de interfaces, de hecho Microsoft ofrece un producto llamado Microsoft Agent.

De igual forma la explosión del número de usuarios en Internet sufrió un gran incremento de 1995 a 1996, ya que el número de servidores fue prácticamente duplicado por lo que los agentes de Internet que ya existían tomaban más fuerza.

De igual forma el tráfico excesivo en la red así como el sueño de crear una tecnología que realizará las tareas tediosas en Internet hizo buscar a los investigadores una nueva forma de solventar el problema por lo que se retomó el tema de agentes móviles. Aunado a esto se presenta el gran éxito del lenguaje Java, cuya mayor ventaja era la independencia de la plataforma y ofrecía una base segura para el desarrollo de agentes móviles.

Java se encontraba difundido en todo el mundo y brindaba características fundamentales como la invocación de métodos remotos (RMI) y la serialización para implementar sistemas de agentes móviles. Quizá debido a esto ha sido la plataforma que ha tomado más fuerza para el desarrollo de agentes móviles y de la que se espera ver lo mejor en los próximos años.

De agentes inteligentes a agentes móviles

Poco a poco se ha ido transformando la importancia del atributo *inteligencia* por el de *movilidad*, ciertamente se ha analizado que crear agentes muy inteligentes suele ser demasiado costoso y en ocasiones poco fructífero, por otro lado crear agentes móviles es relativamente sencillo y para ciertas aplicaciones brindan grandes beneficios y mejoran en el desempeño de la aplicación.

La necesidad de una nueva tecnología que ayudará a resolver el tráfico en Internet así como lograr la independencia de tareas rutinarias en Internet ha hecho que en los últimos años haya habido una gran proliferación de sistemas de agentes móviles.

El concepto de movilidad cada día ha sido más aceptado y valorado al empezarse a comprender los beneficios de la tecnología, de igual manera otras tendencias como la venta de productos por Internet a favorecido la creación de diversos sistemas de software en donde no sólo se exige que los agentes deban ser móviles sino inteligentes y autónomos también.

A pesar de que los nuevos sistemas exijan todas las características deseables en un sistema de agentes, la *movilidad* cada día juega un papel más relevante por lo que muchas aplicaciones para el desarrollo de agentes crean nuevas versiones incluyendo movilidad o bien se desarrollan nuevas, entre ellas podríamos citar una larga lista de aplicaciones de agentes móviles que se encuentran hoy disponibles en el mercado.

La necesidad de una nueva tecnología que ayudará a resolver el tráfico en Internet así como lograr la independencia de tareas rutinarias en Internet ha hecho que en los últimos años haya habido una gran proliferación de sistemas de agentes móviles. Por ejemplo:

- Agent Tcl (Dartmouth College)
- Aglets Workbench (IBM)
- ARA - Agents for Remote Action (University of Kaiserslautern)
- Concordia (Mitsubishi Electric Information Technology Center America)
- ffTVIAIN - The Frankfurt Mobile Agents Infrastructure (Johann Wolfgang Goethe Universität)
- Java Agent Template (H. Robert Frost)
- JATLite (Stanford University)
- MOA - Mobile Objects and Agents (OpenGroup)
- Mole (Mole Team)
- Odyssey (General Magic)
- Voyager (ObjectSpace)

La mayor parte de estos sistemas de agentes móviles o frameworks están implementados en Java, no obstante prevalecen algunos que siguen siendo en lenguajes propietarios.

2.1.15 Agentes móviles

Es importante antes de profundizar en la tecnología que se de una definición detallada de agente móvil por lo que presentaré mi definición personal:

Un agente móvil es un programa con características especiales que presenta tres estado; un estado de datos (estructura de datos y otros objetos no agentes), un estado de código (las clases del agente y otras referencias a objetos) y un estado de ejecución (el control de procesos que se ejecutan en el agente).

Accede a un conjunto de métodos que le permiten moverse de un servidor a otro, siendo ésta su característica principal. El agente posee un programa de ejecución principal que define las tareas que realizará y el grado de inteligencia con el que actuará para interactuar con los usuarios y para resolver tareas adversas en su entorno. Finalmente es autónomo porque no necesita consultar a ningún supervisor humano para tomar una decisión y porque posee un hilo de ejecución propio e independiente del resto de los agentes.

Las características básicas de un agente móvil son:

- *Autonomía*: tienen su propio hilo y actúan por sí mismos.
- *Móvil*: serializable.
- *Concurrente*: puede ejecutarse con mas agentes a la vez.
- *Direccionable*: se puede definir su comportamiento.
- *Continuo*: se ejecutan continuamente y por tiempo indefinido.
- *Reactivo*: reacciona a su entorno mediante métodos.
- *Social*: interoperan con objetos y otros agentes.

2.1.16 El paradigma de Agentes móviles

Debido a la dificultad de la arquitectura actual de Internet para satisfacer el ritmo de crecimiento exponencial de sus los usuarios, era necesario una nueva aproximación que satisficiera dos necesidades aparentemente contradictorias: incrementar la sofisticación de los tipos posibles de comunicación sin restringir el ancho de banda disponible de los componentes de Internet. Una solución que satisface ambas necesidades son los agentes móviles.

Aproximación actual

El principio de organización central de las comunicaciones en las redes de ordenadores de hoy en día, que son las llamadas procedimientos remotos RPC (de la lengua inglesa *Remoto Procedure Call*), los cuales habilitan a un ordenador llamar procedimientos en otro (Véase figura 1).

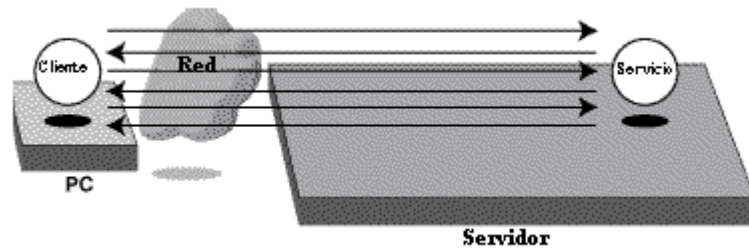


Figura 1 - Aproximación mediante el uso de RPC

Cada mensaje en la red transporta solicitudes o confirmaciones de procedimientos. Una solicitud incluye datos que son los argumentos del procedimiento y la respuesta incluye datos que son sus resultados. El procedimiento mismo es interno al ordenador que lo ejecuta. Y todos los mensajes viajan desde el punto origen al destino

Nueva aproximación

Una alternativa a las llamadas de procedimientos remotos es la promoción remota (obsérvese la Figura.2). Dos servidores que se comunican con el paradigma de programación remota hacen un acuerdo sobre las instrucciones que son permitidas en un procedimiento y los tipos de datos que son permitidos en su estado. Estos acuerdos constituyen un lenguaje. El lenguaje incluye instrucciones que permiten al procedimiento tomar decisiones, examinar, y modificar su estado, y llamar procedimientos proporcionados por el ordenador que se está recibiendo el programa. Tales llamadas a procedimientos son locales en vez de remotas. El procedimiento y su estado son llamados agente móvil para enfatizar que ellos representan al ordenador que los envía aunque ellos residen y operan en el ordenador que les recibe.

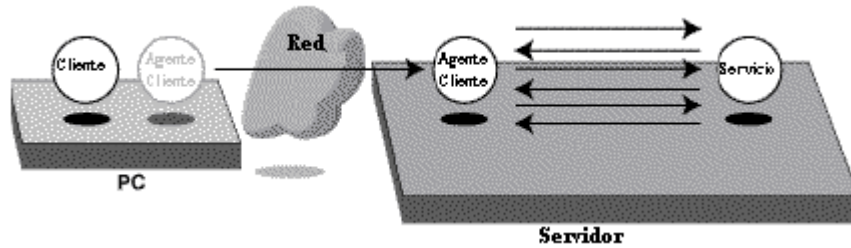


Figura 2 – Nueva Aproximación mediante programación móvil

Ventajas de la programación móvil

La ventaja táctica inmediata de la programación remota es el desempeño. Ya que será más rápido enviar un agente al servidor y directamente trabajar localmente en vez que remotamente.

Por otra parte la ventaja estratégica de la programación remota es la personalización. La programación remota cambia no sólo la división de tareas entre los fabricantes de software, sino también facilita la instalación de ese software. Los componentes del servidor de una aplicación basada en programación remota, son dinámicamente instalados por la misma aplicación que se está ejecutando desde el ordenador del usuario, desde que cada componente es un agente.

2.1.17 La tecnología de agentes móviles

Una tecnología para agentes móviles es una infraestructura de software que puede montarse encima de una gran variedad de computadoras y hardware de comunicaciones, presente y futuro. Esta tecnología, implementa los conceptos mencionados en la sección anterior y otros relacionados con ellos que les permite a los agentes interoperar.

La tecnología tiene tres componentes principales: el lenguaje en el cual los agentes y los lugares son programados; una máquina o intérprete para ese lenguaje; y los protocolos de comunicación que permite a esas máquinas residir en diferentes ordenadores para lograr el envío en intercambio de agentes.

El lenguaje

El lenguaje de programación de los creadores de aplicaciones de comunicación con agentes debe definir los algoritmos que los siguen los agentes y la información que llevan conforme viajan por Internet.

Para facilitar el desarrollo de las aplicaciones de comunicación, y la interacción entre lugares y agentes el lenguaje de programación debe ser:

- *Completo*: para que cualquier algoritmo pueda ser expresado en el lenguaje, transfiera la codificación de un agente de una a otra.
- *Orientado a objetos*: para obtener los beneficios de esta tecnología
- *Dinámico*: para que pueda transportar la clases que se requieran para crear instancias de agentes en máquinas remotas.
- *Persistente*: para que el agente y su información sean respaldados en un medio no volátil.
- *Portable y seguro*: para que se pueda ejecutar sobre cualquier plataforma de una forma segura.

- *Versátil*: que permita tareas complejas de transportación, autenticación y control de acceso.

La máquina o intérprete

La máquina es un programa de software que implementa el lenguaje para mantener y ejecutar los lugares dentro de su contexto, al igual que a los agentes que ocupan esos lugares. Al menos conceptualmente la máquina consta de tres interfaces de programación de aplicaciones (APIs), obsérvese la figura 3. Una API de almacenamiento que le da acceso a memoria no volátil que requiere para preservar a los lugares y agentes en caso de fallas. Una API de transporte que le da a la máquina acceso a los medios de comunicación que requiere para transportar agentes a y de otras máquinas. Y una API de aplicaciones externas que le permite a las partes escritas en el lenguaje de agentes interactuar con otras que no están escritas en dicho lenguaje.

Protocolos

Los protocolos habilitan a dos máquinas que se comuniquen. Las máquinas se comunican para transportar agentes. Los protocolos deben operar sobre una gran cantidad de redes de transporte incluyendo aquellas basadas en el protocolo de Internet TCP IP.

Los protocolos operan en dos niveles. El nivel más inferior gestiona el transporte de los agentes; y el más alto, su codificación y decodificación.

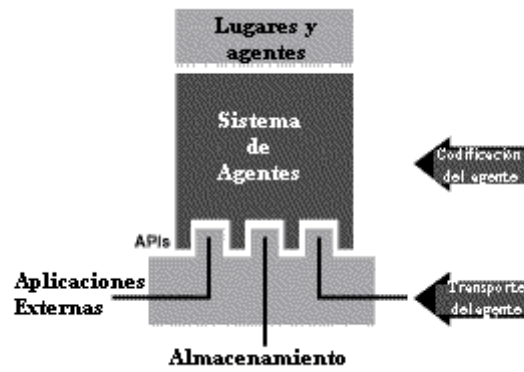


Figura 3 - La máquina y los niveles de los protocolos

El protocolo de interconexión de plataformas especifica como dos máquinas se autentifican otra usando por ejemplo, criptografía de clave pública y entonces transfiere la codificación de un agente a otra.

2.2. ESTADO DEL ARTE

2.2.1 Los Agentes Móviles y Otras Tecnologías

La tecnología de agentes móviles es una tecnología emergente que se han generado de la recopilación de las mejores características de otras tecnologías de programación distribuida. De aquí, que los agentes móviles son el resultado de objetos distribuidos, software intermedio orientado mensajes, applets, etcétera.

A continuación mencionaré brevemente algunas diferencias de los agentes móviles con algunas otras tecnologías de programación distribuida.

Objetos distribuidos, software intermedio orientado a mensajes y agentes móviles

Hoy en día, la mayoría de los programadores de Java están familiarizados con los objetos distribuidos y cómo éstos pueden ser usados para dividir un sistema a través de varias máquinas dentro de una red, como lo muestra la Figura 4. La invocación de métodos remotos (RMI) de Java y varias implementaciones de la especificación de CORBA están disponibles para los programadores de Java. Usando objetos distribuidos, el programador de Java puede invocar métodos de objetos que residen en otras máquinas tan fácilmente como invocar métodos en objetos locales.

Adicionalmente a los objetos distribuidos otros modelos de computación distribuida pueden ser utilizados para segmentar sistemas sobre una red. El software intermedio orientado a mensajes (MOM de la lengua inglesa *Messaging-oriented middleware*), actúa similarmente a la invocación de métodos remotos tradicionales, pero con una variación. MOM, consiste en un conjunto de mensajes que son gestionados por una o más "oficinas postales", las cuales vuelven a enviar estos mensajes a los grupos de interés específicos. Este modelo tiene la ventaja de que puede soportar sistemas con clientes desconectados temporalmente tales como computadoras portátiles.

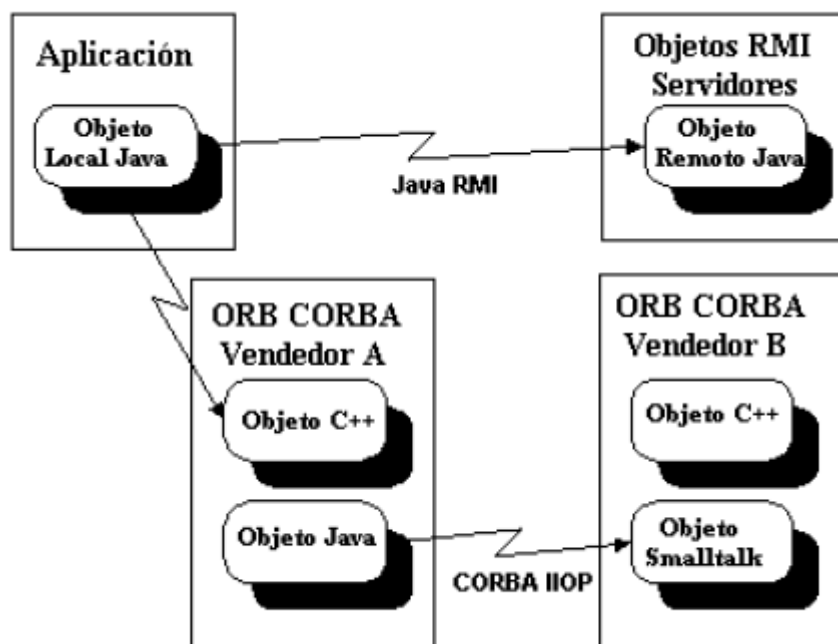


Figura 4 – Metodologías de Objetos Distribuidos

Los agentes móviles difieren de ambos de estos mecanismos de distribución. Con agentes móviles, el contenido ejecutable es trasladado a través de la red. Con relación a esto, un agente móvil puede ser

considerado como un objeto distribuido que se mueve. La facilidad común de agentes móviles está en el proceso de determinar cómo los agentes pueden ser introducidos al mundo de CORBA. Se espera que el trabajo de OMO así como los otros esquemas estándar, terminen en un conjunto de conceptos comunes, mecanismos de comunicación y protocolos para aplicaciones basadas en agentes.

El decidir qué mecanismo de distribución utilizar depende de la aplicación que intentemos desarrollar. Cada una ofrece ventajas sobre las otras para un problema y dominio específicos. No obstante, pueden fusionarse más de una de las tecnologías para lograr resolver un problema de mayor envergadura.

2.2.2 Comparación y tendencias de los sistemas de agentes móviles

El estado del arte de esta tecnología permite vislumbrar los avances que han tenido los sistemas de agentes, las necesidades que se han superado y retos que faltan por resolver de forma satisfactoria. De igual forma se observe una tendencia hacia los cuales se dirigen todos, con el afán de abarcar el amplio mundo de Internet.

Comparación

En términos generales podríamos decir que D'Agents sería apropiado sería apropiado para aplicaciones pequeñas debido a las limitaciones del lenguaje Tcl. Ara por su parte parece ser el indicado para todas las aplicaciones que requieran ser programadas en más de un lenguaje y con un desempeño aceptable. Finalmente los IBM aglets resultan ser los

mejores para grandes aplicaciones que se deseen usar en Internet, ya que tienen toda la funcionalidad que el lenguaje Java puede ofrecer.

Es difícil hacer una comparación de los sistemas de agentes presentados, ya que difieren en su arquitectura y plataformas de desarrollo, no obstante presentaremos una tabla comparativa de las características que son fundamentales en un sistema de agentes para poder visualizar los servicios que cada uno nos puede proporcionar, así como su grado de eficacia.

Es importante hacer mención que todas las características descritas son con las versiones públicas actuales de los productos hasta la fecha de publicación de este trabajo. Ya que algunos sistemas han sido modificados y están por sacar nuevas versiones que ofrecen más características.

Sistema de Agentes	Portabilidad	Seguridad	Trans del estado de ejecución	Lenguajes	Mensajes Remotos	Interfaz con el usuario	Velocidad
D'Agents	Unix	Buena	Si	Tcl, (Java, python) ²⁹	Si	Muy buena	Regular
Ara	Unix	Buena	Si	Tcl, C	No	Buena	Buena
Aglets	Todas	Regular	No	Java	Si	Buena	Muy buena

Tabla 2. Tabla de Principales Características de los Sistemas de Agentes

Aunque cada plataforma tiene sus ventajas y desventajas, se considera las características de Java y su ubicuidad le permite a los IBM Aglets

ofrecer una mayor funcionalidad, aunado a esto los fines comerciales de la plataforma han hecho de que exista todo un trabajo de equipo en IBM Japón que están trabajando en la creación de un entorno completo para el desarrollo de agentes móviles, que en un futuro se ofrecerá al mercado.

Por su parte OMG ha creado una infraestructura bien diseñada para interoperabilidad de Agentes Móviles que soporta la asignación de nombres, seguridad y localización que son indispensables para que los agentes naveguen por Internet, sin embargo al no estandarizar la interpretación y serialización de los agentes, la interoperabilidad en ocasiones se logrará parcialmente. Mas aún la MASIF no da ningún soporte para la comunicación mediante mensajes lo que podría ser una alternativa para la interoperabilidad de sistemas de agentes que no fueran del mismo tipo.

Tendencias

La tecnología de agentes móviles tomó gran fuerza desde la aparición de Java, este lenguaje proporcionaba portabilidad y muchas características que lo hacían ideal para el desarrollo de sistemas de agentes. Desde entonces han aparecido una gran cantidad de sistemas de agentes móviles basados en Java.

Los sistemas que ya existían anteriormente como D'Agents y Ara están modificando su núcleo para soportar a Java como uno más de los lenguajes en los que se puedan programar los agentes.

Los sistemas de agentes que logren una mayor interoperabilidad en la MASIF serán aquellos que estén implementados en Java, por las facilidades de interacción que brinda el lenguaje. Actualmente no ha habido ninguna implementación de un ORB que siga completa y adecuadamente la especificación de la MASIF para dar soporte a la movilidad.

Desde que Java está siendo usado en todas partes será más fácil utilizar un sistema de agentes móviles basado en Java que un ORB que implemente la especificación de la MASIF, Java tiene pues no sólo la ventaja de la ubicuidad/portabilidad sino también la ventaja de tiempo, está claro Java será la tendencia dominante para los sistemas de Agentes Móviles en los próximos meses.

2.2.3. Aplicaciones de los Agentes Móviles

- **Recolección de datos de distintos sitios:** una de las mayores diferencias entre el código móvil, tales como los applets, y los agentes móviles es el itinerario. Mientras que el código móvil usualmente viaja sólo de un punto a otro, los agentes móviles tienen un itinerario y pueden viajar secuencialmente a muchos servidores. De aquí que una aplicación natural de los agentes móviles, es la recolección de información a través de muchas computadoras enlazadas a una red.

- **Búsqueda y filtrado:** dado el constante incremento en la cantidad de información disponible en Internet y en otras redes, la actividad de recolectar información de una red implica la búsqueda entre grandes

cantidades datos de unas cuantas piezas relevantes información. Eliminar la información irrelevante puede ser un proceso que consume mucho tiempo y quizá frustrante. En nombre de un usuario, un agente móvil puede visitar muchos servidores, buscar a través de la información disponible en cada servidor, y construir un índice de enlaces a las piezas de información que concuerdan con el criterio de búsqueda. El filtrado y la búsqueda muestran un atributo común a muchas aplicaciones potenciales de agentes móviles: el conocimiento de las preferencias del usuario.

- **Monitorización:** en algunas ocasiones la información no está distribuida a través de un espacio como puede ser un conjunto de ordenadores, sino a través del tiempo. Nueva información constantemente está siendo generada y publicada en la red. Los agentes pueden ser enviados para esperar por ciertas clases información hasta que ésta sea generada o se encuentre disponible.

Esta aplicación resalta la naturaleza asíncrona de los agentes móviles. Si se envía un agente, no es necesario sentarse y esperar por resultados de la información recopilada. Se puede programar un agente para que espere todo lo que sea necesario hasta que cierta información se encuentre disponible. También no se requiere permanecer conectado a la red hasta que el agente regrese. Un agente puede esperar hasta que su usuario se reconecte a la red antes de realizar un informe para él.

- **Comercio electrónico:** El comercio electrónico es otra aplicación apropiada para el uso de la tecnología de agentes móviles. Un agente móvil podría realizar compras, incluyendo la realización de órdenes de compra y potencialmente pagar. Por ejemplo, si se quiere volar de un sitio a otro, un agente podría visitar las bases datos de los horarios de vuelo y los precios de varias líneas aéreas, encontrar el mejor precio y horario de salida, hacer la reserva e inclusive pagar con un número de tarjeta de crédito.

El comercio electrónico también puede llevarse a cabo entre agentes. Por ejemplo podría haber un servidor de agentes dedicado a la compra y venta de automóviles. Si se desea comprar un coche, se le podría dar a un agente un conocimiento que especifique las preferencias del usuario, incluyendo el rango de precios y una potencial estrategia de negociación. Entonces se enviaría al agente al servidor dedicado, donde el se reuniría con otros agentes que están buscando vender un coche.

- **Procesamiento paralelo:** dado que los agentes móviles pueden moverse de un modo u otro y pueden reproducirse, un uso potencial de la tecnología de agentes móviles es una forma de administrar aplicaciones que requieran procesos en paralelo. Ciertamente prácticamente todas las aplicaciones descritas en estas líneas pueden hacer uso de la concurrencia para mejorar enormemente su eficiencia y lograr de esta manera aplicaciones más rápidas.

- **Distribución masiva información:** otro uso potencial de los agentes móviles es la distribución interactiva de noticias o publicidad a grupos de

interés. Los agentes móviles pueden usarse en este caso al igual que el correo electrónico para distribuir indiscriminadamente información si se trata de publicidad, o bien especificarle al agente el tipo de servidor en donde habrá de distribuir dicha información en el caso de ser información mas privada.

- **Interacción y negociación:** además de la búsqueda en bases datos y ficheros, los agentes pueden obtener información mediante la interacción con otros agentes. Si por ejemplo, se desea realizar una agenda de reuniones con otras personas, se puede enviar a un agente móvil para que interaccione con agentes representantes de cada una de las personas que se desea en invitar a la reunión. Los agentes pueden negociar para establecer el horario de la reunión. En este caso, cada agente contiene información acerca de la agenda de su usuario. Para lograr el acuerdo del tiempo establecido para la reunión, los agentes necesariamente necesitan intercambiar información.

- **Supercomputadora virtual (cálculos en multiprocesos):** los cálculos complejos con frecuencia pueden ser descompuestos en unidades discretas para la distribución en una pila de servidores o procesos. Cada una de estas unidades discretas puede ser asignada a un agente, el cual es entonces enviado a un sitio remoto en donde el trabajo es realizado. Una vez terminado, cada agente puede regresar a casa con los resultados los cuales pueden ser agregados y sumados.

- **Redes parcialmente desconectadas:** en sistemas con redes frecuentemente desconectadas, los agentes pueden salvar los procesos

de la red al mover el contenido ejecutable hacia los datos origen en vez de intentar constantemente conexiones de red a los datos origen.

- **Entretenimiento:** un último ejemplo de las aplicaciones potenciales para los agentes móviles es el entretenimiento. En este escenario, los agentes representan a los jugadores del juego. Los agentes compiten con otros en representación de los jugadores. Cada jugador programada un agente con una estrategia, entonces enviará al agente al servidor del juego. Esto facilita que los jugadores sean remotos, y si el servidor del juego se ejecutara en un ordenador en Las Vegas, entonces quizá los agentes podrían jugar con dinero real.

- . **Entrega de correo inteligentemente:** los agentes podrían entregar un correo electrónico basándose en el contenido del mensaje. Para lograrlo el agente tendría que llevar en su código las reglas de decisión.

El interés en agentes móviles es motivado por los beneficios que proveen para la creación de sistemas distribuidos. Podemos considerar que los beneficios fundamentales que ofrece la tecnología de agentes móviles son: la reducción del tráfico en la red, la capacidad de desarrollar aplicaciones autónomas y la facilidad que ofrecen para crear aplicaciones que se ejecuten en paralelo y sean más rápidas.

2.2.4. Ventajas de los Agentes Móviles

1. **Ellos reducen el tráfico de la red:** los sistemas distribuidos con frecuencia se basan en los protocolos de comunicación que involucran múltiples interacciones para lograr una tarea dada. Esto es

especialmente verdad cuando las medidas de seguridad son habilitadas. El resultado es una gran cantidad de tráfico en la red. Los agentes móviles nos permiten empacar una conversación y enviarla a un servidor destino donde las interacciones pueden tomar lugar localmente. Los agentes móviles son también útiles cuando vienen a reducir el flujo de registros de datos en las redes. Cuando grandes cantidades de volúmenes de datos son almacenadas en servidores remotos, estos datos deberían de ser procesador en la localidad de los datos, en vez de transferirlos a través de la red. El fin es simple: mover la computación a los datos en vez de los datos a la computación. Aunado a esto debemos mencionar que en los últimos años ha habido un crecimiento exponencial del número de usuarios en Internet (ver figura 5), los agentes móviles de igual forma contribuirán a reducir el número de interacciones dentro de la red mundial, lo que ayudará a que sea más rápida.

Evolución de Internet

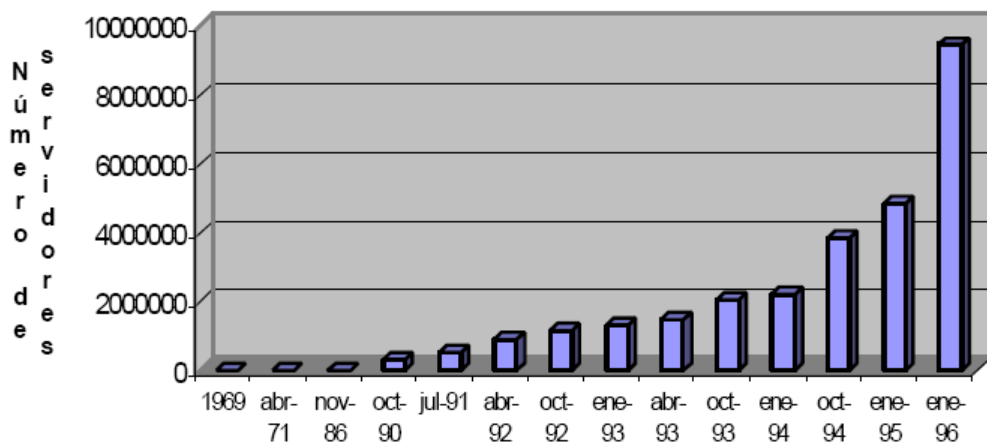


Figura 5 - Incremento del número de usuarios en Internet

2. Ellos superan el estado latente de las redes: los sistemas críticos de tiempo real tales como robots en procesos de manufactura necesitan responder a cambios en su entorno en tiempo real. Controlando tales sistemas a través de una fábrica en red de un tamaño sustancial implica estados latentes significantes. Para sistemas de tiempo real críticos, tales estados latentes no son aceptados. Los agentes móviles ofrecen una solución, desde que ellos pueden ser despachados de un controlador central para actuar localmente y ejecutar directamente las direcciones del controlador.

3. Ellos encapsulan protocolos: cuando los datos son intercambiados en sistemas distribuidos, cada servidor posee el código que implementa los protocolos requeridos para codificar propiamente los datos de salida e interpretar los datos de entrada respectivamente. Sin embargo, como los protocolos evolucionan para acondicionar nuevas eficiencias o requerimientos de seguridad, es muy difícil sino es que imposible tarea la de actualizar el código del protocolo apropiadamente. El resultado es frecuentemente que los protocolos llegan a ser un problema de herencia. Los agentes móviles, por otro lado, son capaces de moverse a servidores remotos con el fin de establecer canales basados en protocolos propietarios.

4. Ellos se ejecutan de forma asíncrona y autónoma: frecuentemente los dispositivos móviles tienen que depender de conexiones de red caras o frágiles. Esto es, tareas que requieren una conexión abierta continuamente entre un dispositivo móvil y una red no serán en la

mayoría de los casos económica o técnicamente factible. Las tareas pueden ser incrustadas en agentes móviles, los cuales pueden ser despachados

5. Ellos se adaptan dinámicamente: Los agentes móviles tienen la habilidad de percibir su entorno y reaccionar de forma autónoma a cambios. Un conjunto de agentes móviles posee la habilidad única de distribuirse a sí mismos entre los servidores en una red de tal forma que mantiene la configuración óptima para la solución de un problema particular.

6. Ellos son heterogéneos por naturaleza: la computación en redes es fundamentalmente heterogénea, con frecuencia desde las perspectivas de hardware y software. Como los agentes móviles son generalmente independientes del ordenador y de la capa de transporte, y dependientes sólo de su entorno de ejecución, ellos proporcionar condiciones óptimas para una integración del sistema transparente.

7. Ellos son robustos y tolerables a fallas: la habilidad de los agentes móviles para reaccionar dinámicamente a situaciones desfavorables y eventos hace más fácil el construir sistemas distribuidos robustos y tolerables a fallas. Si un servidor está siendo apagado, todos los agentes que se ejecutan en ese servidor serán avisados y se les dará tiempo suficiente para que se envíen a sí mismos y continúen sus operaciones en otro servidor de la red.

2.2.5. Desventajas de los Agentes Móviles

Podemos considerar que los sistemas de agentes han superado en gran medida la primera necesidad que debe satisfacer la tecnología ‘Portabilidad’, la llegada de Java ha contribuido en gran medida a solucionar dicho problema, sin embargo quedan muchos requerimientos por satisfacer para que la tecnología pueda proveer un entorno completo de desarrollo de aplicaciones. A continuación algunas de las carencias más importantes que los sistemas de agentes suelen tener.

Seguridad

La seguridad, factor fundamental para que la tecnología de agentes móviles sea aceptada en Internet no ha sido cubierta satisfactoriamente. D’Agents no proporciona un mecanismo en el que se puedan asignar privilegios de forma específica a un agente. Ara tiene un mecanismo de autenticación poco eficiente y los Aglets solo son capaces de restringir el acceso de los agentes generalizando en dos grupos en confiables y en no confiables.

Los sistemas de agentes móviles deberán introducir nuevos métodos de seguridad que garanticen una completa integridad de la información. De igual forma estos métodos deberán poder personalizar para que cada usuario pueda configurarlo de una forma que satisfaga sus necesidades.

Actualmente los sistemas carecen de métodos de autenticación con firmas digitales, tampoco utilizan la criptografía para transferir la información de forma encriptada. Finalmente carecen de bases de datos

en donde se puedan dar privilegios específicos a un agente para acceder a los recursos del sistema.

Interoperabilidad

La MASIF logra una interoperabilidad parcial de los sistemas de agentes, es decir si los sistemas de agentes son del mismo tipo se puede lograr una buena interoperabilidad, pero si por el contrario son de tipo diferente la interoperabilidad es casi nula.

A pesar de la existencia de la MASIF, ésta solo aplica para el mundo CORBA, por lo que sí agentes móviles del sistema de los Aglets quisieran interactuar con agentes móviles de otros sistemas no podrían hacerlo, el único medio de interacción sería la implementación de las interfaces que especifica la MASIF de CORBA, lo cual implica una ardua labor.

La aparición de Java produjo la aparición de una gran cantidad de sistemas de agentes móviles basados en Java, con la rapidez que evoluciona esta tecnología, en poco tiempo

Entornos visuales de desarrollo e interfaces gráficas del usuario

Actualmente los sistemas de agentes móviles no cuentan con un entorno visual de desarrollo, lo que exige a los desarrolladores tener un nivel de conocimientos (de programación) informáticos avanzado.

Si observamos la mayor parte de las aplicaciones de los agentes móviles son enfocadas a la búsqueda y recopilados de datos en redes,

dichas tareas son requeridas por todo tipo de usuarios y por tanto cualquier tipo de usuario debe ser capaz de crearlos. Por tanto se requiere un tipo de desarrollador de aplicaciones que todo usuario pueda utilizar. De igual forma los agentes deben ser capaces de utilizar interfaces gráficas que les permita interactuar con los usuarios de una forma más sencilla, para que estos puedan seleccionar el tipo de tarea e información que requieran.

Incorporación a los navegadores

Uno de los objetivos principales de los agentes móviles es la búsqueda y recopilación de datos en Internet, sin embargo actualmente no se encuentran incorporados a los navegadores lo que facilitaría la creación de búsquedas asíncronas por parte de los usuarios de Internet.

2.2.6. Caso de Exito de las Tecnologías de Solución Existentes

2.2.6.1 Middleware

Filtración rápida, Flexible con Phlat - Fácil Búsqueda Personal y Organización ¹²

RESUMEN

Los sistemas para búsqueda rápida de información personal rápidamente se hacen ubicuos. Tales sistemas prometen radicalmente mejorar la dirección personal de la información, aún la mayoría son modelados sobre la búsqueda de Web en la cual los usuarios conocen muy poco sobre el contenido que ellos buscan. Describimos el diseño y el despliegue de un sistema llamado Phlat que optimiza la búsqueda de la información personal con un interfaz intuitivo que combina la búsqueda y navegación a través de una variedad de señales asociativas y contextuales. Además, Phlat apoya un esquema de marcación unificada (etiqueta) para organizar el contenido personal a través de sistemas de almacenaje (archivos, correo electrónico, etc.).

El sistema ha sido desplegado a cientos de empleados dentro de nuestra organización.

Introducción

Los motores de Búsqueda son un instrumento popular y ubicuo para encontrar información, sobre todo en la Web. Con unas pulsaciones, la información deseada aparece como por arte de magia desde miles de millones páginas Web. Ahora, los instrumentos de búsqueda están comenzando a disfrutar del uso extendido para encontrar la información personal. De verdad, un número de sistemas para búsqueda personal

¹² Tomado de la referencia [15]

están disponibles en todos los ordenadores personales de hoy. Sistemas de Microsoft (desktop.msn.com), Yahoo (desktop.yahoo.com, anteriormente X1), Google (desktop.google.com), Copernic (copernic.com) y otros están disponibles para descarga gratuita, y esta búsqueda de funcionalidad está siendo construida en la generación más reciente de sistemas operativos para ordenadores personales (p.ej., el Proyector de Apple para el OSX y el Vista OS de Microsoft). Sin embargo, buscar información personal es diferente de buscar sobre el Web en un número de formas. Quizás la diferencia más grande es que la información personal es, también, muy personal. La gente es familiar con una multitud de detalles y características sobre su información, así como los contextos que rodean su empleo. Por consiguiente, buscar información personal puede ser mucho más rica que una simple búsqueda de una palabra clave de contenido. Buscar información personal puede influenciar toda clase de detalles asociados con el contenido. El desafío yace en la creación de un interfaz de usuario que explote los detalles amplios y variados y conexiones entre fuentes de la información que los usuarios pueden recordar, manteniendo la simplicidad de búsqueda de palabra clave que hace la búsqueda Web tan poderosa y fácil.

Creamos Phlat (ver Figura 6) para explorar este desafío de UI. Phlat combina la palabra clave y la búsqueda de valor de característica en una manera sin costuras e intuitiva, permitiendo a usuarios encontrar información basada en lo que ellos pueden recordar, en cualquier parte donde pueda ser almacenada. Además, Phlat proporciona una

instalación para el contenido de marcación con una metadata creado por usuario para asegurar que los usuarios pueden volver a su contenido.

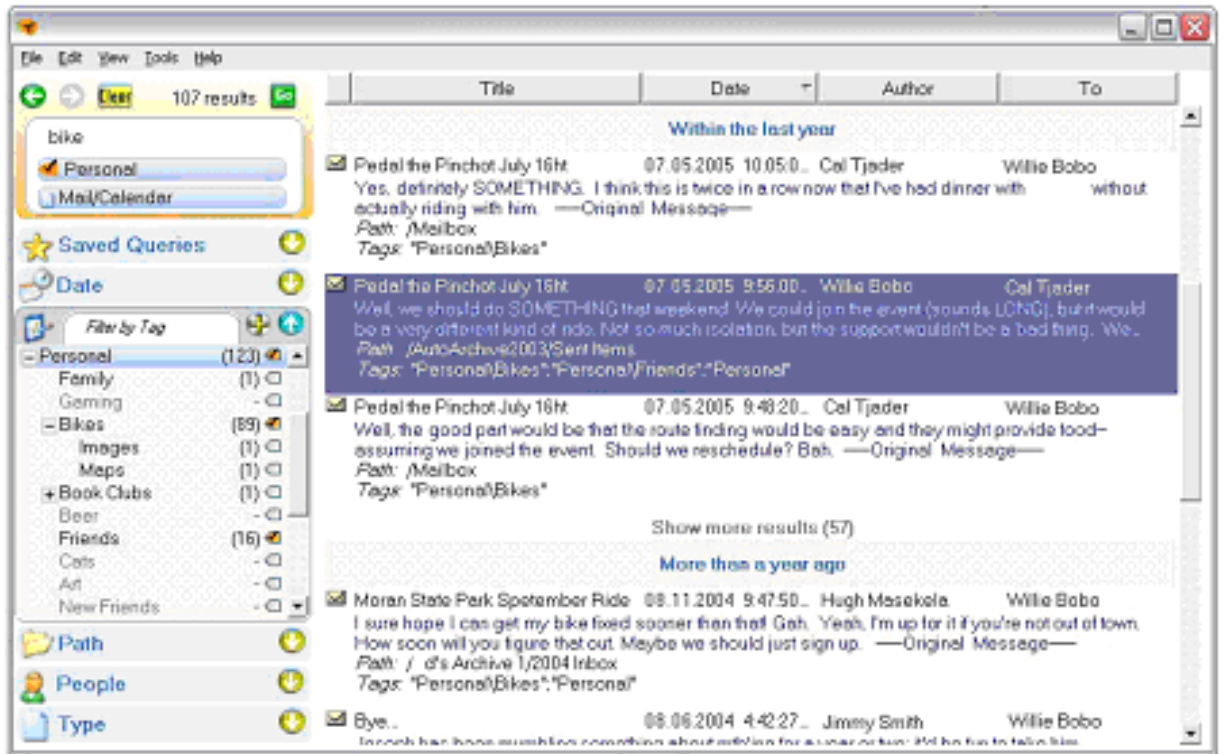


Figura 6: La interface Phlat

Los sistemas de búsqueda personales de escritorio disponibles hoy son notablemente poderosos. Ellos unifican un número de sitios posibles de información (p.ej., volúmenes de sistema de fichas, correo electrónico, historias de Web, etc.), tal que una búsqueda de un interfaz solo puede atravesar muchos tipos de posiciones e información. Además, estos sistemas son capaces de indexar un rango de metadata estructurado más allá del contenido de artículos almacenados, incluyendo fechas, autores, bitrates, etc. Phlat fue creado para influir estos rasgos

poderosos de sistemas de búsqueda de modo que los usuarios puedan confiada e intuitivamente encontrar su contenido.

TRABAJO RELACIONADO

Desde que Vannevar Bush primero describió el memex [2], " el dispositivo ... en el cual un individuo almacena todos sus libros, registros, y comunicaciones, y que pueden consultar ... con excedente velocidad y flexibilidad, " investigadores informáticos y diseñadores de sistema han trabajado para hacer su visión manifiesta.

Creemos que los materiales para la visión de Bush están disponibles ahora, pero todavía hay mucho por hacer. El volumen de información personal crece exponencialmente mientras nuestros cerebros permanecen iguales. La Dirección de Información personal (PIM) es un área rápidamente creciente de investigación preocupada en como la gente maneja esta avalancha de información.

En 1983, Malone observó trabajadores en oficinas organizar el material de papel y luego sugirieron como los sistemas informáticos podrían apoyar actividades similares para el contenido electrónico.

Después de esto, Lansdale describió muchos de los desafíos psicológicos asociados con PIM. Él caracterizó dos problemas claves que cualquier instrumento PIM debe dirigir: *Primero, categorizar artículos es cognoscitivamente difícil*. La mayor parte de la dificultad asociada con la información directiva personal hallada y manejada se deriva de actuar recíprocamente con sistemas de clasificación y estructuras de categoría. Estos problemas ocurren primero en la fijación de estructuras

apropiadas y decidir como archivar el contenido; y luego, cuando es tiempo de recuperar la información, decidir como podría haber sido clasificado. *Segundo, la gente recuerda mucho más sobre artículos que puede ser usado según procedimientos de recuperación.* Si el contenido en memorias de la gente pudiera ser usado por sistemas de recuperación, nosotros realmente tendríamos "una memoria externa".

La mayor parte de la investigación en PIM se ha enfocado en la primera de estas cuestiones: hacer la categorización más fácil, más útil y menos dolorosa. Al contrario hubo menos esfuerzo aplicado al segundo de los desafíos de Lansdale: permitiendo a sistemas de recuperación utilizar lo que la gente recuerda sobre su contenido. Lifestreams permite la búsqueda del contenido personal con un énfasis particular en tiempo y la naturaleza visual de documentos electrónicos.

Las memorias Landmarks también se enfocaron en el tiempo, pero con especial atención en acontecimientos personales y públicos. MyLifeBits y las Cosas que he Visto (SIS) ambos proporcionan búsqueda de contenido y filtran para varias propiedades metadata, pero estos sistemas están más enfocados en la infraestructura de sistema y la funcionalidad que el interfaz de usuario. Aunque no está enfocado en la información personal, Flamenco explora UI para una búsqueda flexible sobre una variedad de propiedades ortogonales (facetadas) para proporcionar la exploración rápida y iteración de duda basada en los intereses del usuario. Finalmente, varios de los productos comerciales notaron más temprano proporcionan la búsqueda y filtración sobre las propiedades que van bien más allá de la simple palabra clave para la

búsqueda del contenido. Sin embargo, creemos que mucho más puede ser hecho para realmente encontrar el segundo desafío de Lansdale.

La memoria humana es sumamente dependiente de la relación entre el contenido codificado y las señales y el contexto asociado con aquel contenido. Esto quiere decir que la memoria de los usuarios para la información es sumamente dependiente de la mayor parte de otra información no directamente asociada con lo que ellos buscan. Czerwinski y Horvitz demostraron que la gente olvidaron mucha de sus tareas de computadora después de solamente un mes. Pero cuando fueron incitados por vídeos y fotos de su trabajo durante aquel tiempo, ellos recordaron muchos detalles sobre lo que ellos habían estado haciendo. Instrumentos de dirección centrados por tarea como UMEA y TaskMaster direccionan esto a algún grado capturando el contexto de tarea de contenido. Pero a menudo los hilos asociativos que conectan los pedazos de información son imposibles para un sistema deducir y probablemente sólo tiene sentido para el propio usuario. Algo que hace estas asociaciones más fáciles para explorar hará la información más fácil para encontrar.

Creamos Phlat con el deseo de construir y desplegar un interfaz que haría el encuentro de la información personal fácil e intuitiva sin importar que los usuarios puedan recordar sobre su contenido. Creemos que la búsqueda personal será cada bit tan importante en la evolución de PIM como la búsqueda de Internet ha sido para el World Wide Web.

LOS PRINCIPIOS DE DISEÑO

Nosotros acercamos el diseño de Phlat con varios principios claves. Ante todo quisimos diseñar un interfaz que se estire la idea de qué es "la búsqueda". Creemos que la dicotomía de búsqueda y navegador es una distinción artificial. En Phlat quisimos facilitar toda la información buscando como texto libre y/o la búsqueda de característica estructurada. En vez de ver la búsqueda y el navegador como comportamientos separados, Phlat los trata como dos terminaciones de un continuo liso. Además, mientras creemos que un buen sistema de búsqueda radicalmente reducirá la necesidad de estructuras de organización como carpetas, hay momentos claros cuando los usuarios tienen que aplicar etiquetas externas a su contenido. Para apoyar esta necesidad, quisimos diseñar un mecanismo conveniente e intuitivo para asignar y filtrar un metadata generado por el usuario (etiquetas). Sentimos que los puntos de diseño siguientes eran críticos para alcanzar estos objetivos:

1) Unificar la entrada de texto y la filtración. Si no hay nada más, el enorme éxito de búsqueda de Web ha demostrado que la gente está muy cómoda buscando una amplia variedad de información con solamente unas palabras de texto. Phlat debe animar la entrada de texto libre y la iteración como una parte principal del interfaz. Pero además, queremos aclarar que un filtro es una pregunta. Por lo tanto un usuario debería ser capaz de iniciar una pregunta aplicando un filtro, escribiendo a máquina el texto, o ambos, en cualquier orden y la representación de

pregunta deberían reflejar esta unidad de texto libre y la búsqueda estructurada.

2) Criterios corrientes de búsqueda deben ser visibles y salientes en cualquier momento. En cualquier momento, un usuario debe ser capaz de echar un vistazo en el interfaz y saber exactamente cuales son los criterios corrientes de búsqueda, lo que los filtros están colocados, que términos de búsqueda han sido entrados y cual clase de orden es.

3) Proporcionar la iteración de pregunta rápida. Cuando un usuario presiona Enter, hace click en un botón con filtro o de otra manera cambia la pregunta corriente, los resultados deben aparecer rápidamente y el cambio debe ser obvio. Tomamos una señal de los interfaces de pregunta dinámicos de Shneiderman, que proveen las actualizaciones continuas visuales de los resultados se ponen basado en la manipulación de atributos de datos. Vistas anticipadas de pregunta los datos de pareja se cuentan con atributos para mejorar las posibilidades que una manipulación de pregunta (p.ej., un filtro) retornará un juego útil de resultados - ni demasiado enorme, ni demasiado pequeño. Queremos que usuarios sean capaces de explorar su información personal en una manera natural y fluida.

4) Permitir a la iteración basada en el reconocimiento. Como la memoria de reconocimiento es generalmente mucho más robusta que la memoria recuperada, Phlat debería explotar este hecho para ventaja de sus

usuarios. Cuando un usuario ve un juego de resultados y propiedades, él puede ver algo que lo señala a su objetivo de la información. Phlat debería permitirle usar cualquier información que él vea más allá de su búsqueda.

5) Permitir la abstracción a través de valores apropiados. Como Phlat está diseñado para buscar a través de una amplia gama de tipos y fuentes, es importante apoyar un número de abstracciones apropiadas para emparejar los modelos cognoscitivos de nuestros usuarios. Un usuario no debería recordar si la imagen que él busca es tras un bmp, jpg, png, etc... Él necesita sólo hacer click sobre el filtro "Imagen" y Phlat preguntará para todos ellos.

6) La Etiqueta UI debe apoyar tanto marcación como filtración. Los usuarios no quieren tener UI separado para el uso de etiquetas y filtración. Debería haber un lugar que los usuarios van a actuar recíprocamente con etiquetas y esto debería apoyar ambas filtración (para encontrar cosas etiquetadas por una etiqueta) y etiquetas de aplicación a la información personal.

7) Integrar con el sistema de fichas común / operaciones de correo.

Como esperamos que usuarios empleen Phlat para actuar recíprocamente con su contenido a través de una variedad de fuentes, los usuarios deberían ser capaces de aplicar operaciones comunes

como el corte, la copia, la pasta, la rastra y la gota, la respuesta, etc. desde dentro el interfaz.

PHLAT

Arquitectura General

La aplicación Phlat fue diseñado como una cáscara para actuar recíprocamente con la información personal sobre la máquina de cliente. Phlat está escrito en Microsoft C Visual y usa el índice de Búsqueda de Windows Desktop y el motor de búsqueda. El Windows DS es una aplicación gratuita asociado con la Barra de tareas MSN. Para los detalles de la arquitectura de sistema de Windows DS.

Phlat proporciona acceso para contentar en el índice de Windows DS.

Por defecto, Windows DS indexa el correo electrónico del usuario y expedientes personales incluyendo medios de comunicación. Otras fuentes pueden ser añadidas, incluyendo el caché de Web y directorios de red compartidas.

PHLAT además, permite a usuarios añadir metadata creado por el usuario jerárquicamente, o etiquetas al contenido mostrado. Aplicando etiquetas después de una búsqueda no es claramente el único (o mejor) el tiempo para proporcionar esta funcionalidad, sentimos esto era un primer paso razonable para explorar la marcación UI.

Consideramos dos caminos para poner en práctica etiquetas: nosotros podríamos o poner en práctica una base de datos separada de etiquetas o aplicar las etiquetas directamente a artículos almacenados. Escogimos el éste para varios motivos: Primero, esto permite a un sistema de

marcación solo para atravesar múltiples fuentes (el correo electrónico, citas, archivos en posiciones diferentes físicas o lógicas). Segundo, Windows DS es capaz de directamente incluir en un índice las etiquetas, permitiendo a Phlat tratar preguntas para etiquetas justo como cualquier otra pregunta de característica. Tercero, porque las etiquetas directamente son asociadas con los archivos, el trabajo de marcación es mucho más seguro. Si algo pasa al índice, el metadata todavía es seguramente asociado con los artículos. Cuarto, los artículos tienen que sólo ser etiquetados una vez: cuando copio un archivo a otro sistema, los viajes de etiqueta con ello. Phlat pone en práctica la marcación insertando la información en un metadata del cliente apropiadamente asociada con el artículo objetivo. Para artículos de correo electrónico, esto es una característica MAPI y para archivos (y artículos de caché de Web), esto es una característica NTFS. Una limitación implícita por esta arquitectura es que la marcación para el correo sólo trabaja para el correo electrónico en Microsoft Outlook/Exchange que apoya MAPI. La marcación para archivos sólo trabaja sobre sistemas que apoyan el sistema de fichas NTFS.

Interfaz de Usuario

El interfaz de usuario Phlat es compuesto de 3 áreas principales (ver la Figura 7). En la esquina superior izquierda está el Área de Pregunta, comprendiendo mandos de pregunta (p.ej., Vaya, detrás, Adelante, Limpie la Parada de/) y una caja de pregunta para entrar y mostrar preguntas.

Debajo del área de pregunta está el Área Con filtro que comprende un juego de botones para varias propiedades ortogonales. Finalmente, a la derecha del Area de Pregunta y Áreas Con filtro está el Área de Resultados.

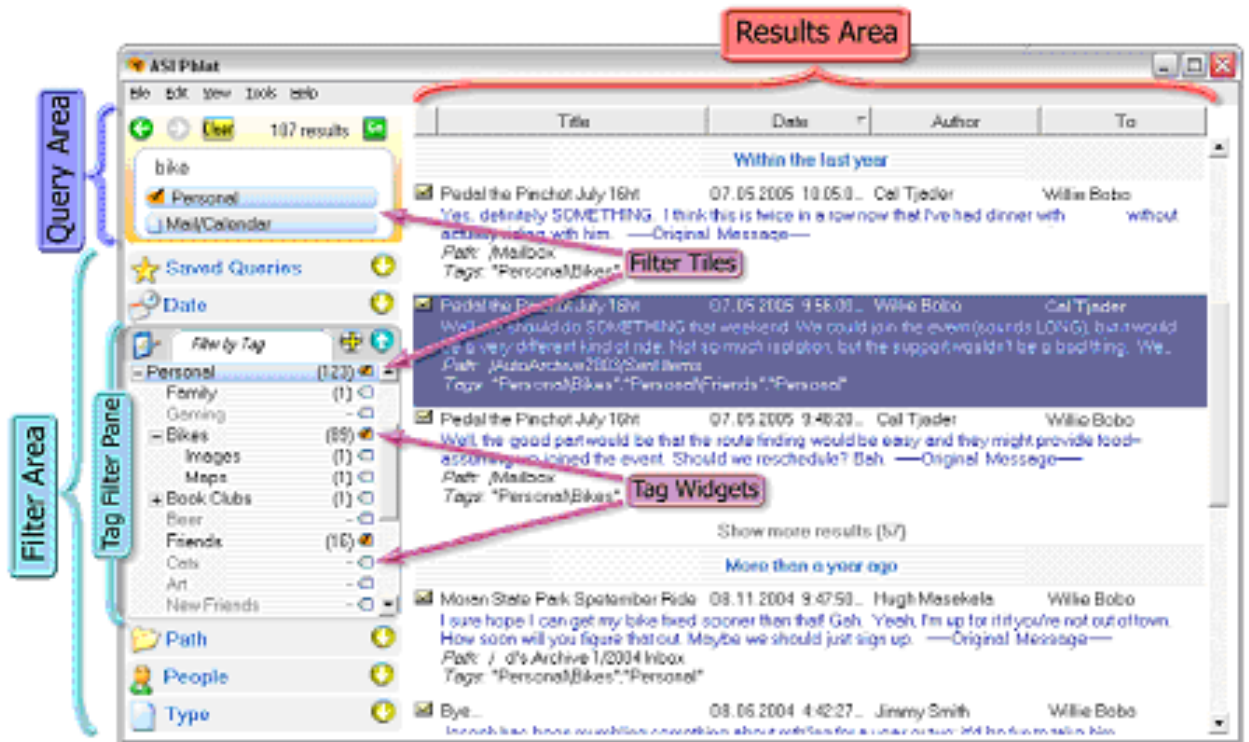


Figura 7: La interfase Phlat con un query de una clave simple y dos filtros

2.2.6.2 Web Services

Vida Transitoria: Colectando y compartiendo información personal¹³

Millones de personas envían información personal a través de Internet, con toda la información real varía grandemente. Algunas partes son

¹³ Tomado de la referencia 16.

extremadamente breves, otras son altamente detalladas. Algunos enfocan los cambios de acuerdo al estado de uno y sus pensamientos, otros describen rasgos estables y duraderos. Para manejar esta diversidad, creamos *Vida Transitoria*: un sistema que permite a la persona recopilar la información “transitoria”, compartir esta información recogida con otros. Vida Transitoria está diseñada como sidebar modular situada en la periferia de la pantalla. Una persona utiliza sus módulos para: actualizar el estado personal momentáneo (sentimientos, localización, sucesos, y pensamientos), grabar partes de la actividad hechas sobre el día así como también una lista “para hacer” de cosas dejadas para hacer, coleccionar URLs interesantes y fotos vistas, y componer ensayos de texto de cualquier cosa que haya capturado su interés. Una persona puede enviar selectivamente esta información como “mensaje de hoy” a su comunidad, y el ensayo a su blog personal. La información se guarda en un calendario histórico, que permite que uno vea la información registrada en el pasado.

INTRODUCCION

El estudio y el desarrollo de los sistemas que enfocan la colección y el almacenaje de información personal es actualmente un esfuerzo popular. Como las tecnologías digitales se hacen ubicuas, nuestras capacidades y deseos para registrar e intercambiar experiencias personales e información han aumentado con el tiempo. Los boletines de noticias del correo electrónico, la foto en línea que se comparte, páginas web personales, blogs, grupos de encuentro de planificadores

son solamente algunas de las cosas que contienen la información digital sobre la vida diaria de la gente.

La idea de grabación de la información personal apareció primero en 1945, donde Vannevar Bush propuso la idea de mecanizar el almacenaje de la memoria personal de información llamada un "Memex" (Bush, 1945). Sin embargo, su visión que el personal de construcción principalmente afectado arrastra a través de la información organizacional "exactamente como si los artículos físicos hubieran sido recogidos de fuentes extensamente separadas y reunidos juntos para formar un nuevo libro " (Bush, 1945). Esta visión estaba basada en el almacenaje y aún no se incorporaba compartir o la publicación a otros. Ahora, varios investigadores miran la mecánica de ahorro de todo lo que encontramos en una tienda de memoria digital, su organización, y sus enlaces para compartir (Bell, 2001; Czerwinski et al., 2006; Gemmell, Bell, y Lueder, 2006).

Nuestros intereses particulares están en la recolección de pequeñas piezas de información digitalizada que la gente tiende a compartir con otros en una base diaria y facilitar la publicación de tal información. Los puntos de interés para nosotros son los artículos de captura espontánea debido al interés personal y la información pertinente al mismo. Estos son acontecimientos diarios y la información contextual compartida con otros que tienen un interés a nuestras vidas.



Figura 8: Cuatro mecanismos que presenta la información personal.
 A. Perfiles B. Mensajes de hoy C. Blogs D. Nombres IM

LAS PROPIEDADES DE INFORMACIÓN PERSONAL

Considerando aún cuando esta lista seleccionada de información y mecanismos, varias propiedades surgen.

Para ayudar a articular estas propiedades, consideramos cuatro mecanismos diferentes detalladamente: Nombres IM, Mensajes de Hoy, Blogs, y Perfiles (Figura 8). Todos varían en cuanto más una persona revela a otros, cuán frecuente la información es cambiada, y su sensibilidad al tiempo. Cada sistema provee las pistas para la formación de un cuadro de cinco propiedades claves resaltando la información personal que la gente publica sobre ellos: contenido, frecuencia de

actualización y tiempo de vida, valor de archivo, y audiencia esperada. Estos son resumidos en la Tabla 2 y discutidos abajo.

Nombres de Display IM son los campos en el Mensajero Instantáneo (IM) de los clientes, diseñados de modo que una persona pueda mostrar un nombre fácilmente identificable a sus contactos en vez del default y posible criptación de direcciones IM. Como se mencionó antes, la gente se apropiaron de este campo añadiendo información suplementaria, usándolo así para comunicar más que su nombre (Grinter y Palen, 2002; Piepmeyer, 2003; Smale y Greenberg, 2005). Dada las limitaciones de espacio de este campo de nombre, la información es típicamente muy breve - por lo general una palabra o la frase corta. En Smale y Greenberg (2005) estudiamos y clasificamos contenidos de nombre de demostración. Para el contenido, encontramos que los cuatro mayores elementos de comunicación enviados a aquel campo, otro que el nombre de alguien, eran que el humor de la persona, un comentario personal, la ubicación corriente de alguien, y actividades corrientes. Su importancia de tiempo también fue encontrada para ser sumamente volátil, donde la gente lo actualizó con frecuencia, aún varias veces por día, para relatar su estado momentáneo de ser (Smale y Greenberg, 2005). Es decir esto típicamente mantiene la información de momento-a-momento que rápidamente puede salir de fecha. Quizás debido a esta alta volatilidad, cambios no son archivados ni realmente aparecen ser una necesidad de querer tal archivo. Finalmente, la información es difundida a una audiencia limitada, p. ej., que la lista de contacto de la persona (Tabla 2, la segunda columna para un resumen).

Los Mensajes de Hoy son correos electrónicos breves pero regulares que una persona envía a compañeros de trabajo para comunicarles lo que uno hace y facilitar la coordinación (Brush & Borning, 2005). Su contenido típicamente detalla actividades relevantes organizacionales de alguien a lo largo del día, y a menudo incluye una lista "Para Hacer" de tareas para ser completadas. El desafío de Mensajes de Hoy no es el envío y la entrega, sino en el desarrollo de una cultura alrededor de su empleo regular. Para el contenido, la longitud de mensaje es por lo general modesta, p. ej., por lo general en forma de una o dos listas numeradas de fragmentos cortos pero descriptivos de la información que pueden ser mostrados sin scrolling. En un trabajo orientado a la cultura, el contenido de mensajes diarias típicamente describe las actividades diarias e intencionadas del remitente, pero es a menudo vacío de otra información (Brush & Borning, 2005). La Información del tiempo de vida es modestamente volátil, como cubre la actividad diaria y/o acumulativa desde el último envío. Su valor de archivo varía. El archivar es a veces automático, a veces es manual, p.ej., automático si es enviado a una lista de direcciones archivada, y manual si los receptores escogen grabar manualmente (quizás con criterio selectivo) un mensaje de hoy en sus propios archivos, o si el remitente los guarda como un diario personal.

La mayor parte del tiempo sin embargo, mensajes de hoy no son guardados; ellos son desechados inmediatamente después de la lectura. Los mensajes de Hoy trabajan porque una persona puede enfocarlos a una audiencia selectiva, p.ej., fijar específico la gente o una lista de

direcciones como designado en el campo 'Para' (Tabla 2, la tercera columna).

Los blogs son páginas web, son entradas cronológicas que invitan al comentario. Los blogs tienden a ser de dos formas: blogs tópicos y blogs personales; esto es el contenido posterior que es del interés para nosotros. La gente mantiene blogs personales por varios motivos:

Para documentar su vida, proporcionar comentario, articular ideas, o expresar emociones profundamente sentidas (Nardi, Schiano, y Gumbrecht, 2004; Nardi, Schiano, Gumbrecht, y Swartz, 2004). Los que lo hacen para documentar su vida usan blogs como un diario personal, una crónica o el boletín de noticias, un álbum de foto o el libro de recuerdos, un folleto de viajes, o una actualización de estado (Schiano, Nardi, Gumbrecht, y Swartz, 2004).

Las entradas de blog pueden ser muy breves, pero son por lo general de longitud sustancial y muy detallada. Mientras algunas personas actualizan su blog varias veces por día, la mayoría típicamente lo actualizan cada varios días.

Su contenido tiene un tiempo de vida modesta, y las entradas típicamente incluyen acontecimientos más significativos o acontecimientos personales juzgados dignos de enviarse. Las entradas de blog son cronológicamente ordenadas como un archivo; la adición de nuevas entradas empuja a las más viejas fuera de la página principal y en el archivo (el propietario puede quitar/corregir entradas, pero esto es raro). Así mientras la vieja información puede ser recuperada, la página web de blog típica parece diferente día a día dependiendo del número

de entradas. Finalmente, las entradas de blog son dirigidas a una más amplia comunidad. A menos que se marque expresamente como privado, la mayor parte de blogs son abiertos al público general de Internet para verlos (Tabla 2, la cuarta columna).

	IM Names	Today Messages	Blogs	Profiles
Content	Brief surface information	Detailed daily activity information	Detailed life events, activities and commentary	Basic personality and person characteristics
Update frequency	Up to several times a day	Daily / every few days	Daily to weekly to monthly	Once; rarely changed
Lifetime	Volatile			Stable
Archival Value	Not logged	Can be saved as archived messages, but often deleted	Archive available for blog or site lifetime	Available for duration of account or site lifetime
Audience	IM contact list only	Specific email recipients, mail list	General internet public, unless specifically restricted	General internet public, or limited to site members

Tabla 2: Panorama de cuatro áreas y su relación con los cinco atributos

Los perfiles generalmente son conectados a una cuenta de usuario particular sobre varias ubicaciones de sitio web sobre la Internet, son la parte de la página inicial de una persona, o son la parte del grupo de juego o chat. El contenido de perfil típicamente colecta características de la persona y su personalidad, p.ej., la edad, el género, el estado civil, la ocupación, direcciones e información de contacto, y las listas de cosas favoritas. El contenido de perfil para sitios diferentes puede ser construido con cuidado por su propietario con información selectiva, adornada o aún imaginaria para comparar la situación y acceder a los otros de intereses similares (boyd, 2004). Lo que los otros ven es lo que la persona quiere revelar.

Los perfiles por lo general son editados sólo una vez, y su contenido es generalmente muy estable. La Información cambia sólo cuando alguna característica duradera personal cambia, p.ej., un cambio de trabajo altera la información de empleo y el contacto. La información personal contenida en perfiles es rara vez influenciado por actividades diarias o el estado momentáneo. Por definición, los perfiles son archivables, y son guardados de por vida en línea por la persona.

Ellos también son requeridos para el consumo general. Dependiendo del sitio, los perfiles están públicamente disponibles para la inspección, o están disponibles estrictamente a otros miembros del sitio (p.ej., una intranet de organización) (Tabla 2, la última columna).

En resumen, mientras toda la información enviada es sobre uno mismo, esto puede diferenciarse según las propiedades siguientes (Tabla 2):

- Los rangos de contenido se extienden desde lo que pasa en este momento a acontecimientos de vida de largo plazo, y registra breves a características estables personales.
- La frecuencia de Actualización puede extenderse de varias veces por día a muy de vez en cuando a raras veces.
- Tiempo de Vida: la información puede cambiarse rápidamente y tener poco valor una vez que esto se añeja, o puede ser muy estable.
- Los rangos de Audiencia se extiende de muy personal al gran público, p. ej., de contactos personales y pequeños grupos de trabajo a comunidades de organización, y aún a la población general de Internet.

- Valor de archivo: algunos envíos inmediatamente son desechadas después de que han sido leídos; algunos son archivados con criterio selectivo; unos son mantenidos como una cronología de largo plazo, los otros forman un registro estático.

DISEÑO DE PREMISAS

El Sistema pretende integrar el envío de información personal en un sistema simple. Su objetivo era de diseñar una herramienta simple que actúa como un portal a múltiples aplicaciones de publicación diferentes, donde esto complementaría y/o simplificaría que ahora se haga. Esto proveería de la gente de un interfaz único para coleccionar, consolidar y publicar información personal y exquisiteces diarias.

Decidimos concentrar nuestros esfuerzos iniciales en lo que parece ser el área más desafiante: el espacio de la información que se extiende sobre Nombres IM, Mensajes Hoy, y Blogs.

A diferencia de perfiles, los cambios en estas otras áreas son un resultado de actualizaciones regulares a la información personal de la gente. Nosotros basados nuestras premisas de diseño en los siguientes requisitos.

1. Apoyar la edición espontánea del estado de información personal. Del estudio de nombre IM displayado (Smale y Greenberg, 2005) sabemos que los cuatro principales elementos de comunicación que la gente envía eran cambios de momento a momento del humor, la ubicación, la actividad y los pensamientos personales. La ubicación es

también un elemento tradicional en perfiles en línea, y los indicadores de humor son comunes en sistemas de software de blog.

2. Publicar la información de estado personal al campo de nombre de demostración en clientes IM. Reconociendo que la gente ahora usa a uno o varios clientes IM, la información de arriba, el instrumento automáticamente debería publicar esta información de estado en el espacio existente de nombre de demostración IM.

3. Apoyar el mantenimiento dinámico de tipos de actividad y listas para hacer. Nosotros vimos que los mensajes de hoy a menudo contienen tipos de actividad y listas para hacer, aunque estos tienen que ser creados retroactivamente cuando el mensaje de hoy es compuesto. En cambio, argumentamos que la gente dinámicamente debería mantenerlos en marcha. La adición de artículos debería ser el esfuerzo mínimo, y debería transferir el día (si es deseado) a través de días múltiples (mirar también Bellotti et al., 2004). Las actividades de grabación y los artículos para hacer son todas las actividades que son las más fáciles para lograr en el contexto, p. ej. la grabación una actividad como es completado, o la adición de artículos para hacer como uno piensa en ello.

4. Apoyar la reunión de información exquisita. Muchos envíos de información recogen partes de información que la gente busca, p.ej., un enlace interesante o una foto curiosa. Estas 'informaciones exquisitas' son generados esporádicamente a lo largo del día. Nosotros y otros creemos que la gente debería ser capaz de capturar espontáneamente y juntar estos artículos como se encuentran (p.ej., Markopoulous et al.,

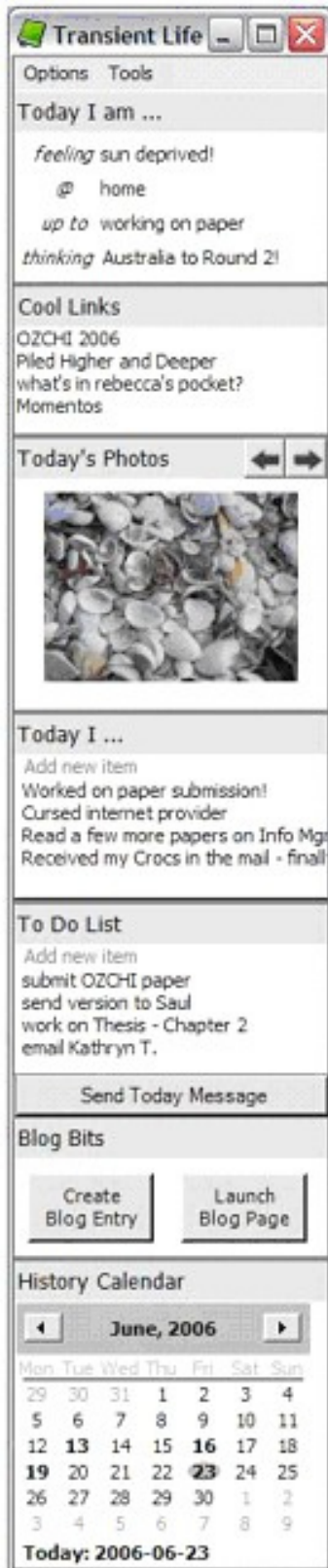


Figura 9: Sistema de Vida Transitoria

2004). Cuando un momento pasa, la información puede verse no ser digna de compartir, o puede ser difícil de recordar más tarde.

5. Publicar tipos de actividad, listas para hacer, y exquisiteces colectadas como 'deseado'.

Sabemos que la gente quiere compartir sus actividades y descubrir artículos de interés con otros, sea para la investigación, el trabajo, la conversación o el entretenimiento.

La actividad y listas para hacer dan a los lectores el sentido de objetivos de corto plazo de alguien y si ellos fueron logrados, mientras las exquisiteces son cosas con significado de invocar el interés de otros. Mientras hay algunos mecanismos que permiten para fijar esta información a una comunidad cercana en tiempo real (McEwan y Greenberg, 2005), sugerimos que un Mensaje de Hoy automáticamente construido de estos elementos de información trabajaría también.

6. Proporcionar mecanismos para recoger y publicar comentarios personales, diarios detallados, y reflexiones a blogs.

Los millones de personas ahora mantienen blogs sobre la

Internet para publicar la información a otros sobre sus vidas (Schiano et al., 2004). Como mencionado, la tasa en la cual las entradas son

compuestas y enviadas es sumamente variable. Creemos que la creación de entrada de blog debería ser sumamente liviano de peso, donde una persona continuamente pueda editarla y añadirle como sea su deseo a lo largo del día, la semana o aún el mes. El envío de correspondencia ocurriría sólo cuando la persona se siente lista de compartirlo con otros.

7. Proporcionar el acceso a una historia personal de información juntada. El valor de archivo de información es sumamente variable y difícil de predecir. Por consiguiente, recomendamos que el instrumento automáticamente mantenga un registro de cada publicación, y que estas publicaciones sean fácilmente recuperables. Como la información en estas publicaciones tiene una vida limitada, la historia debería ser visible sólo si el usuario explícitamente quiere verlo.

Todas estas exigencias comparten criterios comunes: la información debería ser compuesta y registrada tal cual pasa, y públicamente debería ser diseminado de un modo oportuno y por un medio que refleja su contenido. Esto que sigue requiere que el instrumento sea fácilmente accesible y sumamente ligero de peso para usar.

VIDA TRANSITORIA: DESCRIPCIÓN

Después de las exigencias de arriba, diseñamos Vida Transitoria (Figura 9). Esto es un instrumento que permite a la gente fácilmente recoger información como ocurre. Esto también distribuye la información recolectada de varios modos correspondiente a las diferentes salidas de

información personal, p. ej., es selectivamente mostrado en el campo de nombre de demostración de IM, enviado por correo electrónico como un Mensaje de Hoy multimedia (Figura 9), añadida a un blog, y registrado como una historia. Como ilustrado en la Figura 8, Vida Transitoria esta diseñada como slidebar estrecho localizado en el lado de la pantalla; el intercambio es constantemente visible y el acceso contra el espacio de pantalla.

Vida Transitoria contiene ventanas retallables, cada uno colectando y mostrando contenidos diferentes de la información. Para una colección fácil de información, una persona puede añadir y editar la mayor parte de las ventanas por la mecanografía directa, y arrastrando y dejando caer objetos encontrados. Su estilo de interfaz es minimalista; se concentra en la recolección y la demostración de la información más bien que controles complejos. Por un menú, la gente también puede corregir propiedades de perfil duraderas y opciones de sistema. Corrientemente el sistema de Vida Transitorio contiene siete enlaces de la información, individualmente descritos debajo.

HOY SOY... (Figura 8, la ventana superior) deja a la gente registrar e inmediatamente distribuir uno o varios de los cuatro elementos de información de estado personales: humor (sentimiento), posición (@), actividades (hasta qué) comentarios (que piensan) a lo largo del día. Como esta información es sumamente volátil, inmediatamente es publicado al espacio de demostración de un sistema externo de mensajería instantánea, de modo que los otros puedan verlo. Aunque

nosotros preveamos aquella esta información sería publicada a todos los clientes IM usados por su propietario, nuestros trabajos de versión corrientes sólo con el Mensajero MSN. La Información también es registrada como una historia a base de tiempo en el Mensaje de Hoy de modo que los otros puedan repasar las actividades momentáneas de una persona (Figura 9, el lado derecho).

ENLACES COOL (Figura 8, la segunda ventana) permite a una persona recoger páginas interesantes de Internet encontradas durante el día.

Mientras su objetivo primario es de compartir enlaces interesantes con otros, la ventana también puede ser usada por su propietario como un lugar para almacenar rápidamente hallazgos personales relevantes para la revisión posterior.

Páginas de Web son añadidas como ellos son vistos: una persona simplemente arrastra y deja caer el icono 'del enlace' sobre la barra de dirección del navegador de web en la ventana. El propietario fácilmente puede visitar de nuevo una página pulsando el enlace en la ventana.

FOTOS DE HOY (Figura 8, la tercera ventana) dejan uno recoge fotos en una manera similar a Enlace Cool). Esto también funciona con la funcionalidad de gota y arrastre. También es añadido a mensajes salientes hoy (' los Cuadros que encontré hoy ' la sección en la Figura 9). Las imágenes pueden venir de varias fuentes. Ellas pueden ser fotos personales localizadas sobre el ordenador de alguien, o ellos pueden ser agarrados de páginas web vía el arrastramiento y la caída. Las

imágenes automáticamente son reducidas en el tamaño entonces ellos no abruman al recipiente.

HOY ... (Figura 8, cuarta ventana) permite entrar en tareas completadas, actividades y acontecimientos en una lista. Esta lista es publicada más tarde en el Mensaje de Hoy (la sección Hoy yo ... en la Figura 9). Con esta ventana, una persona inmediatamente puede añadir artículos como ellos son logrados, en vez de tratar de recordar todo lo que fue hecho al final del día.

PARA HACER LA LISTA (Figura 8, la quinta ventana) deja a la gente añadir artículos y tareas que ellos esperan hacer en los pocos días siguientes. Como con la mayor parte de tales listas, artículos pueden ser reorganizados según su prioridad, y así esto tiene el valor inmediato personal.

LAS PIEZAS DE BLOG (Figura 8) permiten a una persona componer, añadir a, y corregir un ensayo de texto con el tiempo. La persona entonces puede fijarlo a su blog en un tiempo oportuno.

Como este ensayo puede ser grande, el editor de entrada del blog es enviado a otra ventana separada (no mostrado) pulsando el botón 'Crea la entrada de blog'. Esta ventana contiene la última entrada inédita, o una nueva entrada si el previo fuera publicado. La persona puede desechar la entrada corriente, corregir y añadir a su texto, salvarlo para una edición posterior, o publicarlo a su blog. Típicamente una persona

crea una nueva entrada, y le devuelve varias veces antes de la publicación en su blog.

About me today...

June 23, 2006

Today I ...

- Worked on paper submission!
- Cursed internet provider
- Read a few more papers on Info Mgmt
- Received my Crocs in the mail - finally!

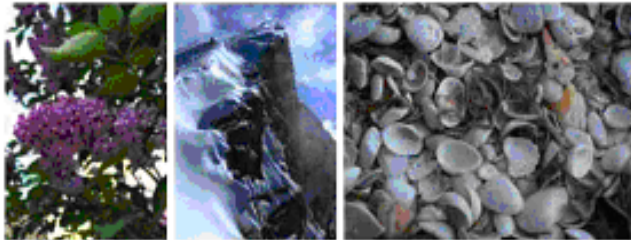
On my ToDo list ...

- submit OZCHI paper
- send version to Saul
- work on Thesis - Chapter 2
- email Kathryn T.

Links I thought were interesting:

- OZCHI 2006
- Piled Higher and Deeper
- what's in rebecca's pocket?
- Momentos

Pictures I ran across today:



Today messages are emails outlining a person's activities throughout the day. This rich today message not only provides details on what has been done though out the day, but also personal status information and tidbits about interesting things that are worth sharing

Figura 10: Un típico Mensajes de Hoy en multimedia generado con Vida Transitoria

CALENDARIO DE HISTORIA (Figura 8, la ventana inferior) deja a uno fácilmente repasar la información archivada pasada. Los días con contenido son resaltados en el calendario. En la práctica, esperamos que este calendario sea usado sólo de vez en cuando, p.ej., si una persona quiere recuperar algo reunido durante un día particular. Mientras esto es un empleo discrecional, creemos que el costo de automáticamente crear un archivo personal que más a menudo es almacenado que recuperado es leve. Finalmente, la gente puede especificar como la información es archivada.

2.2.6.3 Sistema Multiagentes¹⁴

Empleo de tecnologías de agentes para la gestión de tutorías en un campus universitario

Descripción del sistema

En esta sección se muestra una aplicación desarrollada en la cual se ha evaluado la aplicabilidad de la tecnología de agentes integrando diferentes dispositivos. En la versión actualmente implementada de este sistema, un estudiante puede consultar los horarios de tutorías de un profesor determinado de acuerdo con sus preferencias usando un teléfono móvil o una PDA. Una vez que ha seleccionado un horario, el estudiante puede establecer una reserva de un periodo de tiempo

¹⁴ Tomado de la Referencia [17]

específico para su tutoría. Este proceso es establecido entre un agente interfaz estudiante y un agente gestor de horarios de tutorías.

Arquitectura del sistema

Para elaborar un diseño acertado del sistema, debemos identificar todos los componentes involucrados en la gestión de tutorías en un campus universitario. En primer lugar, el servicio está dirigido a dos tipos de usuarios, que son, profesores y estudiantes. Cada uno de ellos tiene diferentes necesidades y diferentes características. Debido a esto, se decidió incluir en el sistema dos agentes diferentes para la gestión fundamentalmente de interfaz con dichos usuarios: *Teacher Agent* y *Student Agent*.

Por otro lado, es importante poder tener acceso a la información en cualquier momento y lugar dentro del campus universitario. Por esta razón, tanto el *Teacher Agent* como el *Student Agent* fueron diseñados para ser usados sobre dispositivos móviles como un teléfono móvil o un PDA. Para un profesor es importante tener la capacidad de administrar la información acerca de sus tutorías de forma ordenada y eficiente. Además, la gestión del horario de un profesor debe tener continuidad temporal para poder ofrecer sus servicios de forma ininterrumpida.

Debido a esto, se planteó introducir otra entidad que administrase toda la información sobre los horarios de tutorías de un profesor específico, el *Tutorship Manager Agent*. Este agente debe mantener la información disponible todo el tiempo. Pensando en una arquitectura distribuida y eficiente, cada *Teacher Agent* tendrá un *Tutorship Manager Agent* asociado para manejar su horario de tutorías.

Considerando que los agentes deben ser capaces de ponerse en contacto de alguna forma entre ellos, necesitamos una entidad que ofrezca un conjunto de servicios que faciliten la comunicación a otros agentes usando algún conocimiento sobre los requerimientos y las capacidades de esos agentes. Dicha funcionalidad la aportará la figura de un *Broker Agent*. El broker, además de las facilidades de comunicación, se encarga de realizar la búsqueda, procesar la información obtenida y enviarla directamente al estudiante interesado.

Resumiendo, el sistema está constituido básicamente por cuatro tipos de agentes: el *Broker Agent*, el *Teacher Agent*, el *Tutorship Manager Agent* y el *Student Agent*. Las funcionalidades principales de estos agentes son: (I) El *Broker Agent* contiene información actualizada de los *Tutorship Manager Agents* registrados con él. Este se encarga de establecer la comunicación entre el *Student Agent* y el *Teacher Agent*. (II) El *Tutorship Manager Agent* maneja toda la información acerca del horario de tutorías de un profesor específico. Será exclusivo a un profesor, y provee los siguientes servicios a los profesores: consultar, modificar y establecer restricciones en su horario de tutorías, y consultar las tutorías asignadas para un día en particular. (III) El *Teacher Agent* permite a los profesores acceder y utilizar los diferentes servicios ofrecidos por el *Tutorship Manager* mediante una GUI ejecutándose en su dispositivo móvil. (IV) El *Student Agent* será exclusivo para cada estudiante interesado, y provee los siguientes servicios: consulta de horario de tutorías por materia o nombre de profesor, reserva de tutorías y consulta de tutorías solicitadas para un día en particular. La

comunicación entre el estudiante y su agente se llevará a cabo mediante una GUI asociada que el estudiante ejecuta en su dispositivo móvil. Debido a que los agentes necesitan comunicarse entre ellos, fue indispensable establecer un vocabulario conceptual común como una representación de la información para establecer y controlar tareas. Toda esta información queda representada dentro de una ontología específica. La ontología implementada proporciona una descripción detallada de los profesores, tutorías, estudiantes, horario de profesores, etc. Adicionalmente, la ontología desarrollada contiene acciones y predicados que ayudan a controlar y establecer tareas dentro del sistema.

La Figura 11 muestra la arquitectura del sistema multi-agente implementado ilustrando la comunicación entre los diferentes agentes comentados anteriormente. Básicamente, si el usuario (profesor o estudiante) se encuentra ubicado en cualquier punto dentro del campus, usando la red *wireless* puede utilizar el sistema de gestión de tutorías, enviando preguntas o acciones al *Broker Agent*. El *Broker Agent* establecerá una interacción con el *Tutorship Manager Agent* correspondiente intentando encontrar la información deseada del profesor. Posteriormente, el *Broker Agent* enviará la respuesta apropiada al agente profesor o estudiante interesado según sea el caso.

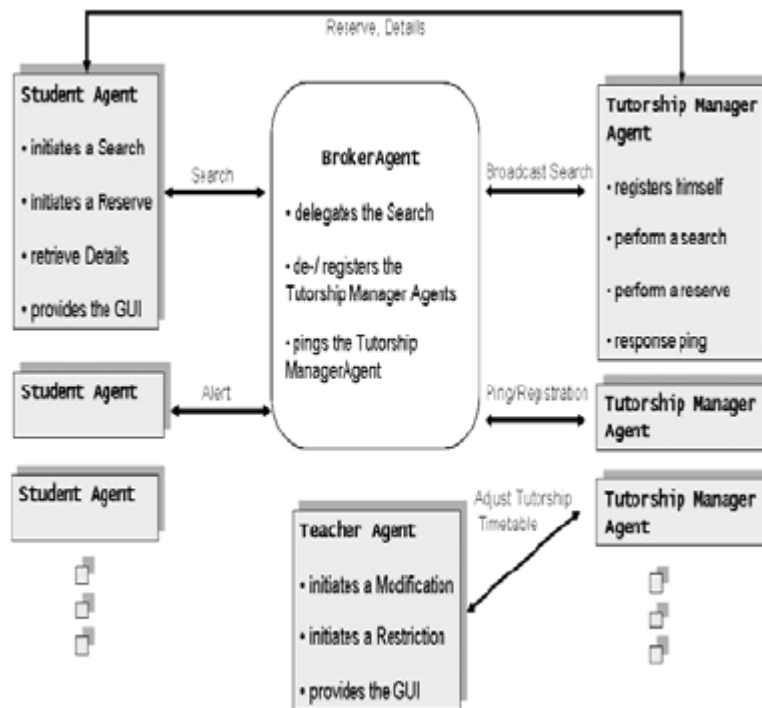


Figura 11. Arquitectura del Sistema multi-agente

Después de que el usuario obtiene una respuesta de su búsqueda, este establecerá una comunicación directa con el *Tutorship Manager Agent* correspondiente.

Los usuarios pueden establecer la conexión requerida usando el acceso *wireless* que se encuentra disponible en el campus universitario.

La selección de este tipo de conexión se hizo debido a que esta cubre la mayor parte del campus y permite una velocidad de conexión apropiada.

Obviamente esta opción es posible únicamente en dispositivos móviles como una PDA con acceso mediante *wireless* integrado. En el caso de teléfonos móviles, la conexión puede ser establecida mediante GPRS.

Este proceso se ilustra en la Figura 12.

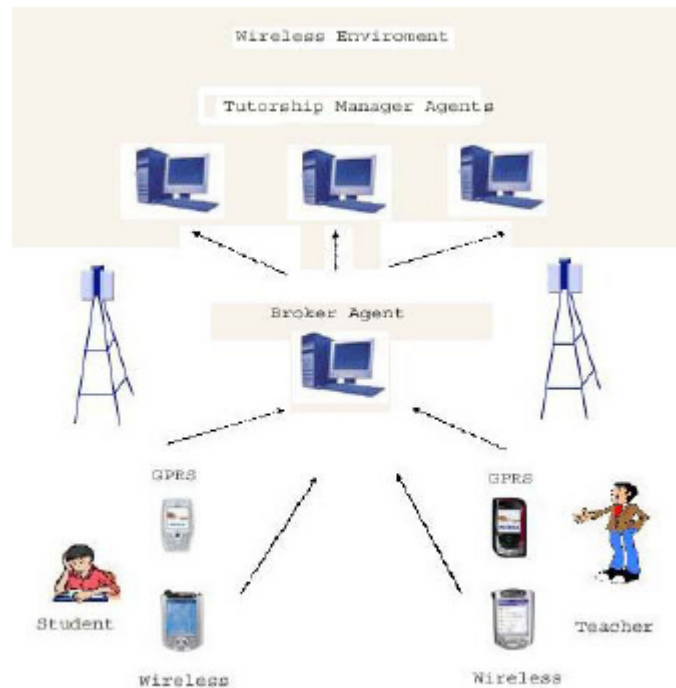


Figura 12. Esquema sistema ejecutándose a través de conexiones diferentes

2.3 APLICACIONES DE PIM¹⁵

Para entender cuáles son las aplicaciones de PIM en la actualidad, ABETA, Akira & KAKIZAKI, Ken'ichi escribieron "*Operation Record based Work Events Grouping Method for Personal Information Management System*" durante una Conferencia de Software y Aplicaciones en 1998, en cuyo trabajo proponen un sistema el cual posee toda la funcionalidad que soporta las operaciones básicas de PIM.

¹⁵ Tomado de la referencia [18]

REGISTRO DE OPERACIÓN BASADO EN EL MÉTODO QUE AGRUPA EVENTOS DE TRABAJO PARA UN SISTEMA DE GESTION DE INFORMACIÓN PERSONAL

Resumen del trabajo

Se propone un método que acumula y reutiliza el conocimiento de trabajo automáticamente extrayendo el conocimiento de los registros de operación de usuarios durante el empleo ordinario a un sistema de Dirección de Información Personal (PIM). Se asume que los acontecimientos de trabajo relacionados en el PIM son introducidos y referidos consecutivamente por el usuario. El método que extrae el conocimiento disponible construye un flujo de proceso agrupando todos los acontecimientos relacionados y detectando hitos basados durante el tiempo y la orden de acontecimientos objetivo de operaciones. Por lo general, la referencia a acontecimientos es la inspección del usuario de acontecimientos sobre el calendario sin implicar operaciones, de modo que la grabación de un acontecimiento objetivo y el tiempo de referencia son difíciles. Para registrar el comportamiento de referencia del usuario exactamente, se introduce un globo de ayuda a la función basada para el apoyo de referencia de acontecimiento.

Los trabajadores pueden realizar varios proyectos de trabajo satisfactoriamente aprendiendo el conocimiento de trabajo incluyendo los detalles, la duración, recursos materiales, y trampas potenciales. Como los trabajadores a menudo realizan el mismo tipo de proyecto de trabajo, ellos pueden reutilizar el conocimiento de trabajo hecho antes para hacer mejores trabajos y más rápido.

Si trabajadores inexpertos pudieran reutilizar el conocimiento de trabajo que los compañeros de trabajo experimentados tienen, los trabajadores inexpertos también podrían realizar los proyectos de trabajo bien y a tiempo. Para hacer este conocimiento de trabajo disponible para la reutilización, debemos registrarlo de algún modo. Hay algunos sistemas que comparten y reutilizan el conocimiento acumulándolo en una computadora. Sin embargo, la dificultad y costo de representar e introducción del conocimiento es a menudo prohibitiva.

Los programas para un proyecto de trabajo consisten en algunos acontecimientos detallados y las relaciones entre acontecimientos como el orden y el tiempo de estos acontecimientos. Por tanto, los cronogramas para un proyecto de trabajo presentan un “flujo de proceso” en un PIM. Pero un ordinario PIM tiene que registrar separadamente los eventos individuales colocados sobre el calendario. No puede manejar relaciones entre eventos que trabajan de eventos que esencialmente tienen, y no podemos reutilizar todos los eventos de trabajo relacionados como un flujo de trabajo bulto.

Por lo tanto, un PIM ideal tiene que comprender relaciones entre acontecimientos y manejar programas como un flujo de proceso, incluyendo todos los acontecimientos necesarios relacionados con un proyecto de trabajo y el orden de los acontecimientos.

En sistemas de gestión de proyecto que manejan las relaciones entre acontecimientos para un proyecto de trabajo, un gerente cataloga todos los acontecimientos necesarios para un proyecto de trabajo por adelantado con escrupuloso cuidado e introduce sus relaciones

registrándolos. En países occidentales, un proyecto de trabajo es planificado y realizado en un estilo descendente donde el gerente decide el contorno de un proyecto de trabajo y delega tareas detalladas a sus trabajadores. En este estilo, un sistema de gestión de proyectos es conveniente. Sin embargo en Japón, un proyecto de trabajo es planificado y realizado usando un estilo ascendente en donde las negociaciones ocurren entre los trabajadores y la suma de sus tareas detalladas decide los programas del proyecto de trabajo. En este estilo, si el sistema de gestión de proyecto no es conveniente, un gerente no puede usarlo. Típicamente en empresas japonesas cada trabajador usa un PIM en cambio. Por lo tanto, un PIM debe automáticamente agrupar acontecimientos de trabajo relacionados y organizar un proceso laboral extrayendo relaciones entre acontecimientos como el usuario interactúa recíprocamente con el PIM en el curso del trabajo diario.

Los grupos de método relacionados en acontecimientos de trabajo basan sus semejanzas entre acontecimientos extraídos del texto contenido dentro de acontecimientos individuales sobre un PIM. Se estudiaron técnicas que calculan las semejanzas entre palabras en cada oración para encontrar oraciones similares en sistemas de tratamiento de lengua natural.

El usuario de un PIM que introduce acontecimientos entiende bien las relaciones entre estos eventos. Creemos que él señala relaciones entre acontecimientos cuando usa el PIM. El método propuesto automáticamente construye un flujo de proceso agrupando

acontecimientos de trabajo relacionados y detectando hitos de las acciones de un usuario sobre el PIM.

El prototipo PIM desarrollado y probado en algunas organizaciones reales confirmó que registra muchas referencias a acontecimientos, y podía hacer grupos de acontecimientos de trabajo relacionados con probabilidades exactas analizando los registros de operación del usuario. Además, encontraron que el método podía detectar hitos basados en relaciones entre acontecimientos y construir un flujo de proceso laboral.

Cronograma como Flujo de Proceso laboral

Programas sobre un PIM

Como se muestran en la Fig. 13, un usuario PIM hace cronogramas introduciendo acontecimientos en el marco apropiado sobre el calendario, y se referirá a los acontecimientos sobre el calendario para realizar proyectos de trabajo según el programa.

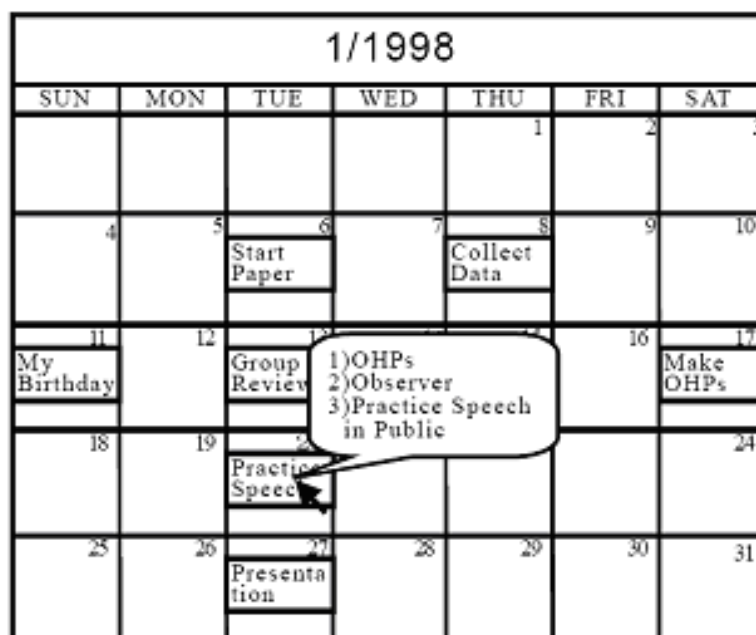


Figura 13: Calendario de un PIM

La relación entre acontecimientos se explica en una serie secuencial de de acontecimientos (Fig. 13). Como se muestra en la Fig. 13 y las cajas alineadas de acontecimientos en la Fig. 14, un PIM ordinario solamente muestra los acontecimientos individuales y no maneja las relaciones entre acontecimientos. Por lo tanto es imposible comprender la relación entre acontecimientos a no ser que un usuario produzca el cronograma. A pesar de la introducción de la información de acontecimientos en un computador, todavía existe el problema que no podemos reutilizar aquella información como un " conocimiento de trabajo " que incluye la relación entre acontecimientos.

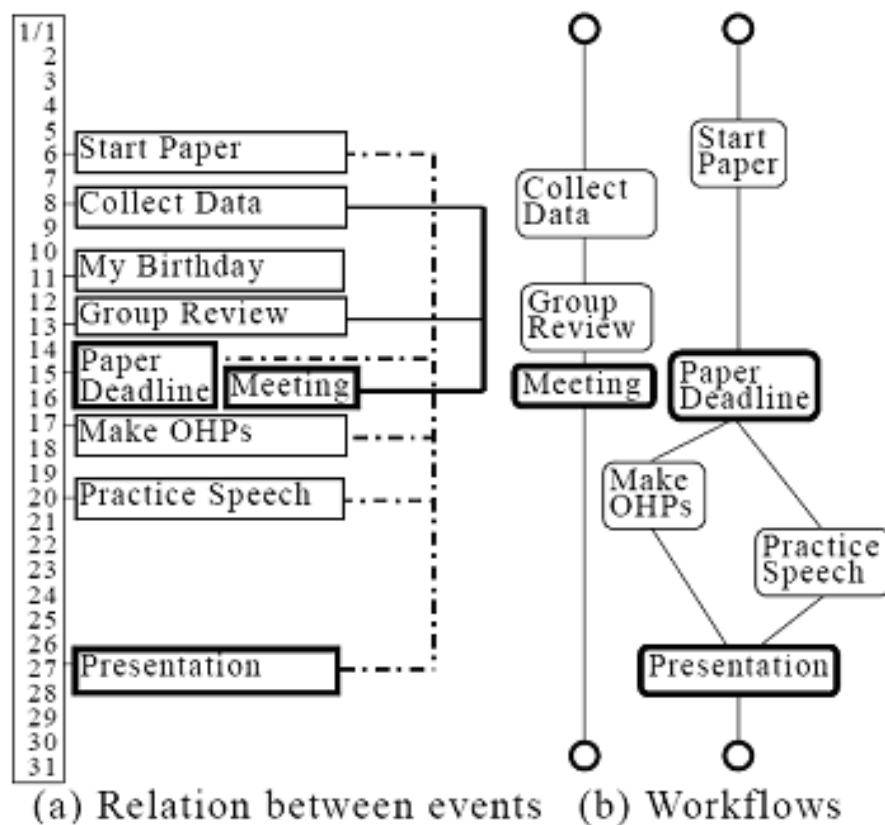


Figura 14: Relación entre eventos y flujo de trabajo

El flujo de proceso en un PIM

Un trabajador experimentado que usa el PIM se imagina las relaciones entre acontecimientos mostrados como líneas que se unen e identifica puntos críticos mostrados como cajas en negritas en la Fig.14(a), así el trabajador usa la información del PIM como conocimiento de trabajo. Los puntos críticos son acontecimientos primarios tanto en la planificación como en la ejecución de programas, llamados "hitos".

Los hitos determinan el orden de los acontecimientos que preceden y siguen. El PIM ideal debería distinguir el entendimiento del usuario de relaciones e hitos para producir un flujo de proceso diagrama PERT como el de la Fig. 14(b). El PIM también debería retornar el flujo de proceso de modo que el usuario pueda realizar el proyecto de trabajo de una manera más eficiente usándolo.

Estructura "de un flujo de proceso" en un PIM

Para manejar relaciones entre acontecimientos en un PIM, se introdujo una estructura que llaman "un flujo de proceso" en un PIM como se muestra en la Fig. 14. Un flujo de proceso está compuesto de algunos acontecimientos y relaciones entre estos acontecimientos. En la Fig. 14, las cajas blancas muestran acontecimientos y la línea negra entre las cajas muestran la relación entre acontecimientos.

Cada acontecimiento tiene la información fundamental tales como el título, la fecha de inicio, la fecha final, el período, o el tiempo etcétera. Las cajas negritas son hitos. En nuestro PIM, más lejos, cada acontecimiento puede tener la información en cuanto a si esto es un hito

o no y si una relación entre acontecimientos tiene una restricción. Por ejemplo, un acontecimiento tiene que terminarse antes de que otro acontecimiento haya comenzado.

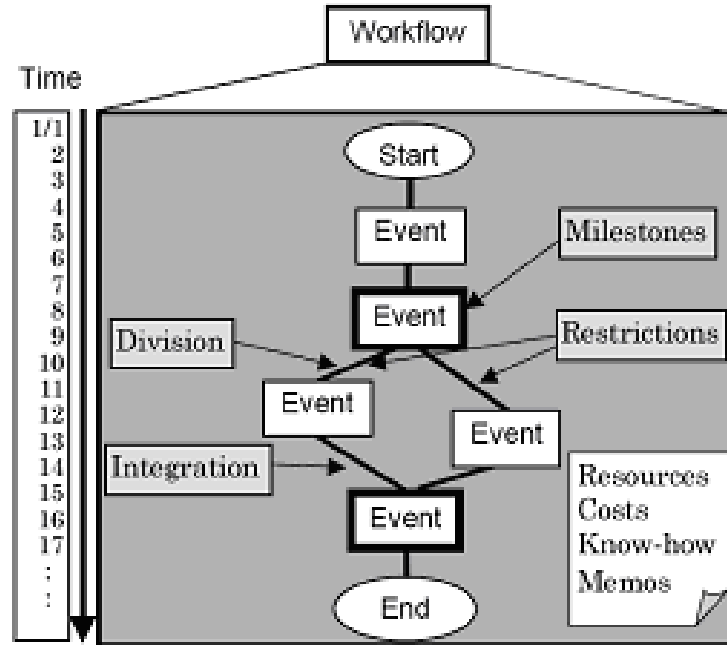


Figura 15: Estructura de un flujo de trabajo

Un flujo de proceso cubre todos los acontecimientos necesarios para hacer un proyecto de trabajo y todas las relaciones entre aquellos acontecimientos. Además, podemos adjuntar la información asociada a un flujo de proceso, como recursos, costos y puntos significativos, y manejar un flujo de proceso y todos juntos. Se hace posible transferir, suprimir, o reutilizar un flujo de proceso como un "conocimiento de trabajo".

Sistema Propuesto

La Fig. 15 ilustra el sistema ideal. Se propuso un PIM que automáticamente obtenga el conocimiento de trabajo de la mente del usuario y lo acumule como un flujo de proceso usando una base de datos.

Reutilización de cronogramas acumulados

En el ideal PIM, los usuarios pueden elegir un flujo de proceso de un proyecto de trabajo similar hecho antes y reutilizar sus cronogramas fundamentales de una base de datos. Estos cronogramas incluyen casi todo el conocimiento necesario de trabajo porque hacemos las tareas similares repetidamente. Se catalogan las ventajas de reutilización abajo:

- Los usuarios no tiene que hacer cronogramas para un proyecto de trabajo similar otra vez.
- No hay ninguna omisión de acontecimientos necesarios para un proyecto de trabajo en cronogramas acumulados en una base de datos.
- Los usuarios pueden guardar recursos necesarios para un proyecto de trabajo con seguridad.
- Los usuarios puede acumular conocimientos basados en sus experiencias.

Trabajadores experimentados e inexpertos pueden hacer los cronogramas que pueden hacer trabajos mejor y más rápidos reutilizando flujos de procesos.

El refinamiento de programas acumulados

Los usuarios modifican los programas fundamentales para adaptarlos para su proyecto de trabajo corriente y luego realiza las tareas allí. Cuando los programas no van bien, el usuario modifica los programas

otra vez. La suma de estas modificaciones es el conocimiento verdadero de trabajo porque el usuario experimenta el proyecto de trabajo en realidad. Entonces el ideal PIM mejora el flujo de proceso de los programas fundamentales en la base de datos basada en aquellas modificaciones. Llamamos a este mecanismo "el refinamiento". En el sistema propuesto, los programas se hacen disponibles como ellos son reutilizados. Se catalogan las ventajas de refinamiento debajo:

- Mejora el conocimiento de trabajo en calidad y cantidad.
- Adapta el conocimiento de trabajo a cambios del entorno de un proyecto de trabajo.

Por ejemplo, el mecanismo de refinamiento automáticamente puede adaptar programas acumulados al cambio revisando sus duraciones basadas en un análisis de las modificaciones del usuario a los acontecimientos.

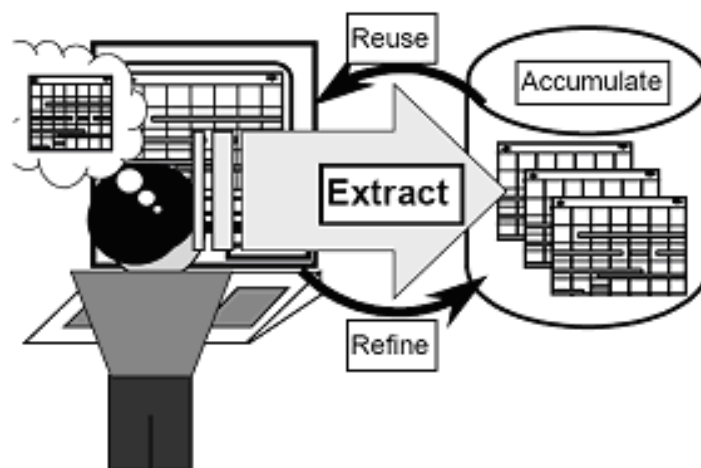


Figura 16: Sistema Propuesto

Sin embargo, si los programas acumulados son muy pocos o pobres en calidad, los usuarios no reutilizarán aquellos programas. Entonces los

programas no serán refinados sobre múltiples empleos. Es importante que el sistema extraiga programas sin errores o insuficiencia de conocimiento de trabajo.

Extracción de las acciones del usuario

Es muy difícil adquirir automáticamente el conocimiento que una persona tiene en su mente. Sin embargo, podemos observar su comportamiento durante el empleo de un PIM. Durante la confirmación de programas de acontecimiento, acciones como un cambio de punto de vista o instrucciones son realizadas basadas en su conocimiento del trabajo. Por lo tanto, observando y analizando estas acciones, es posible comprender el conocimiento de trabajo que él tiene en mente.

2.4 Importancia o Justificación del Problema

Dado el auge de las comunicaciones inalámbricas Wi-Fi¹⁶, Bluetooth¹⁷ y de telefonía celular y el amplio despliegue que han tenido en los últimos años los dispositivos móviles, se ha percibido la potencialidad que tiene este nuevo paradigma de computación conocido como Computación Móvil (“Mobile Computing”) en todos los ámbitos de nuestra sociedad y principalmente, como una oportunidad para la comunidad universitaria de aprender a utilizar estos nuevos modelos de aplicaciones.

La computación móvil, permite independiente del lugar o tiempo, realizar diferentes actividades de procesamiento en los dispositivos móviles

¹⁶ Wi-F.-es un estándar para redes inalámbricas basados en las especificaciones IEEE 802.11. Creado para ser utilizado en redes locales inalámbricas, es frecuente que en la actualidad también se utilice para acceder a Internet.(Wikipedia, La enciclopedia libre)

¹⁷ Bluetooth.- es una tecnología que provee un camino fácil para la computación móvil, para la comunicación entre dispositivos y conectarse a Internet a altas velocidades, sin el uso de cables

conectados a Internet o redes corporativas a través de las diferentes opciones de comunicación. Desde aplicaciones de entretenimiento, mensajería, comercio electrónico móvil (m-commerce), entre otras muchas más que apenas comienzan a desplegarse en nuestro país (como servicios multimedia, vigilancia, etc.) .

A pesar de estos grandes avances en la tecnología celular, solo por poner un ejemplo tenemos¹⁸ :

El Sr. López, arquitecto, toma un avión para desplazarse a Arequipa y revisar la marcha de las reformas que está supervisando en un bloque de apartamentos. Como es habitual en estos casos, las obras se han alargado más de lo previsto inicialmente y este es ya el décimo viaje a la Ciudad Blanca que realiza en lo que va de año. El objetivo de este último desplazamiento es realizar una visita de obras para revisar la estructura de la escalera exterior. El coste del trayecto asciende a un total de 300 euros (entre taxis, vuelos y dietas). Y eso sin tener en cuenta las más de 10 horas invertidas en viajes. ¿Cómo cambia este escenario haciendo uso de la tecnología móvil?

Gracias al servicio de vídeollamada de su teléfono 3G y el de su capataz, no sólo se ha podido revisar la estructura de la escalera a distancia sino que se ha hecho una visita guiada por todo el edificio y se han planificado las próximas 5 visitas virtuales. Al acabar la visita, el arquitecto ya ha enviado cinco mensajes con instrucciones detalladas a proveedores y aparejadores. Y todo en 20 minutos y por cuatro euros.

¹⁸ Tomada de la Referencia 19

La tercera generación de móviles posee como uno de sus principales servicios: la vídeollamada. Aunque con cierto retraso, la tecnología UMTS ya está aquí para ofrecernos cobertura y acceso a Internet a velocidades cada vez más cercanas a los 2Mbps. Un ancho de banda más que suficiente para explotar, además de la vídeollamada, otras aplicaciones que exigen gran capacidad, como la posibilidad de descargar música, juegos e incluso vídeos a través del teléfono móvil.

La idea es que los trabajadores pueden acceder a sus principales aplicaciones de oficina en todo momento y desde cualquier sitio a través de sus dispositivos móviles. Por ejemplo, algunos teléfonos móviles ya son capaces de sincronizarse con el gestor de correo electrónico de Microsoft Outlook, si bien todavía con una capacidad limitada. Estos terminales móviles incorporan cada vez más funciones de agenda electrónica o PDA. Y, en sentido inverso, los PDA incluyen entre sus prestaciones las de teléfono móvil. Uno de los híbridos que más éxito han tenido este año es la denominada Black-Berry. Gracias a su competitivo precio (199 euros), los dispositivos con tecnología BlackBerry, propiedad de la canadiense Research in Motion, en Estados Unidos, se conoce a este tipo de teléfonos con capacidad para consultar el correo electrónico con el sobrenombre de CrackBerry por la adicción que llegan a generar.

A pesar de estos grandes avances en tecnología celular, estos no disponen de un sistema que permita realizar una adecuada gestión de correos electrónicos, de tal manera que su acceso sea simple, rápido y eficaz.

También son cada vez más los ordenadores portátiles que se conectan a Internet con la tecnología inalámbrica de red de área local (WLAN) Wi-Fi. Si bien esta tecnología todavía está sujeta a un radio de acción limitado por el alcance de las antenas (hotspots).

Parece que la oficina sin papel se está imponiendo gradualmente.

La tendencia de futuro apunta a una migración progresiva de usuarios hacia una red multimedia única que sustituirá a las distintas redes de acceso actuales (teléfono fijo, móvil, PC o televisión). Eso no necesariamente supone la sustitución de un tipo de plataformas por otras, sino más bien un uso combinado e inteligente de todas ellas según las necesidades y disponibilidad de cada tipo de acceso.

Pero la convergencia no va a limitarse a plataformas y dispositivos, sino que también afectará a las mismas redes de conexión inalámbrica a Internet.

Por otra parte, la creación de un entorno regulatorio que permita el uso de agentes inteligentes y de la identificación por radiofrecuencia, hará posibles aplicaciones todavía mucho más sofisticadas.

2.5 Problemática actual

A pesar de los grandes avances en tecnología celular, los usuarios que intentan recuperar su información de archivos o contactos entregados en un mensaje de correo electrónico a menudo usan la recordación asociativa basada en las señales sociales y temporales indirectas (como el remitente, destinatario, y fecha del mensaje).

Los usuarios se aprovechan de estas señales accediendo las carpetas del correo electrónico y su bandeja de entrada, mientras se ordena para ver por el remitente, fecha, o una combinación de remitente y fecha para la recuperación. El volumen del mensaje del correo electrónico también es una señal importante para usuarios que intentan localizar la información.

La señal importante para la recuperación puede ser una palabra clave o un el tema a que el contacto o el adjunto está relacionada. El propio mensaje puede contener información explicativa, eso ayuda teniendo sentido del contacto o el adjunto. Pero ninguno ofrece una solución completa para problemas de correo electrónico y PIM.

A pesar de estas ayudas, en la actualidad no hay ningún sistema que permita el acceso simple, rápido y eficaz a la información personal, tal como es el correo electrónico, pues cuando necesitamos ver un mensaje en nuestro dispositivo celular, primero necesitamos que ingresar a la dirección de la página del servidor del correo que necesitamos ver, y luego navegar hasta encontrar el area donde podamos realizar el proceso de autenticación, es decir ingresamos nuestro usuario y contraseña, para posteriormente recién poder acceder a nuestra bandeja de entrada, en la cual posteriormente tenemos que navegar hasta encontrar la información que nos interesa.

Si repetimos esta operación por el número de cuentas de correo electrónico a las cuales necesitamos acceder, la gestión de correos se vuelve más complicado, no solo por el tiempo empleado, la incomodidad que genera la navegación usando un celular, y acceder a información no importante como pueden ser los correos spam o propagandas, en un entorno donde el empleo del tiempo y la gestión de la información es sumamente importante.

CAPITULO III

LA PROPUESTA

3.1 La Propuesta

El Sistema de Acceso a Información Personal desde Entornos con Conectividad Limitada a desarrollar es un sistema basado en agentes móviles que facilita al usuario el acceso a información personal cuando usa dispositivos móviles y una conexión de red inestable, como puede ser las redes inalámbricas.

Se aprovecharán todas las ventajas que nos ofrece la tecnología de agentes, para resolver este problema. Asimismo para la determinación de su uso, se comparará frente al de otras tecnologías existentes como son el Middleware y los Web Services, mediante un adecuado sistema de indicadores que nos permitirá decidir cuál es la mejor tecnología a usar.

Asimismo después de haber elegido la tecnología adecuada, se realizará un proceso de selección acerca de la metodología a usar, para lo cual se utilizarán varios criterios de selección.



Figura 17 - Diseño de Prototipos

El sistema permite acceder a cuentas de correo electrónico para buscar mensajes de correo y leer sólo los que interesen, sin que para ello se tengan que obtener todos los demás mensajes que puede haber en dicha cuenta, poseerá una interfaz similar a la que se muestra en la Figura 17. El sistema además permitirá buscar documentos usando alguno de los sistemas de búsqueda registrados en el sistema y después descargarlos si así se desea, desplazando los procesos más pesados del Dispositivo de Usuario desde el que se ejecuta la aplicación a un dispositivo distinto con mayor capacidad de cálculo, denominado Dispositivo Servidor tal como se ilustra en la Figura 18.

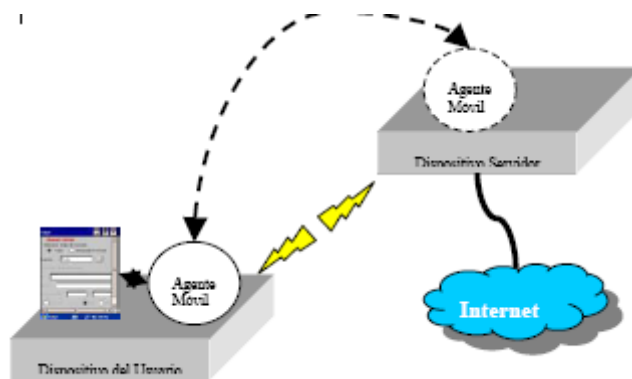


Figura 18 - Distribución de Dispositivos

Cuando se ordena a la aplicación hacer una búsqueda de documentos o de mensajes de correo electrónico, leer mensajes de correo o descargar documentos, la interfaz le ordena a uno de los agentes que forman parte del sistema que realice dicha acción. Este agente (agente móvil) cuando recibe la orden se desplaza al Dispositivo Servidor para que se realicen los cálculos, en donde hace todas las operaciones necesarias para dar respuesta a la petición. Una vez obtenida la información regresa al origen con la información obtenida. Esta información se muestra al usuario a través de la interfaz del sistema.

Para desarrollar este sistema se ha seguido una metodología de desarrollo de Sistemas Multiagente con el fin de probar y valorar el uso de esta clase de metodologías en el desarrollo de un problema real. De entre las metodologías existentes, se ha elegido INGENIAS.

El proceso de desarrollo que se seguirá es el siguiente:

Fase Requisitos: En esta fase se realiza un *Documento de Especificación de Requisitos* que aporta la información necesaria para tener una visión global del sistema. A partir de este documento se pueden identificar las funcionalidades del mismo y se identifican los

Agentes y los Casos de Uso, creando un *Diagrama de Casos de Uso*. Estos casos de uso se priorizan para identificar en que iteración serán desarrollados y posteriormente se detallan.

Fase Análisis: con los Casos de Uso ya identificados se procederá a identificar aquellos en los cuales se produce alguna interacción entre los agentes y se asociarán estos con *Modelos de Interacción*. Tras el desarrollo de estos *Modelos de Interacción* se creará un primer *Modelo de Organización* en el cual se esbozará la arquitectura del sistema e identificarán los flujos de trabajo. Se generará el *Modelo de Entorno* y se crearán los *Modelos de Agente*. Finalmente se creará el *Modelo de Tareas y Objetivos* para generar restricciones de control (los objetivos principales y la relación de las tareas con los objetivos).

Fase Diseño: En esta fase se completará el *Modelo de Organización* mediante el desarrollo de los flujos de trabajo identificados durante el análisis. A continuación se completarán los *Modelos de Tareas y Objetivos* y los *Modelos de Interacción* con especificaciones GRASIA de las interacciones que estos representan. Se definirá la ontología del sistema y finalmente se realizará el diseño de las clases que forman parte del mismo.

Fase de Implementación: se desarrollará el sistema a partir de la especificación del diseño y se realizará un diagrama de componentes indicando la relación entre los diferentes ficheros de código y componentes del desarrollo.

3.2. Evaluación de las Tecnologías de Solución Existentes

Se realizó una comparativa de las Tecnologías expuestas en los puntos anteriores a fin de determinar cual es la más óptima para dar solución al problema planteado; para ello se establecieron los siguientes indicadores:

Indicadores de evaluación:

- **Transparencia en el acceso:** Se refiere a la capacidad de poder obtener recursos de la misma manera, sin importar que estos sean remotos o locales.
- **Concurrencia:** Tanto usuarios como aplicaciones deben de poder operar al mismo tiempo sin que el trabajo de uno interfiera con el del otro.
- **Escalabilidad:** Permite que el sistema pueda incrementar o disminuir su tamaño según se requiera. Esto debe de poder llevarse a cabo sin necesidad de cambiar la estructura del sistema. Es decir, pueden agregarse o quitarse dispositivos.
- **Facilidad de implementación:** Simplicidad en el desarrollo y Adaptaciones.
- **Funcionamiento independiente del ancho de banda:** El ancho de banda es la cantidad de información o de datos que se puede enviar a través de una conexión de red en un período de tiempo dado, Siendo este un factor importante para el óptimo funcionamiento cuando existe alto volumen operativo, tornándose crítico en algunos sistemas.

- **Seguridad:** Factor Importante incluye diversas técnicas, aplicaciones y dispositivos encargados de asegurar la integridad y privacidad de la información de un sistema informático y sus usuarios.
- **Independencia a la disponibilidad de Red:** Continuidad de la operatividad del sistema aun cuando no este disponible la Red.
- **Flexibilidad:** Capacidad de ser un sistema configurable y extensible.
- **Interoperatividad:** Es la condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos.
- **Reusabilidad de Software:** Reducir costes de desarrollo y mantenimiento en la implantación y gestión del proyecto, utilizando módulos existentes.

3.3. Elección de la Tecnología:

Estos indicadores expuestos en el punto anterior, fueron ponderados según la importancia en el contexto del problema planteado. Una vez evaluados estos indicadores para cada tecnología se obtuvieron los siguientes resultados:

FACTORES	PESO EXTENDIDO (0.0 - 1.0)	MIDDLEWARE		WEB SERVICES		MULTIAGENTES	
		Puntaje (0 al 3)	Peso Extendido	Puntaje (0 al 3)	Peso Extendido	Puntaje (0 al 3)	Peso Extendido
TRANSPARENCIA DE ACCESO	1	3	3	3	3	3	3
CONCURRENCIA	1	3	3	3	3	3	3
ESCALABILIDAD	0.7	2	1.4	1	0.7	3	2.1
FACILIDAD DE IMPLEMENTACIÓN	0.3	1	0.3	2	0.6	2	0.6
FUNCIONAMIENTO INDEPENDIENTE DEL ANCHO DE BANDA	1	1	1	1	1	3	3
SEGURIDAD	1	2	2	3	3	2	2
INDEPENDENCIA A LA DISPONIBILIDAD DE RED	1	1	1	1	1	3	3
FLEXIBILIDAD	1	2	2	2	2	2	2
NIVEL DE INTEROPERATIVIDAD	1	2	2	3	3	3	3
REUSABILIDAD DEL SOFTWARE	1	3	3	3	3	3	3
TOTAL		20	18.7	22	20.3	27	24.7

Tabla 3: Comparacion de Tecnologias Propuestas para el SICIP

Del resultado de la comparativa anterior podemos concluir que la Tecnología de Agentes Móviles resulta más óptima para la solución del problema. Para el desarrollo con Agentes Móviles es necesario seleccionar el método y las herramientas necesarias que nos ayuden a diseñar e implementar la solución, por ello se ha analizado las diferentes alternativas de metodologías más conocidas.

3.4. Evaluación de las Metodologías de Solución Existentes¹⁹

La elección de las metodologías en base al análisis de las características que la metodología de agentes debe cumplir. A continuación se expone brevemente las ventajas y desventajas de las metodologías existentes:

Vowel Engineering

Vowel Engineering ha sido una de las primeras metodologías en modelar sistemas utilizando diferentes vistas. Aunque es prometedor, el trabajo en *Vowel Engineering* está incompleto ya que no termina de estabilizarse con herramientas de soporte. Además, no existen instrucciones acerca de cómo describir cada uno de los aspectos considerados.

MAS-CommonKADS

La especificación de SMA que proporciona MAS-CommonKADS detalla la mayoría de aspectos en lenguaje natural. Esta particularidad dificulta el análisis automático de la especificación generada y supone una gran

¹⁹ Tomado de la Referencia [20]

desventaja frente a semi-formalismos como UML, soportado por muchas herramientas y con la posibilidad de hacer chequeos para verificar el desarrollo (¿Existen elementos no utilizados? ¿Se ha asociado especificaciones de comportamiento a los casos de uso?). Para lograr lo mismo en MAS-CommonKADS habría que restringir el uso de lenguaje natural o bien incluir formalismos que logren una definición más precisa y menos ambigua del SMA.

En INGENIAS, el problema se salva utilizando meta-modelos como mecanismo de especificación. El desarrollo de meta-modelos está soportado por herramientas que permite el procesamiento automático de los modelos generados. Los meta-modelos en algunos casos profundizan más en el detalle que MAS-CommonKADS. Tal es el caso del meta-modelo de organización, el de tareas y objetivos, o el de agentes

BDI

En INGENIAS, los principios del BDI se hallan presentes de diferentes formas. Los planes, entendidos como secuenciación de tareas, aparecen como flujos de trabajo. Los objetivos han sido introducidos explícitamente, incluyendo su refinamiento en subobjetivos, el establecimiento de dependencias entre objetivos (inicialmente árboles Y/O) y dependencias con tareas. Su uso se ha extendido a otros modelos como el de organización y el de interacciones, vertebrando la ejecución de tareas e iniciación de interacciones a lo largo de toda la metodología. Por lo tanto, el grado de integración del modelo BDI en la metodología es superior al

logrado en la propuesta de [Kinny y Georgeff]. Y todo ello dentro de un ciclo de vida de software complejo, como es el RUP. Además, se ha logrado la independencia total de implementación dotando a la metodología de un procedimiento para parametrizar con los modelos generados armazones software escritos en cualquier lenguaje.

MASE

La herramienta de soporte permite generar código automáticamente a partir de la especificación. La generación de código es independiente del lenguaje de programación utilizado, ya que se realiza recorriendo las estructuras de datos generadas en la especificación y generando texto como salida. No obstante, el proceso de generación de código es mejorable, ya que el código de los componentes a generar está entremezclado con el código que lee la especificación.

En INGENIAS, se persigue un proceso de desarrollo similar al seguido mediante otras herramientas, como Rational Rose, TogetherJ o Paradigm+. En estas herramientas, el usuario tiene libertad para elaborar diagramas incompletos o generar sus propias vistas del sistema, tomando elementos de diferentes modelos e incorporándolos a un nuevo diagrama. También se intenta separar de UML en el sentido de no repetir las mismas soluciones. La forma en que se tratan las interacciones en INGENIAS es un buen ejemplo de ello. Las interacciones buscan generalizar diferentes alternativas existentes, como los MSC, diagramas de secuencia o colaboración, y diagramas de protocolo (AUML). Aparte de la

generalización, las interacciones se integran completamente dentro de otros modelos como un elemento más. De hecho, la iniciación de interacciones se representa como el producto de ejecutar una tarea.

Otro aspecto tenido en cuenta en INGENIAS es el proceso de generación de los modelos, esto es, qué actividades están involucradas en su producción. El resultado es un conjunto estructurado de actividades detalladas que guiarán a futuros usuarios de la metodología en su utilización. Las actividades se enmarcan en diferentes flujos de trabajo para cada modelo, con lo que se facilita una futura integración de la metodología en entornos automatizados de gestión de proyectos software como el Entorno de Creación de Servicios.

ZEUS

El ámbito de la metodología se limita a estudiar cómo agrupar la funcionalidad del sistema dentro de cada rol, dejando aparte consideraciones acerca de cómo organizar las tareas, definir las ontologías y las dependencias sociales, aspectos que son modelables dentro de la herramienta.

La cuestión que se plantea al estudiar ZEUS y MaSE es si hay que tener una buena herramienta o una buena metodología. La postura de INGENIAS es que las herramientas de soporte no tienen que condicionar la metodología y que de hecho han de ser independientes. La independencia en este trabajo se consigue usando meta-modelos como elemento de

construcción. Ello facilita el portar la metodología a diferentes herramientas, ya que cualquier herramienta que soporte meta-modelado podría servir como herramienta soporte de desarrollo.

Otra ventaja de los meta-modelos es que facilitan la evolución de la metodología. Tanto ZEUS como MaSE tendrían que cambiar en gran medida sus herramientas asociadas, para, por ejemplo, incluir el meta-modelo de organización de este trabajo. Sin embargo, el paso inverso, incluir elementos de MaSE o ZEUS en meta-modelos, no supone un gran esfuerzo

GAIA

En INGENIAS las dependencias entre los distintos meta-modelos se revisan independientemente. La mayoría se refiere a que al introducir ciertas entidades, hay que definir aspectos adicionales en otros modelos. Por ejemplo, al crear un objetivo, siempre hay que asociar una tarea o tareas que permiten alcanzarlo. Si el objetivo se identifica dentro de un modelo de agente, entonces, se necesita que en un modelo de objetivos y tareas se indique qué tarea o tareas deben ejecutarse.

Para terminar, el modelo de organización en GAIA es superficial ya que no se tienen en cuenta las relaciones estructurales. En la extensión de GAIA comentada en [Zambonelly, Wooldridge M. y Jennings N.R.] y supuestamente dedicada a cubrir este hueco, se habla más de restricciones sociales respecto a uso de roles que de la organización en sí.

En INGENIAS el meta-modelo de organización se ve respecto del SMA como el equivalente a la arquitectura del sistema de un sistema convencional. Sirve para definir a alto nivel cómo se organizan los elementos del sistema para hacer posible los objetivos comunes a los agentes que participan en la organización.

MESSAGE

En INGENIAS, se plantea la evolución a lo largo del ciclo de vida del software de los modelos generados. El paso de una etapa a otra está marcado por el nivel de detalle alcanzado en cada modelo. Así, las interacciones inicialmente pueden detallarse con diagramas de colaboración para luego concretarse en el diseño con otros tipos de diagramas que alcancen más detalle en aspectos como la motivación de la interacción o actos del habla empleados durante el proceso. El paso a implementación, como se ha comentado antes, se ha generalizado en forma de proceso de parametrización de armazones software. Esta forma de implementación es una evolución del trabajo de MESSAGE, donde se proponían arquitecturas y componentes adecuados para esta tarea.

Los meta-modelos en general se han modificado para integrar resultados de investigación tales como planificación de tareas, el modelo BDI, la estructuración de elementos de la comunidad o el uso de tareas. De forma similar a MAS-CommonKADS se ha estudiado el dominio de aplicación de cada meta-modelo para que se puedan aplicar los resultados correspondientes.

También se ha incluido un nuevo meta-modelo, el de entorno. MESSAGE no tenía en cuenta lo que rodeaba la aplicación, por lo cual la inclusión de elementos como servicios del sistema, recursos o aplicaciones que no fueran agentes, eran difíciles de tratar. En INGENIAS, el meta-modelo de entorno permite incluir este tipo de elementos de forma coherente. De hecho, la percepción de los agentes se expresa en función de estos elementos. Así, se puede representar que un agente de interfaz se conecte a una aplicación existente.

	Ingenias	MascommonKads	Zeus	GAIA	Message
ANALISIS					
Completitud	⊗⊗⊗	⊗⊗	⊗⊗	⊗	
Aplicabilidad	⊗⊗	⊗⊗⊗	⊗⊗⊗	⊗	
Complejidad	⊗⊗⊗	⊗⊗	⊗⊗	⊗	
Reusabilidad	⊗⊗	⊗	⊗	⊗	
DISEÑO					
Completitud	⊗⊗⊗	⊗⊗	⊗⊗	⊗⊗	
Aplicabilidad	⊗⊗⊗	⊗⊗	⊗⊗⊗	⊗⊗	
Complejidad	⊗⊗⊗	⊗	⊗⊗	⊗⊗⊗	
Reusabilidad	⊗⊗	⊗⊗	⊗⊗	⊗⊗	
DESARROLLO					
Completitud	⊗⊗⊗	⊗⊗		⊗⊗	⊗⊗⊗
Aplicabilidad	⊗⊗⊗	⊗⊗⊗		⊗⊗	⊗⊗⊗
Complejidad	⊗⊗⊗	⊗⊗		⊗⊗⊗	⊗⊗
Reusabilidad	⊗⊗	⊗⊗⊗		⊗⊗	⊗⊗⊗
DESPLIEGUE					
Completitud	⊗⊗⊗			⊗⊗	⊗⊗⊗
Aplicabilidad	⊗⊗			⊗⊗	⊗⊗⊗
Complejidad	⊗⊗⊗			⊗⊗⊗	⊗⊗
Reusabilidad	⊗			⊗⊗	⊗
OTROS CRITERIOS					
Disponibilidad	⊗⊗⊗	⊗⊗⊗	⊗⊗	⊗⊗⊗	⊗⊗⊗
Soporte	⊗⊗	⊗⊗	⊗	⊗⊗⊗	⊗⊗⊗

-Nada ⊗ -Poco ⊗⊗ -Regular ⊗⊗⊗ -Bueno ⊗⊗⊗

Tabla 4 : Comparación de las Metodologías de Agentes

3.5. Elección de la Metodología a usar

A un nivel más fino, podríamos explorar aspectos más técnicos, tales como la manipulación de procesos, movilidad, seguridad, estandarización, etc. pero limitaremos nuestro análisis al nivel más conceptual. En Conclusión general:

Si hubiera que elegir una metodología, ¿cuál sería la ganadora?

La respuesta es que cada metodología, por el bagaje de sus creadores, se especializa en áreas concretas. Si se está acostumbrado a trabajar con sistemas basados en conocimiento como con la metodología CommonKADS, lo lógico es que se elija MASCommonKADS. Si por el contrario está interesado en un enfoque más orientado a agentes, puede seleccionar ZEUS, INGENIAS. Y si lo que se quiere es tener un soporte de herramientas, la lista de metodologías válidas se: ZEUS, INGENIAS.

En los casos en que se requiera un proceso de desarrollo robusto, detallado y ensayado en desarrollos reales, la recomendación sería MASCommonKADS o INGENIAS. Otra opción que tienen los desarrolladores es quedarse con lo que más les interese de cada metodología. Una recomendación sería utilizar la técnica de modelado de INGENIAS junto con su herramienta de especificación e implementar con JADE. Las metodologías estudiadas son representantes de diferentes líneas de investigación en agentes: Agentes como sistemas basados en conocimiento (MAS-CommonKADS), agentes BDI (BDI, ZEUS), agentes según la definición de Newell (INGENIAS). La aplicación de estas metodologías depende mucho de la formación que tenga su usuario final.

Esto quiere decir que son interpretadas en función de los intereses de quien las aplica. De momento, no hay soluciones que fuercen un uso concreto. Es de destacar que la mayoría de las metodologías expuestas carecen de ejemplos detallados y suficientemente extensos que expliquen cómo se aplica. La presentación de una metodología debería de acompañarse con desarrollos que la avalen y muestren su utilidad.

Finalmente, de la revisión se concluye que la metodología INGENIAS presenta muchas ventajas para la implementación de Sistemas basados en agentes móviles frente a otras metodologías específicas.

CAPITULO IV

EL DESARROLLO

4.1 Implementación

En este apartado se comentan algunas decisiones que se tomarán durante la implementación del sistema y que pueden ayudar a su mejor comprensión. Principalmente se abordan cuestiones sobre la plataforma JADE-LEAP.

4.1.1 Agente Interfaz:

Dentro de la plataforma JADE-LEAP los agentes son subclases de la clase Agent. El agente interfaz hereda de la clase GuiAgent proporcionada por la plataforma JADE-LEAP lo cual le permite tener su propia interfaz gráfica.

Se implementan dos tipos de interfaces gráficas distintas. Una basada en las librerías swing, para ser usada en dispositivos que usen una máquina virtual de Java 1.2 y otra basada en las librerías AWT para usarse en dispositivos con una

máquina virtual de Java con las restricciones de la especificación de PersonalJava o la más moderna Java 2 Micro Edition (J2ME) en su versión CDC.

Este agente cuando recibe una petición de ejecución de alguna acción que requiere de la colaboración con otros agentes añade a su lista de comportamientos el adecuado para dar respuesta a la petición. Estos comportamientos gestionan las comunicaciones con el agente encargado de hacer el trabajo solicitándole que ejecute la acción adecuada y procesando finalmente la respuesta obtenida. Estas comunicaciones se ajustan al estándar FIPA [6]

4.1.2 Agentes Cartero y Buscador Documentos:

Estos dos agentes tienen un funcionamiento similar al agente interfaz. Al iniciarse añaden un comportamiento cíclico (CyclicBehaviour) que espera la llegada de mensajes. Cuando recibe un mensaje si lo entiende responde con un mensaje Agree y añade a la lista de los comportamientos del agente el comportamiento que ejecuta la acción solicitada en el mensaje.

Estos comportamientos que se encargan de ejecutar las acciones son comportamientos secuenciales (SequentialBehaviour) formados por subcomportamientos que se ejecutan de forma secuencial siguiendo el siguiente esquema: primero ejecutan un comportamiento que se encarga de trasladar el agente del dispositivo del usuario al dispositivo en donde está el Contenedor Principal de la plataforma de agentes (contenedor en donde están instalados los servicios de la plataforma). Después ejecuta el comportamiento que se encarga de

obtener la respuesta a la acción solicitada. Una vez que tiene la respuesta vuelve a ejecutar el comportamiento de migración para regresar al dispositivo del usuario y finalmente le envía la respuesta de la acción al agente Interfaz.

4.2 Herramientas y tecnologías utilizadas

Para documentar las fases de análisis y diseño se ha utilizado la herramienta INGENIAS Editor proporcionada por los creadores de la metodología INGENIAS y que permite la definición de los modelos de dicha metodología.

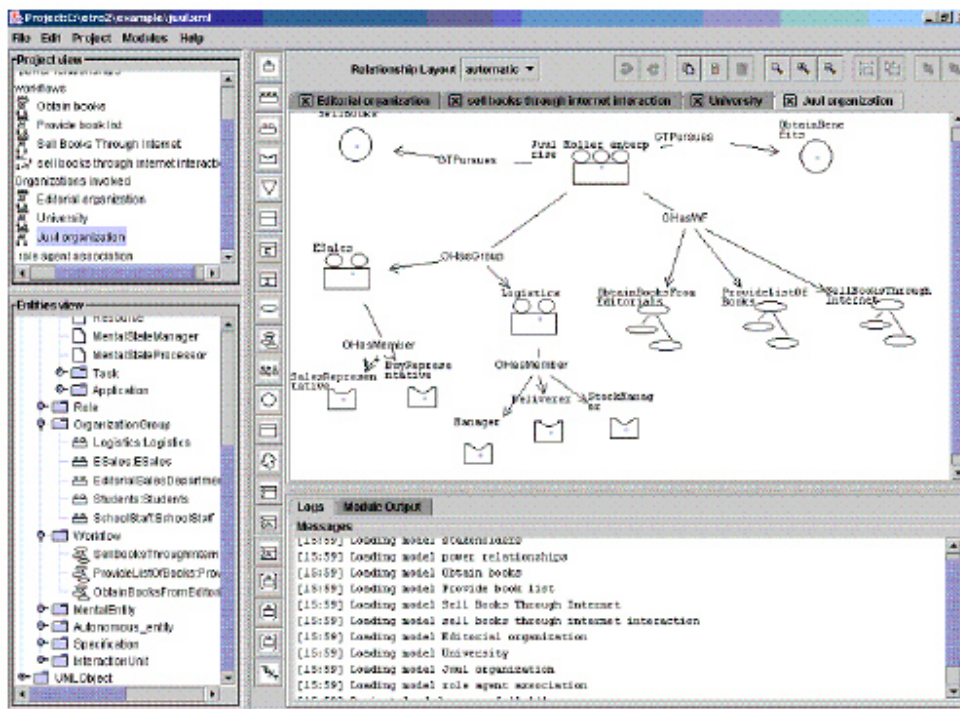


Figura 19 : Editor de Ingenias

- La implementación se llevaría a cabo con NetBeans IDE 3.5.1, interfaz de desarrollo de aplicaciones Java.

4.3 Mecanismos de seguridad

Para la solución propuesta se ha realizado una evaluación de los diferentes niveles de seguridad que debería contemplar en el Sistema Integrado de Consulta en este marco se plantea:

Control de seguridad local del usuario y usuario General de Acceso:

Cada Cuenta de Correo se registrará por primera y única vez al momento de realizar la configuración de cada cuenta, asimismo deberá contar con un control de Acceso por usuario para el aplicativo móvil, el cual deberá brindar un servicio de Validación de usuario que será utilizado por el SICIP.

Para la implementación de Sistemas haciendo uso de tecnología de agentes móviles existen técnicas de seguridad propuestas, aunque aún son muy escasas las investigaciones realizadas en este aspecto.

En la presente tesina se ha evaluado las diferentes técnicas, a continuación planteamos las más importantes a considerar:

Prevención de Ataques de Host maliciosos.

Dentro de las propuestas existentes para prevención de Ataques de Host maliciosos seleccionamos el Modelo Basado en Confianza en, el cual se limita la ejecución de los agentes a aquellas maquinas de Confianza, que estarán parametrizadas en el servidor principal del Sistema.

Modelo Basado en la Confianza:

Es este modelo se limita la ejecución de los agentes sólo a aquellas maquinas que consideramos de confianza. Esto es, máquinas de las cuales no se espera ningún tipo de actividad anómala o maliciosa e Incluso se puede pensar en cierto control social de los agentes y establecer unos “veremos” de “reputación” para los hosts.

4.4. Arquitectura de SICIP (Sistema de Información para la consulta de información personal).

En la arquitectura presentada especificamos tres tipos de agentes cuyo objetivo será la de gestión, búsqueda y presentación de información de correos electrónicos. Así tenemos:

- ✓ Al Agente Interfaz
- ✓ El Agente Buscador
- ✓ El Agente Emisario
- ✓

SICIP sigue una arquitectura basada en agentes móviles de la que forman parte los agentes Interfaz, Buscador y Emisario. Estos agentes se inician en el Dispositivo del Usuario cuando se arranca el sistema. El agente Interfaz es un agente estático que se encarga de gestionar las comunicaciones con el usuario mediante una interfaz gráfica con la que interactúa. Los otros dos agentes son agentes móviles que se trasladan a un dispositivo distinto del usuario (Servidor) para ejecutar en éste las acciones que les que ordene el agente Interfaz. Una vez se desplazan al Dispositivo Servidor ejecutan las acciones que les fueron

solicitadas haciendo uso de la conexión a Internet de banda ancha ofrecida como servicio a agentes móviles así como la utilización de los servicios necesarios. Una vez obtienen la respuesta a la acción regresan al Dispositivo del Usuario y le comunican al agente interfaz esa respuesta. El agente móvil Emisario realiza todas las funciones de manejo de correo electrónico mientras que el agente móvil Buscador lleva a cabo las operaciones sobre documentos (por ejemplo, búsqueda o descarga de documentos, etc.).

ARQUITECTURA DE SICIP

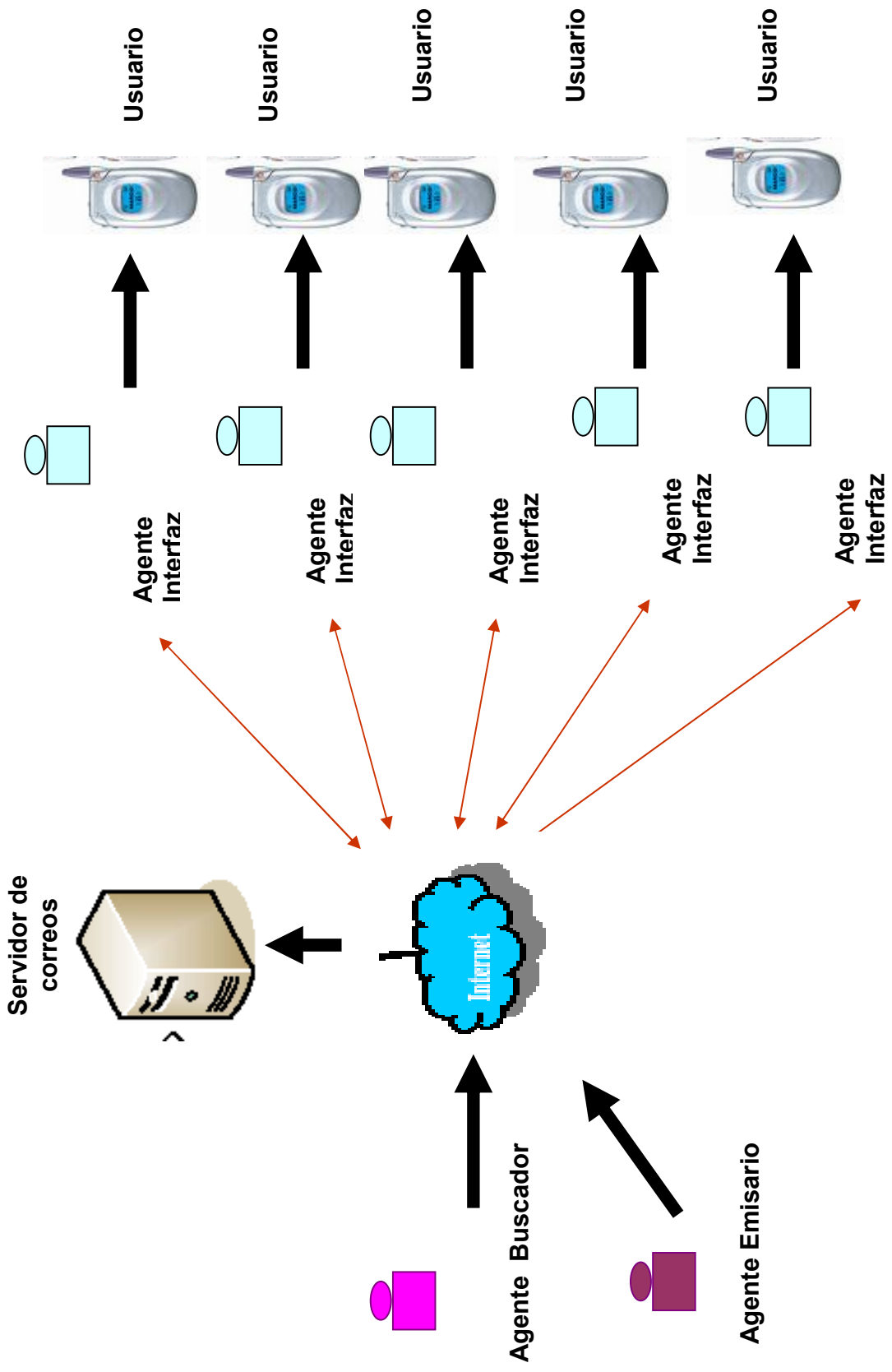


Figura 20 : Arquitectura del Sistema SICIP

4.5 Análisis

Se empieza identificando los casos de uso más importantes que reflejan los problemas principales que se van a encontrar y cuáles van a ser los componentes del sistema que participarán en su resolución.

Las funcionalidades más relevantes del sistema para la Consulta de Información Personal son:

- Gestionar Cuentas de Correo, se refiere a la configuración y registro de las cuentas de correo a fin de precisar que tipo de mensajes son los más relevantes.
- Borrar mensajes de correo, luego de identificar los correos que no son trascendentes para el usuario serán eliminados.
- Leer correo, se refiere a mostrar el contenido del mensaje.
- Obtener Documentos, se refiere a mostrar el contenido de los documentos identificados como relevantes para el usuario.
- Buscar Documentos, se refiere a filtrar aquellos documentos que son identificados como relevantes para el usuario.

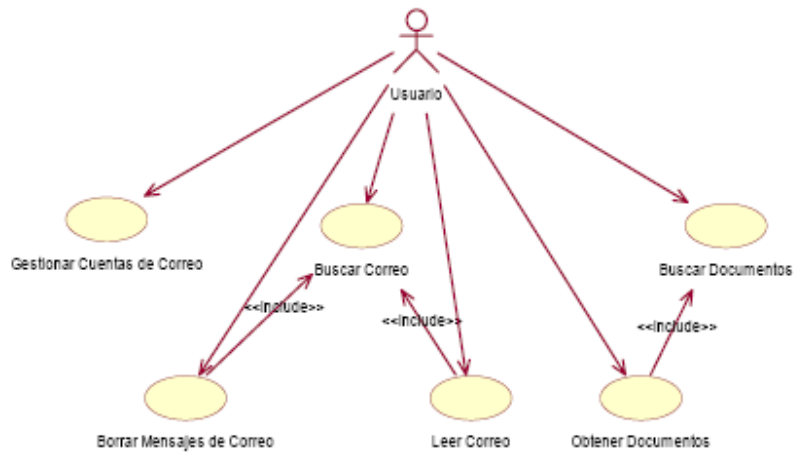















Figura 21 - Casos de uso

El método de desarrollo de SMA propuesto en INGENIAS concibe el SMA como la representación computacional de un conjunto de modelos. Cada uno de estos modelos muestra una visión parcial del SMA: los agentes que lo componen, las interacciones que existen entre ellos, cómo se organizan para proporcionar la funcionalidad del sistema, qué información es relevante en el dominio y cómo es el entorno en el que se ubica el sistema a desarrollar.

Se define a continuación un resumen de la nomenclatura necesaria para el desarrollo de la metodología INGENIAS.

Tipo	Notación	Descripción
Objetivo		<p>Los objetivos tienen asociado un estado mediante la propiedad estados posibles, que es una lista de constantes que denotan los estados por los que pasa. Los objetivos se satisfacen con la presencia de ciertas evidencias y pueden darse por fracasados cuando estas evidencias no se han producido. Hasta llegar a un objetivo satisfecho o fracasado, se pueden recorrer varios caminos. Aquí se propone que un objetivo pueda estar pendiente, recogiendo evidencias, refinado, en proceso, satisfecho o fallido.</p>
Rol		<p>El rol es una abstracción de un conjunto de funciones, que puede tener estado y que para existir necesita de otra entidad no abstracta que lo desempeñe, en este caso el agente. De esta forma es posible aplicar los roles en ISOA para describir conjuntos de responsabilidades (las funciones asociadas), protocolos (gracias a que el rol tiene estado implícito), o permisos de actuación (cambiando los roles de un agente, se cambia su capacidad de actuación).</p> <p>Los roles aparecen en las interacciones y en las organizaciones. Los roles carecen de gestores y procesadores de estado mental, y, por lo tanto, de capacidad de toma de decisiones.</p>
Procesador de estado Mental		<p>Su propósito es la toma de decisiones en sí, el control del agente. Las decisiones se pueden producir de forma algorítmica, en función de las tareas a ejecutar y el beneficio a obtener, u obedecer a la consecución de objetivos eligiendo secuencia de tareas a ejecutar. La ventaja de esta separación entre gestor y procesador del estado mental es que se desacoplan los mecanismos que implementan la autonomía e inteligencia del agente de la conceptualización de agente.</p> <p>La ejecución de la tarea debe decidirla el procesador de estado mental.</p>
Gestor de estado		<p>Su propósito es desarrollar la evolución del estado mental mediante las operaciones de creación, destrucción, modificación y monitorización del</p>

mental		conocimiento del agente. Es responsable de mantener la coherencia del conocimiento almacenado y de hacerlo evolucionar.
Agente		Un agente es una entidad autónoma. Una Entidad Autónoma se caracteriza por tener propósitos y una identidad única. La entidad agente, por herencia de Entidad Autónoma, adquiere la capacidad de perseguir objetivos y, mediante las asociaciones con roles y tareas, de alcanzarlos. Un agente es una entidad autónoma que actúa en el nivel de conocimiento y que se basa en el principio de racionalidad. El principio de racionalidad dice que la entidad realiza sólo aquellas acciones que le llevan a satisfacer sus objetivos. El principio de racionalidad ve los objetivos como guía y justificante de las acciones del agente.
Grupo		Cada grupo contiene agentes, recursos, aplicaciones o roles. Los grupos son herramientas inestimables para la estructuración de la organización, ya que permiten al desarrollador aplicar divide-y-vencerás para facilitar el problema de la definición del sistema, y también para modelar estructuras organizativas existentes en el mundo real.
Organización		Los objetivos perseguidos por la organización son los objetivos comunes a los agentes que la componen y el motivo por el cual se han agrupado. No obstante, una organización no es un agente. La diferencia fundamental es que la organización no tiene capacidad de ejecutar tareas ni para tomar decisiones, son los agentes que la componen quienes se encargan de ello.
Flujo de Trabajo		El objetivo del flujo de trabajo es establecer cómo se asignan los recursos, qué pasos (tareas) son necesarios para la consecución de un objetivo, y quiénes son los responsables de ejecutarlas. Según el la WorkFlow Management Coalition (WfMC), un flujo de trabajo es la automatización de un proceso de negocio, en su totalidad o parcialmente, durante el cual los documentos, información o tareas son pasadas de un participante a otro, de acuerdo con un conjunto de reglas procedimentales. En el flujo de trabajo, se habla de actividades en lugar de tareas, aunque en este contexto se pueden emplear indistintamente.

Interacción		Las interacciones determinan el comportamiento de los agentes mostrando cuál es su reacción cuando actúan sobre ellos. Y cómo el comportamiento va a ser función de las objetivos de los agentes y las tareas a ejecutar, se puede concluir que existe un importante vínculo entre interacciones, objetivos y tareas.
Consulta de entidades autónomas		Ejecuta la acción según las condicionales existentes.
Unidad de interacción		Las unidades representan mensajes o protocolos, dependiendo del nivel de abstracción. La composición de unidades de interacción siguiendo un orden determinado lleva a la especificación de la ejecución de la interacción.
Tarea		Una tarea se puede ver como: transformaciones del estado global, respuestas a eventos, un proceso, una acción física o un comando. De estas acepciones, se han elegido dos, que son complementarias: tarea como transformadora del estado global y tarea como proceso. La primera es útil porque concibe la tarea como pre-condiciones y post-condiciones (el cómo son estas pre y post-condiciones se discutirá más tarde). Esto permite su incorporación en mecanismos de planificación y razonamiento. La segunda es más pragmática y acorde con la realidad final: que la tarea será un conjunto de instrucciones que han de ejecutarse.
Aplicación		Las aplicaciones además de servir como actuadores y sensores de los agentes, se utilizan para integrar software en el desarrollo del SMA.. Las aplicaciones pueden emplearse para modelar servicios pasivos, esto es, un conjunto de operaciones que no requiere la interacción con ningún agente y que son empleadas por varios agentes. Ejemplos de tales aplicaciones serían servicios de nombrado (Naming service), servicios de emparejamiento (matchmaking service) o gestión del ciclo de vida de agentes (Life cycle management). Estos servicios pueden existir previamente en el entorno o no. Para el segundo caso, se hablará de aplicaciones internas.


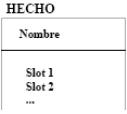

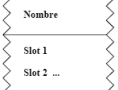
Recurso		<p>Por recurso se entiende toda aquel objeto del entorno que no proporciona una funcionalidad concreta, pero que es indispensable para la ejecución de tareas y cuyo uso se restringe a consumir o restituir. Cuando el uso sea más complejo, como la funcionalidad requerida de una base de datos, se empleará el término aplicación. Por último, la denominación de agente se emplea cuando la entidad satisfaga el principio de racionalidad.</p>
Hecho		<p>Información cierta para el agente. Los hechos reflejan información que es inherentemente cierta, como que “el agua se evapora al aplicarle calor” o bien información resultante de la ejecución de tareas.</p>
Creencia		<p>Las creencias reflejan información subjetiva en el sentido de que proviene de los elementos perceptivos del agente. Estas creencias son interpretaciones particulares hechas por el agente de lo que percibe. Las interpretaciones, pueden diferir por motivos técnicos como diseños diferentes de control de los agentes. Las creencias surgen de los eventos que simbolizan cambios ocurridos que han sido percibidos por el agente. Así ante el evento “comunicación con el agente X interrumpida” el agente puede elaborar la creencia “no quiere comunicarse conmigo” o bien “ha ocurrido un fallo en la red” o bien “ha ocurrido un fallo en el ordenador que alojaba al agente X”</p>
Evento		<p>Son los cambios ocurridos, en el mundo que el agente capta.</p>

Tabla 5: Nomenclatura de Metodología Ingenías.

4.6 Modelos establecidos por la metodología

4.6.1. Modelo de agente

El meta-modelo de agente se usa para describir de manera particular cada agente excluyendo las interacciones con otros agentes. Este meta-modelo se centra en la funcionalidad del agente y en el diseño de su control.

La definición del agente se divide en responsabilidades y comportamiento, el comportamiento del agente (tipo de control, especificación de estado mental y su evolución) y sus responsabilidades (asociación de tareas, objetivos y roles al agente).

Se ha identificado tres agentes para el desarrollo de la funcionalidad del sistema:

4.6.1.1 Agente Interfaz

Este agente cuando recibe una petición de ejecución de alguna acción que requiere de la colaboración con otros agentes añade a su lista de comportamientos el adecuado para dar respuesta a la petición. Estos comportamientos gestionan las comunicaciones con el agente encargado de hacer el trabajo solicitándole que ejecute la acción adecuada y procesando finalmente la respuesta obtenida. Estas comunicaciones se ajustan al estándar FIPA.

4.6.1.2 Agentes Cartero y Buscador Documentos:

Estos dos agentes tienen un funcionamiento similar al agente interfaz. Al iniciarse añaden un comportamiento cíclico (CyclicBehaviour) que espera la llegada de mensajes. Cuando recibe un mensaje si lo entiende responde con un mensaje Agree y añade a la lista de los comportamientos del agente el comportamiento que ejecuta la acción solicitada en el mensaje.

Estos comportamientos que se encargan de ejecutar las acciones son comportamientos secuenciales (SequentialBehaviour) formados por subcomportamientos que se ejecutan de forma secuencial siguiendo el siguiente esquema: primero ejecutan un comportamiento que se encarga de trasladar el agente del dispositivo del usuario al dispositivo en donde está el Contenedor Principal de la plataforma de agentes (contenedor en donde están instalados los servicios de la plataforma). Después ejecuta el comportamiento que se encarga de obtener la respuesta a la acción solicitada. Una vez que tiene la respuesta vuelve a ejecutar el comportamiento de migración para regresar al dispositivo del usuario y finalmente le envía la respuesta de la acción al agente Interfaz.

AGENTE INTERFAZ

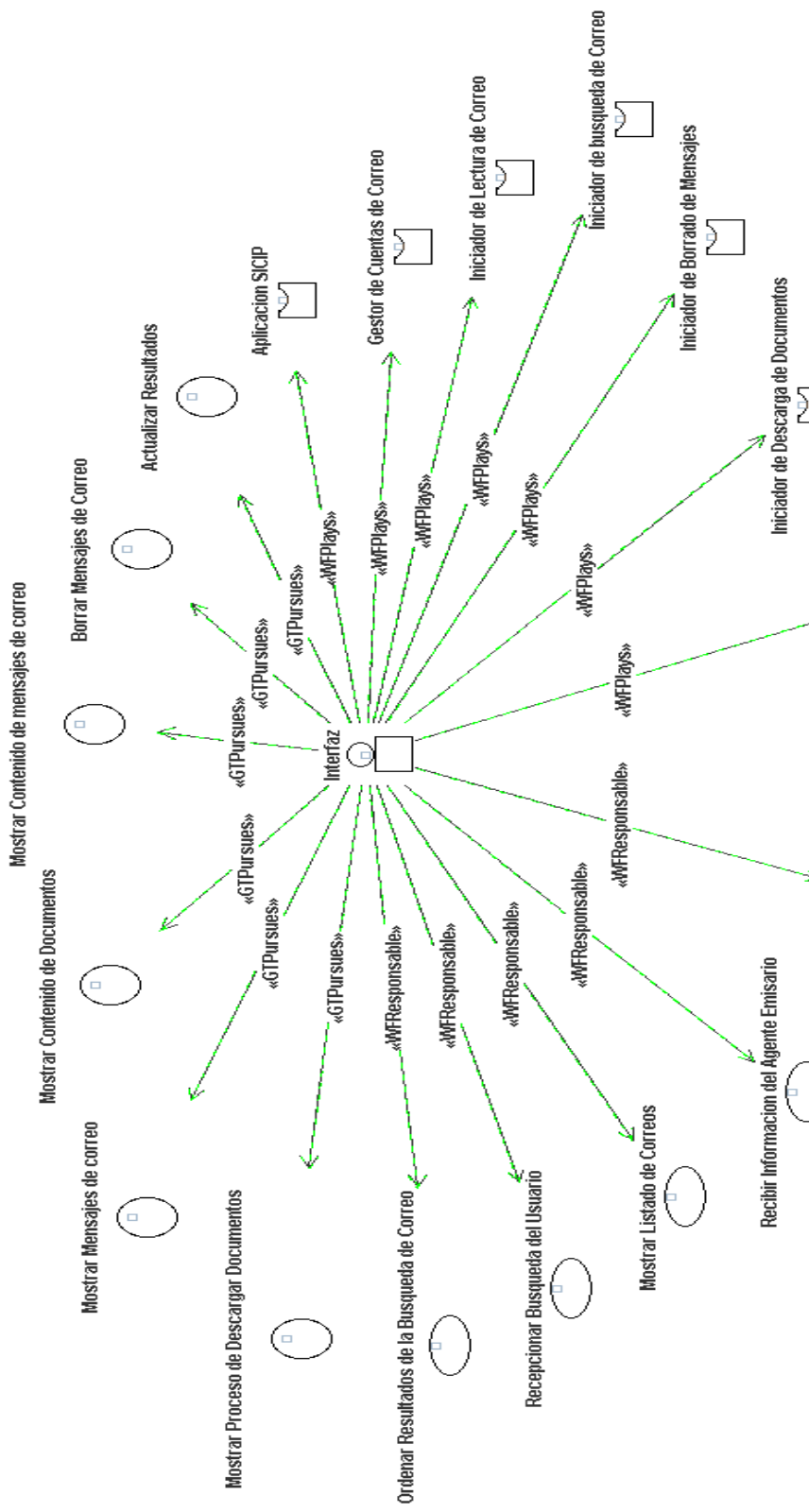


Figura 22 : Agente Interfaz

AGENTE EMISARIO

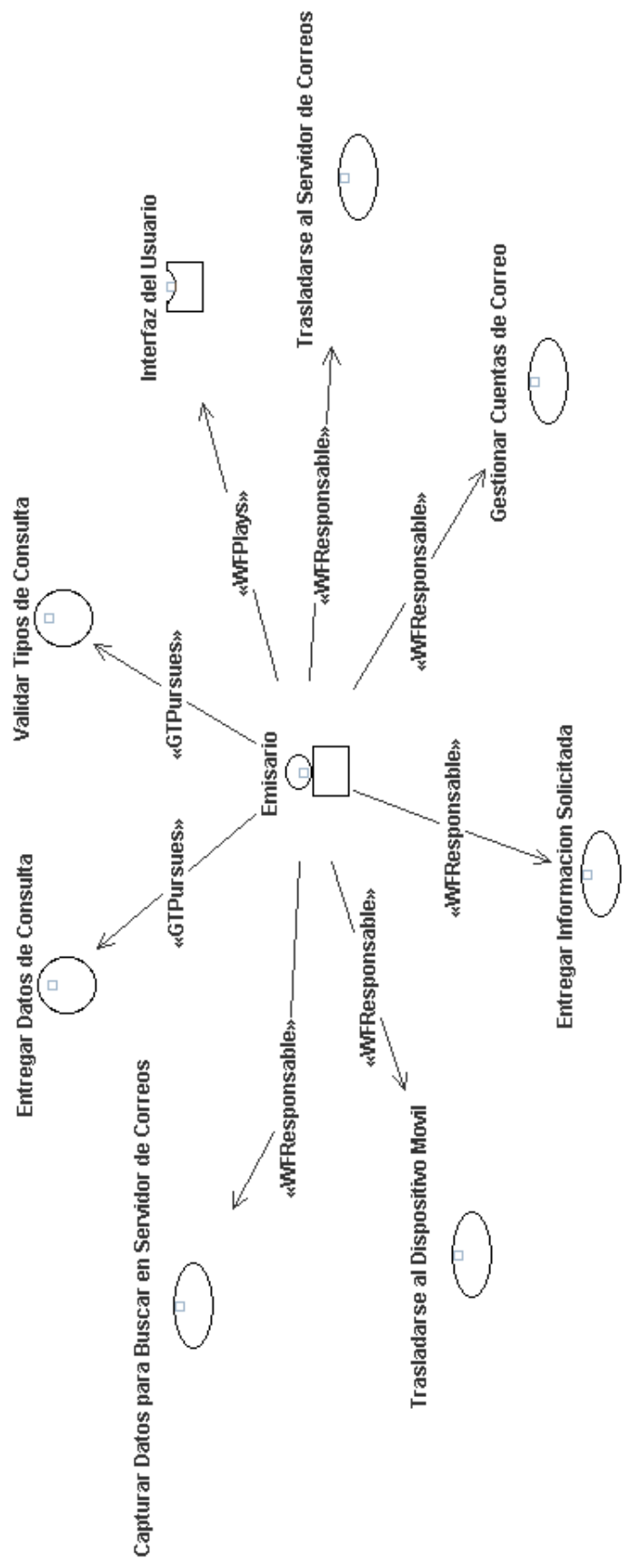


Figura 23 : Agente Emisario

AGENTE BUSCADOR

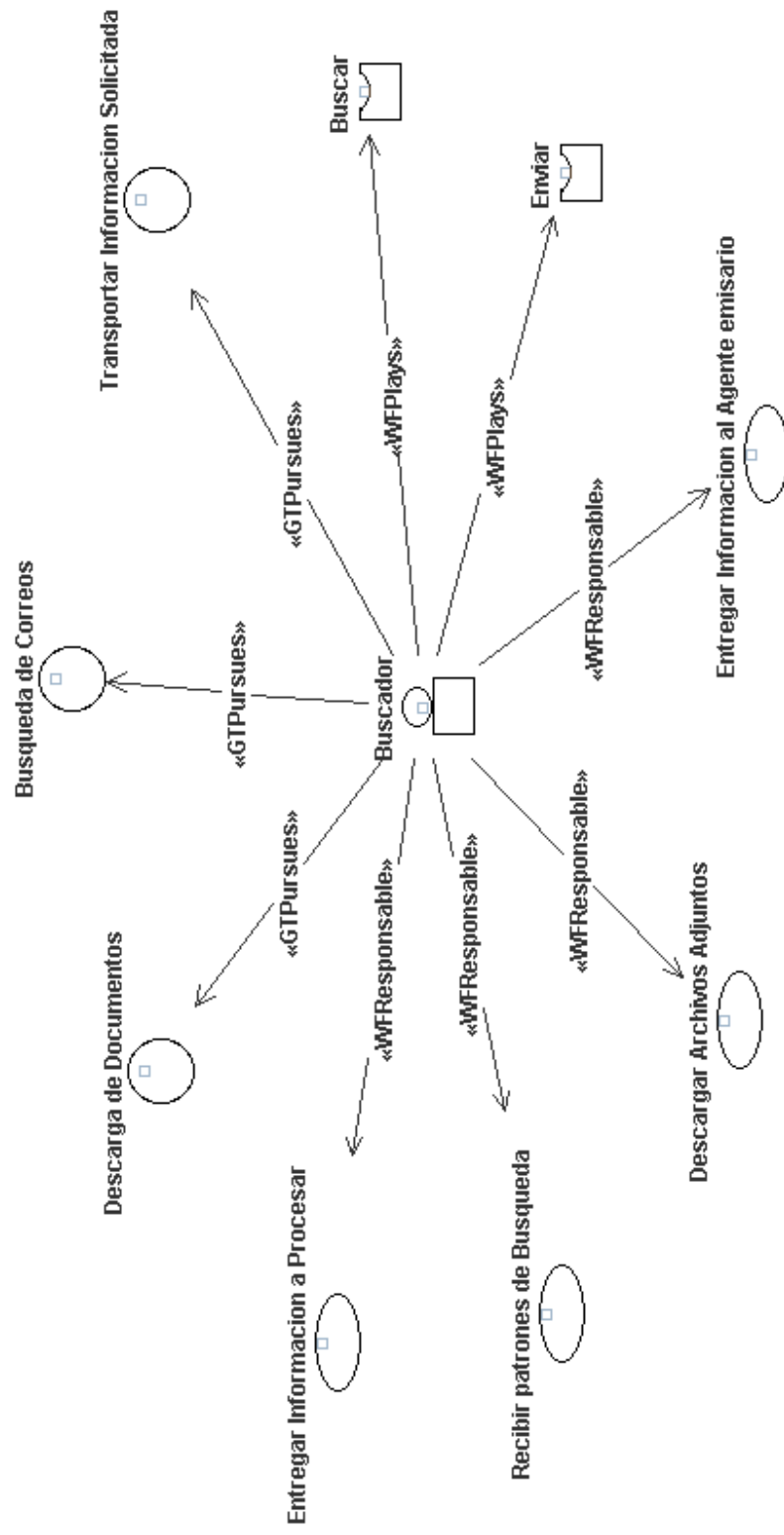


Figura 24 : Agente Buscador

MODELO DE INTERACCION EMISARIO – INTERFAZ

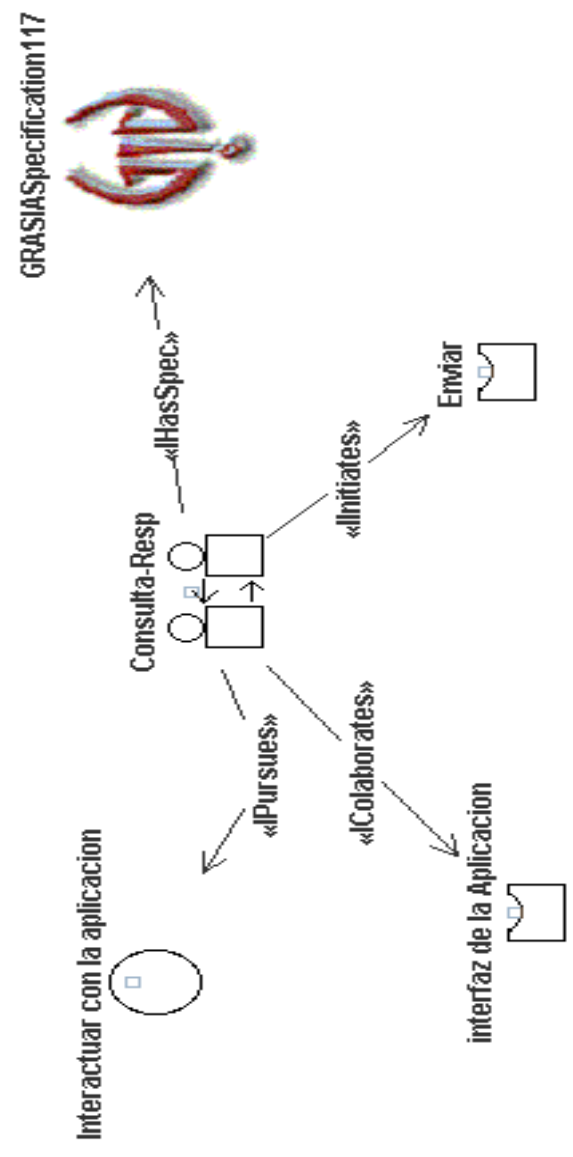


Figura 25 : Modelo de Interacción Emisario - Interfaz

MODELO DE ENTORNO

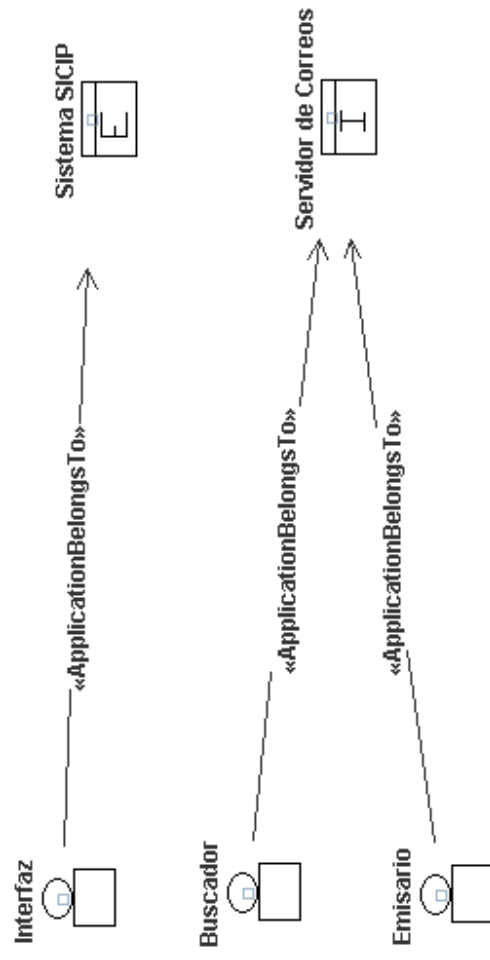


Figura 26 : Modelo de Entorno

MODELO DE ORGANIZACIÓN

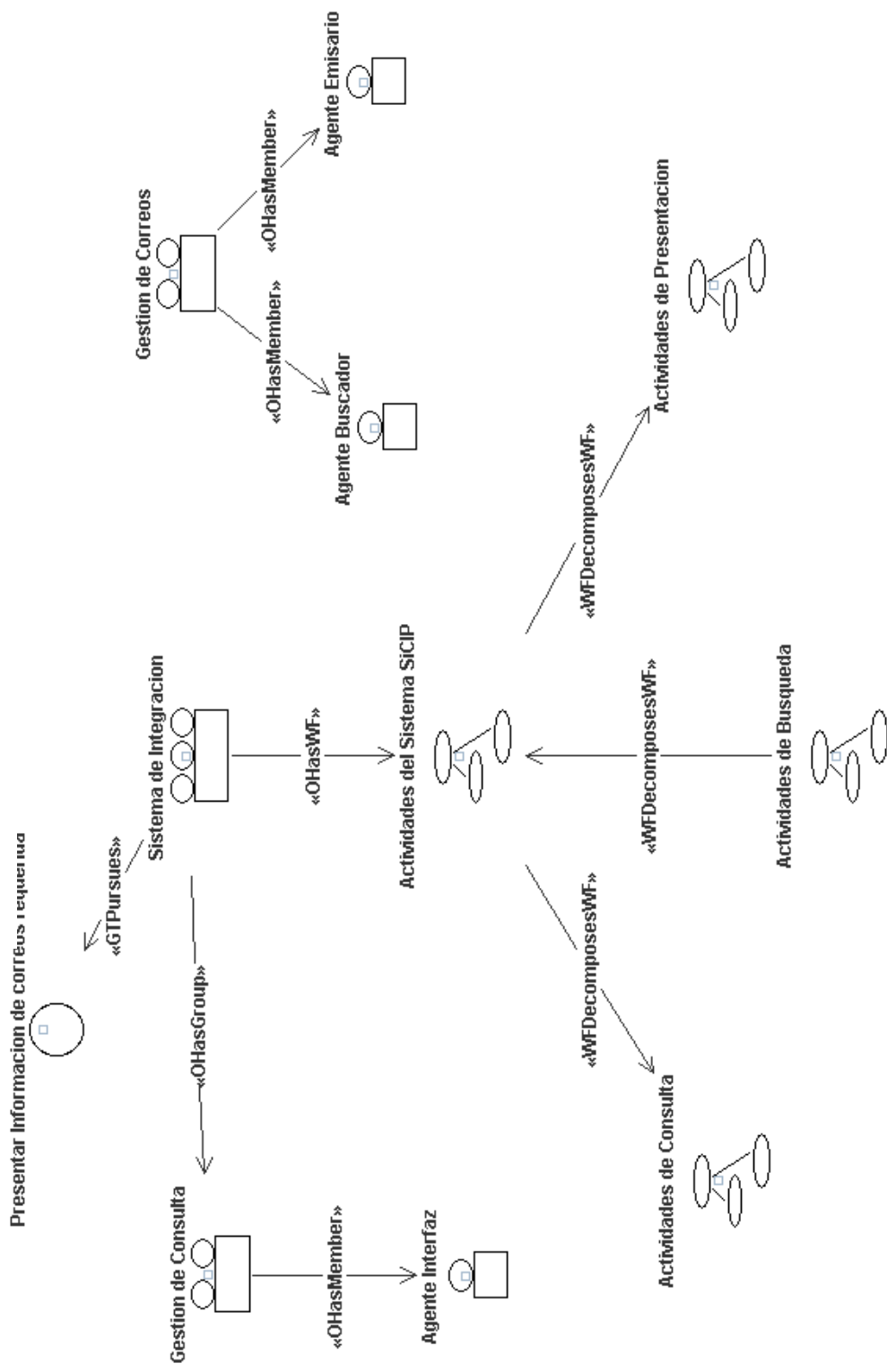


Figura 27 : Modelo de Organización

CAPITULO V

EVALUACION ECONOMICA

La implementación del SICIP como proyecto implica una evaluación económica que demuestre la factibilidad de su implantación por lo que se ha realizado un estudio económico sobre ello.

Desde un primer momento la empresa que adquiera el producto ya esta obteniendo un valor de retorno ya que podrá brindar un servicio de mejor calidad y sus clientes se sentirán más satisfechos al poder realizar consultas a su información personal de manera más rápida y eficiente pues tendrán información importante en el momento que lo necesiten y no tendrán que navegar por información irrelevante.

Se considera las siguientes fases en el desarrollo del aplicativo:

Proyecto SICIP:
1. Análisis de la Necesidad.
1.1. Estudio de Sistemas Actuales.
1.2. Análisis de Estructuras y Características.
2. Analizar Integración.
2.1. Estudiar Procesos.
2.2. Estudiar Datos.
2.3. Modelado con Metodología.
3. Diseñar Aplicación.

3.1. Diseño B.D.
3.2. Diseño Programas.
4. Implementación.
4.1. Construcción del esquema(Generación Estructural de código con IDK).
4.2. Codificación de los Programas.
5. Pruebas
5.1. Prueba de Unidades
5.2. Prueba del Sistema.
6. Implantación

Tabla 6 : Fases del Proyecto

Tiempo de desarrollo establecido de acuerdo a la necesidad del servicio:

FASE	TIEMPO(mes)	TIEMPO (Holgura)	TOTAL
Especificar necesidades	1	0.3	1.3
Analizar Integración	1.5	0.3	1.8
Diseñar Aplicación	1	0.3	1.3
Implementación	1	0.4	1.4
Pruebas	1	0.3	1.3
Implantación	0.5	0.3	0.8
Total	6	1.9	7.9

Tabla 7 : Tiempos por Fases

De la tabla 7, se deduce que el proyecto será entregado en un plazo de 8 meses.



Figura 28 : Planning de fases del proyecto

RECURSOS:**RECURSO DE TRABAJO****Cargas de personal:**

Equipo de Desarrollo

Gasto/mensual**S/.**

2 Analistas de Sistemas	8 000
3 Programadores Señor	7 500
2 Programadores Junior	3 000
1 Diseñador	2 000

Soporte al desarrollo

Especialista en redes locales y comunicaciones	2000
1 Tester	2000

Administrativos

Alta Dirección	3000
Operativo	1000

RECURSOS INMOBILIARIOS**Lugar de Trabajo**

Local Propio	000
---------------------	------------

Servicio de Terceros

Energía	300
Agua	300
Teléfono(4 anexos)	500
Servicios Varios	500

Cargas diversas de Gestión	500
-----------------------------------	-----

HARDWARE

Servidor , características propuestas: HP (Hewlett-Packard)	6 000
--	--------------

Tabla 8 : Tabla de Costos por Recursos

De la Tabla 9, obtenemos como Costo Total el monto de S/. 187 800.

Es de conocimiento que todo costo pierde su valor con el tiempo por lo que debe ser estimado teniendo en cuenta su devaluación, para ello es necesario la evaluación de este costo con un determinado método.

Existen diversos métodos o modelos de valoración económica. Se dividen básicamente entre métodos estáticos y métodos dinámicos. Dentro de los métodos dinámicos usaremos a uno de los métodos más usados y aceptados:

Valor Actual Neto. (V.A.N.).

$$VAN = - A + [FC1 / (1+r)^1] + [FC2 / (1+r)^2] + \dots + [FCn / (1+r)^n]$$

Si $VAN > 0$: El proyecto es rentable.

Si $VAN = 0$: El proyecto es postergado.

Si $VAN < 0$: El proyecto no es rentable.

Para la valoración a obtener con el VAN es necesario informar los diferentes campos:

A: desembolso inicial

FC: flujos de caja

n: número de meses (1,2,...,n)

r: tipo de interés ("la tasa de descuento")

$1/(1+r)^n$: factor de descuento para ese tipo de interés y ese número de años

Para el presente proyecto se tiene los siguientes valores:

Campo	Valor	Descripción
A: desembolso inicial	S/. 50000	Valor inicial, primer pago realizado por el contratista.
N, meses	8	
r	2%	

Tabla 10 : Tabla de Valores para calculo de VAN.

A continuación se realiza una especificación de los gastos, flujo de caja y valor VAN obtenido:

Flujos Pagos : Monto de egreso mensual.

Flujos Ingresos : Monto de ingresos

Flujo de Caja (FC) : Diferencia entre Ingresos y Egresos mensuales.

Flujo de Caja Actualizado : $FC(1 + r)^{-n}$

VAN : $\sum FC(1 + r)^{-n}$

Desarrollo del Aplicativo								
	1	2	3	4	5	6	7	8
Flujos Pagos		20100	14100	23600	26600	28600	25600	23600
Flujos Ingresos		50000			50000			100000
Flujo de Caja		29900	-14100	-23600	23400	-28600	-25600	76400
Acumulado F.C.		29900	15800	-7800	15600	-13000	-38600	12200
Flujo C. Actualiza.	2%	29313,73	-13552,48	-22238,81	21617,98	-25903,90	-22732,07	65206,66
Acumulado F.C.		29313,73	15761,25	-6477,56	15140,42	-10763,48	-33495,55	9424,78

Tabla 11: Tabla con cálculo de VAN

$VAN = 9\,424,78 > 0$ El proyecto es factible si el pago por el desarrollo es de S/. 200 000.

Si se realiza una evaluación con el costo total estimado se puede observar que en el plazo de 8 meses este monto estaría devaluado en S/. 987,81

Desarrollo del Aplicativo								
	1	2	3	4	5	6	7	8
Flujos Pagos		20100	14100	23600	26600	28600	25600	23600
Flujos Ingresos		50000			50000			87800
Flujo de Caja		29900	-14100	-23600	23400	-28600	-25600	64200
Acumulado F.C.		29900	15800	-7800	15600	-13000	-38600	0
Flujo C. Actualiza.	2%	29313,73	-13552,48	-22238,81	21617,98	-25903,90	-22732,07	54794,08
Acumulado F.C.		29313,73	15761,25	-6477,56	15140,42	-10763,48	-33495,55	-987,81

Tabla 12: Tabla con cálculo de VAN sin tener en cuenta costo de devaluación.

CONCLUSIONES

1. El desarrollo de aplicaciones basados en Sistemas de Informacion Personal (SIP) es un campo de desarrollo que esta creciendo de una manera importante y que se expresa en numero importante de investigaciones sobre este tema.
2. El interés de una solución integrada para la recolección de información proveniente de fuentes heterogeneas es importante en varias áreas de desarrollo debido a los nuevos conceptos de integración que se estan desarrollando.
3. Los sistemas multiagente es una de las formas de integración cuya implementacion se ha dado en forma masiva en los ultimos años con resultados prometedores y que permiten alcanzar soluciones concretas en los sistemas distribuidos como el propuesto.
4. Las metodologias para agentes cuentan como herramientas basadas en software libre que permiten manejar los costos de implementacion de forma apropiada.
5. El uso del Correo Electrónico es la aplicación más crítica de PIM, los desafíos se dirigen hacia procesar, capturar, guardar, recuperar y compartir

la información personal, para lo cual se ha explorado desde 2 enfoques técnicos – la centralización y la extracción de datos.

6. La búsqueda rápida y eficaz mejorará el acceso al e-mail y la información de contactos. Sin embargo, no resolverá la administración de tareas, ni reducirá el desorden en la bandeja de entrada, la manera que archiva, ni las carpetas que se hacen para algunos usuarios.
7. Los avances recientes en las aplicaciones de PIM permiten lograr múltiples ventajas, pero a su vez ellos introducen problemas de intrusividad y notificaciones no permitidas con el correo electrónico.

RECOMENDACIONES

1. Se necesita tener un mayor conocimiento sobre tecnologías de integración por lo que se recomienda desarrollar futuras investigaciones sobre este tema.
2. Trabajos futuros deberían considerar la utilización de conceptos sobre ontologías y una evolución a soluciones basadas en Web 2.0 que permiten un mayor control sobre las operaciones que tanto los clientes y servidores web realizan.
3. Es necesario incluir en la currícula de la carrera cursos sobre integración así como de las tecnologías que se usan en integración de sistemas.
4. Las nuevas aplicaciones de PIM deben permitir desarrollar habilidades mejoradas de sistemas para analizar el contenido del texto y desarrollar cada vez funciones de búsqueda más poderosas que probablemente lleven a algunos cambios profundos en los aspectos de administración de PIM.
5. Los investigadores y los desarrolladores deben entender las nuevas tendencias en el comportamiento de GIM para el diseño de los sistemas de información, pues hay varias formas de información personal que circulan a través de las redes de grupos cada vez más y de las instituciones del mundo.

REFERENCIAS BIBLIOGRAFICAS

[1] Ciancarini, P. and Wooldridge, M., 2001. Agent-Oriented Software Engineering. First International Workshop AOSE 2000, Lecture Notes in Computer Science Vol. 1957. Springer-Verlag, Berlin.

[2] DeLoach, S. A., et al, 2001. Multiagent systems engineering. In The International Journal of Software Engineering and Knowledge Engineering, Vol. 11, No. 3.

[3] Jennings, N. R., 1999. Agent-based computing: Promise and perils. In Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol2). Stockholm, Sweden, pp. 1429-1436.

[4] DeLoach, S. A. and Wood, M., 2001. Developing multiagent systems with agenttool. In Intelligent Agents VII. Agent Theories Architectures and Languages, 7th International Workshop (ATAL 2000), C. Castelfranchi, Y. Lesperance (Eds.). Lecture Notes in Computer Science. Vol. 1986, Springer Verlag, Berlin.

[5] Michael Wooldridge "An Introduction to Multiagent Systems".. Published by John Wiley & Sons, February 2002

[6] Lange, D. 1999, 'Seven Good Reasons for Using Mobile Agents', In Communication of ACM, vol. 42, no. 3, pp. 88-89.

[7] TEEVAN, Jaime; JONES, William and BEDERSON, Benjamin B.

“Personal Information Management”

Communications of the ACM, Volume 49 , Issue 1 (January 2006)

[8] WHITTAKER, Steve; BELLOTTI, Victoria; GWIZDKA, Jacek

“Email in Personal Information Management”

COMMUNICATIONS OF THE ACM January 2006/Vol. 49, No. 1

[9] ERICKSON, Thomas

From PIM to GIM: Personal Information Management in Group Contexts

Communications of the ACM, January 2006/Vol. 49, No. 1

[10] KELLY, Diane

Evaluating Personal Information Management - Behaviors and Tools

Communications of the ACM, January 2006/Vol. 49, No. 1

[11] MARSHALL, CATHERINE C.; JONES, William

Keeping Encountered Information

January 2006/Vol. 49, No. 1 COMMUNICATIONS OF THE ACM

[12] PÉREZ, María; MENDOZA, Luis E.; CARVAJAL, Yorika

“Orientaciones para la selección de Tecnologías de Integración de Sistemas de Software”

Universidad Simón Bolívar, Departamento de Procesos y Sistemas, Laboratorio de Investigación en Sistemas de Información (LISI). Venezuela, 2003.

[13] VICENTE PELECHANO, MARTA RUÍZ, JOAN J. FONS, PEDRO VALDERAS

“Desarrollo de Aplicaciones WEB basadas en Servicios WEB XML. Un Caso

Práctico

Departamento de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia. España 2006

[14] PEREZ DIAZ, Jesús Arturo

Tesis Doctoral “Sahara Arquitectura de Seguridad Integral para Sistemas de Agentes Móviles”

Departamento de Informática

Universidad de Oviedo, España, Enero del 2000

[15] CUTRELL, Edward; ROBBINS, Daniel C.; DUMAIS, Susan T.; SARIN, Raman

“Fast, Flexible Filtering with Phlat — Personal Search and Organization Made Easy”

CHI 2006 (Conference on Human Factors in Computing Systems). Proceedings • Personal Information Management, April 22-27, 2006 • Montréal, Québec, Canada

[16] SMALE, Stephanie and GREENBERG, Saul

“Transient Life: Collecting and sharing personal information”

OZCHI 2006. Annual Conference of the Australian Computer-Human Interaction Special Interest Group (CHISIG), 20 - 24 November 2006

Sydney, Australia

[17] Fabián Andrés Bustos, Juan Sebastián López, Vicente J. Julián

Empleo de tecnologías de agentes para la gestión de tutorías en un campus universitario

Dept. de Sistemas Informáticos y Computación, D.S.I.C.

Universidad Politécnica de Valencia. España, 2006

[18] ABETA, Akira; KAKIZAKI, Ken'ichi

Operation Record based Work Events Grouping Method for Personal Information Management System

Proceedings of Computer Software and Applications Conference, 1998.

COMPSAC Twenty-Second Annual International. Volume , Issue , 19-21 Aug

1998 Page(s):548 - 555

[19] Brian Subirana, profesor del IESE y Director del e-business Center

PwC&IESE

Movilidad: ¿dónde está el límite?

62 Anuario Computing España 2005

[20] PÉREZ ARDILA, Yanis Stanley

Tesis de Grado : “Aplicación de Metodologías Ingenias, Zeus, Masina al desarrollo de sistemas multi-agente, partiendo de SMA de subastas para la identificación de mejores practicas”

Departamento de Ingeniería Eléctrica, Electrónica, Sistemas y telecomunicaciones

Pamplona, Colombia. Abril, 2007

[21] Brow Nat, Kindel Charlie. "Distributed Component Object Model Protocol – DCOM/1.0" Microsoft Corporation. November, 1996

<http://www.microsoft.com/workshop/prog/com/>

[22] UMBC AgentWeb. "Software agents mailing list"

<http://www.es.umbe.edu/agentslist/archive/current/>

[23] Murugesan San. "Intelligent agents on the internet and web: progress and prospects" Proceedings of the international Workshop on Intellingent Agents on the Internet an Web. March 1998, Mexico. Pp 3-6

[24] Lawton George. Agents to roam the Internet. Sunworld Online Magazine. Octubre de 1996

<http://www.sun.com/sunwordldonline/swol-10-1996/swol-10-agent.html>

[25] Sommers Bret. Agents: Not Just for Bond anymore. JavaWorld- Agents Magazine. Abril1997. Pp 3-6

<http://www.javaworld.com/javaworld/jw-04-1997/jw-04-agents.html>

[26a] Perez Diaz Jesús Arturo, Cueva Lovelle Juan M. Lopez Perez Benjamín. "Different Platforms for movile agents development" Proceedings of the International Workshop on Intelligent Agents on the Internet and Web. Marzo 1998 Mexico.

Lista de Figuras

Figuras	Página
Figura 1 - Aproximación mediante el uso de RPC	64
Figura 2 – Nueva Aproximación mediante programación móvil	65
Figura 3: La máquina y los niveles de los protocolos	69
Figura 4 – Metodologías de Objetos Distribuidos	71
Figura 5: Incremento del número de usuarios en Internet	80
Figura 6: La interface Phlat	88
Figura 7: La interfase Phlat con un query de una clave simple y dos filtros	97
Figura 8 : Cuatro mecanismos que presenta la información personal	100
Figura 9: Sistema de Vida Transitoria	108
Figura 10: Un típico Mensajes de Hoy en multimedia generado con Vida Transitoria	113
Figura 11: Arquitectura del Sistema multi-agente	118
Figura 12: Esquema sistema ejecutándose a través de conexiones diferentes	119
Figura 13: Calendario de un PIM	123
Figura 14: Relación entre eventos y flujo de trabajo	124
Figura 15: Estructura de un flujo de trabajo	126
Figura 16: Sistema Propuesto	128
Figura 17: Diseño de Prototipos	136
Figura 18 - Distribución de Dispositivos	137
Figura 19 : Editor de Ingenias	154
Figura 20: Arquitectura del Sistema SICIP	158
Figura 21: Casos de uso	160
Figura 22: Agente Interfaz	167
Figura 23: Agente Emisario	168
Figura 24: Agente Buscador	169
Figura 25: Modelo Interaccion Emisario – Interfaz	170
Figura 26: Modelo de Entorno	171
Figura 27: Modelo de Organización	172
Figura 28: Planning de fases del proyecto	174

Lista de Tablas

Tabla	Página
Tabla 1 : Ventajas y Desventajas de los Middleware	43
Tabla 2 : Tabla de Principales Características de los Sistemas de Agentes	73
Tabla 3 : Comparacion de Tecnologías Propuestas para el SICIP	142
Tabla 4 : Comparación de las Metodologías de Agentes	149
Tabla 5: Nomenclatura de Metodología Ingenias	161
Tabla 6: Fases del Proyecto	173
Tabla 7: Tiempos por Fases	174
Tabla 8: Tabla de Costos por Recursos	176
Tabla 9: Tabla de Costos por Recursos especificado por meses	177
Tabla 10: Tabla de Valores para calculo de VAN	179
Tabla 11: Tabla con cálculo de VAN	180
Tabla 12: Tabla con cálculo de VAN sin tener en cuenta costo de devaluación	180