

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación



IDENTIDAD DIGITAL Y GENERACIÓN DE CONTEXTO COMO POTENCIADORES DE LOS SISTEMAS DE RECOMENDACIÓN

TRABAJO FIN DE MÁSTER

Iván Martínez Toro

2010

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos**

TRABAJO FIN DE MÁSTER

**IDENTIDAD DIGITAL Y GENERACIÓN DE
CONTEXTO COMO POTENCIADORES DE
LOS SISTEMAS DE RECOMENDACIÓN**

Autor
Iván Martínez Toro

Director
Joaquín Salvachúa Rodríguez

Departamento de Ingeniería de Sistemas Telemáticos

2010

Resumen

En la actualidad la cantidad de información presente en Internet es ingente y supone un verdadero problema encontrar el contenido buscado de una forma rápida y eficaz, teniendo en cuenta que las necesidades de cada usuario son diferentes y muy dependientes de su contexto.

Para solventar este problema surgieron los sistemas de recomendación que, a partir de unos datos de usuario y una serie de metodologías y algoritmos, son capaces de percibir las necesidades del usuario y filtrar grandes cantidades de información para acabar mostrando tan sólo la más relevante de una forma ordenada y coherente.

El origen de estos sistemas se remonta más de una década, pero su evolución ha sido imparable y las líneas de investigación para mejorarlos siguen abiertas, obteniendo cada vez mejores recomendaciones y necesitando menos datos para conseguirlas.

Una de las vías de mejora de mayor actualidad es la incorporación de un nuevo tipo de datos de entrada como es el contexto, tanto físico como social, de los usuarios. Para obtener este contexto de forma automática sin requerir interacción con los usuarios se pueden utilizar varios métodos, entre otros el uso de la tecnología de flujos de trabajo.

Esta tecnología de flujos de trabajo se está incorporando principalmente al mundo empresarial para automatizar y optimizar la ejecución de procesos de negocio. Con una extensión de la tecnología y aprovechando su potencial se pueden obtener datos de contexto social dentro de grupos de trabajo que pueden beneficiar a los usuarios y servir para mejorar sistemas como los motores de recomendación.

Por otro lado, al igual que existe en Internet una cantidad desmesurada de información, la red se está poblando de servicios cada vez más numerosos. Cada uno de estos servicios supone actualmente para los usuarios un registro con su consecuente par de usuario y contraseña (o cualquier otro tipo de credenciales) que genera una peor usabilidad y un rechazo a la hora de incorporar nuevos servicios a los ya utilizados habitualmente, entre otras cosas por no poder manejar tantas credenciales y percibir que se está poniendo en riesgo la seguridad. Para cambiar este paradigma ha surgido la idea de la identidad digital, entendida como una identidad única por cada usuario que le sirva para acceder a todos sus servicios. El principal desarrollo es OpenID, pero tiene una serie de problemas de seguridad que impiden su total expansión.

Abstract

Nowadays, the amount of information present on the Internet is huge and it is a real problem searching for the right content in a fast and optimized way, taking into account that every user has different needs depending on his context.

To solve this issue, recommender systems appeared; they can use some user's data with several methodologies and algorithms to find out the user's needs and filter the information to finally show the most relevant part of it ordered in a coherent way.

The origin of these systems happened more than a decade ago but their evolution has been constant and investigation lines to make them better are still active, obtaining more reliable recommendation systems which need less data to start working properly.

There is a concrete way of making these systems better based on the addition of a new kind of input data which is the users' awareness, not only physical but also social. To obtain this awareness automatically without the need of user interaction some methods can be utilized; one of them is de workflow technology.

Workflow technology is being incorporated mainly in the business environment to automate and optimize the execution of business processes. With a proper extension of this technology and using its power social context and awareness data can be obtained within working groups; users can benefit from this data and they can also serve as an enhancement of systems such as recommender systems.

On the other hand, as there is a huge amount of information on the Internet, it is also being filled up with a lot of different services. Each of these services imply to the user a new registration with a new pair of user -password (or any other type of credentials) that generates a worse usability and makes the user reject using new services, not being able to manage a growing list of credentials and perceiving a risk of losing his security. To change this paradigm a new ideas has appeared: the digital identity; it is a unique identity for each user that serves as a unique key to access the whole list of services a user employs. The main development of this idea is OpenID, but it has a set of security issues that are making its spread more slowly than what was expected.

Índice general

Resumen	i
Abstract.....	iii
Índice general.....	v
Índice de figuras.....	vii
Siglas	ix
1 Introducción.....	11
2 Estado de la tecnología.....	13
2.1 Identidad Digital	13
2.1.1 Identidad Digital Única	14
2.1.2 Beneficios.....	15
2.1.3 Inconvenientes	16
2.2 Contexto en el Desarrollo Distribuido de Software.....	16
2.3 Flujos de Trabajo.....	17
2.3.1 Definición y Evolución	18
2.3.2 Sistemas de Flujos de Trabajo.....	19
2.3.3 Motores y Lenguajes	20
2.4 Recomendadores	21
2.4.1 Tipos de Recomendadores	22
2.4.2 Algoritmos.....	24
2.4.3 Problemas y Retos	26
2.4.4 Métodos de Evaluación	27
3 Desarrollo Realizado.....	35
3.1 Análisis en Profundidad de OpenID y su problemática.....	35
3.1.1 Funcionamiento	35
3.1.2 URL o XRI.....	37

3.1.3	Herramienta complementaria: OAuth	38
3.1.4	Problemas de Seguridad.....	38
3.1.5	Soluciones Identificadas	41
3.1.6	Adopción	43
3.2	Generación de Contexto a partir de Flujos de Trabajo.....	44
3.2.1	La Herramienta Desarrollada: ItecDesk.....	44
3.2.2	Tecnologías de Desarrollo	46
3.2.3	Motor y Lenguaje de Definición de Flujos.....	47
3.2.4	Servidor.....	50
3.2.5	Cliente	50
3.2.6	Generación Dinámica de Vistas.....	51
3.2.7	Editor Gráfico de Flujos de Trabajo	52
3.2.8	Contexto Conseguído	53
3.3	Recomendadores basados en Contexto.....	54
3.3.1	Web Semántica.....	55
3.3.2	Contexto Físico.....	55
3.3.3	Redes Sociales	56
3.3.4	Ontologías, Taxonomías y Folksonomías	57
3.3.5	Otras Posibilidades.....	57
3.3.6	El Proyecto Galactus	58
4	Conclusiones y Trabajos Futuros	59
4.1	Identidad Digital y OpenID.....	59
4.2	Generación de Contexto a partir de Flujos de Trabajo.....	59
4.3	Recomendadores Basados en Contexto.....	61
4.4	Conclusiones Generales.....	62
	Bibliografía.....	63

Índice de figuras

Figura 1. Partes de un sistema de recomendación.	24
Figura 2. Ejemplo de k-nearest neighbor.	26
Figura 3. Distribución de datos de MovieLens.	31
Figura 4. Distribución de datos de Jester.	32
Figura 5. Distribución de datos de Netflix.	33
Figura 6. Distribución de datos de BookCrossing.	34
Figura 7. Resumen del protocolo OpenID Authentication	36
Figura 8. Definición de un flujo de trabajo concurrente simple.	49
Figura 9. Captura de pantalla de la herramienta ItecDesk.....	51
Figura 10. Ejemplo de definición de vista mediante SFDL.	52
Figura 11. Captura del editor gráfico de flujos de trabajo.....	53
Figura 12. Dimensiones de un sistema recomendador con contexto.....	56

Siglas

BPM	Business Process Management
BPML	Business Process Modeling Language
BPMS	Business Process Management Systems
BPMN	Business Process Modeling Notation
CAS	Central Authentication Service
CSRF	Cross site request forgery
CSS	Cascading Style Sheet
EPC	Event driven Process Chains
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
JSP	Java Server Page
LDAP	Lightweight Directory Access Protocol
MVC	Model View Controller
MXML	Multimedia eXtensible Markup Language
PHP	PHP (Personal Home Page) Hypertext Pre-processor
RAD	Rapid Application Development
REST	Representational State Transfer
RDF	Resource Description Framework
RIA	Rich Internet Applications
RoR	Ruby on Rails
RP	Relying Party

SAML	Security Assertion Markup Language
SDK	Software Development Kit
SFDL	Simple Form Definition Language
SVG	Scalable Vector Graphics
SOA	Service Oriented Architecture
SSO	Single Sign On
STP	SOA Tools Platform
SWF	Shockwave Flash
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	eXtensible Markup Language
XPDL	XML Process Definition Language
XRI	Extensible Resource Identifier
XSS	Cross site scripting

1 Introducción

Se puede observar en la actualidad que la cantidad de información presente en Internet es ingente y supone un verdadero problema encontrar el contenido buscado de una forma rápida y eficaz, teniendo en cuenta que las necesidades de cada usuario son diferentes y muy dependientes de su contexto. Esto implica una pérdida de prestaciones no sólo a nivel particular de cada usuario sino también a nivel corporativo, llevando a la utilización no óptima de recursos tan valiosos como el tiempo que acaban redundando en una pérdida económica.

Para solventar este problema surgieron los sistemas de recomendación que, a partir de unos datos de usuario y una serie de metodologías y algoritmos, son capaces de percibir las necesidades del usuario y filtrar grandes cantidades de información para acabar mostrando tan sólo la más relevante de una forma ordenada y coherente. Estos sistemas de recomendación están ya presentes en una gran cantidad de aplicaciones y servicios utilizados a diario, en ocasiones no como núcleo de la aplicación sino como soporte a la misma, pero en otros casos sirviendo de soporte central al servicio proporcionado. En nuestros días estos sistemas han llegado a suponer en sí mismos un nuevo mercado, generando en torno a ellos un modelo de negocio como el que puede observarse en servicios como los recomendadores de viajes o venta electrónica (Amazon).

El origen de estos sistemas se remonta más de una década, pero su evolución ha sido imparable y las líneas de investigación para mejorarlos siguen abiertas, obteniendo cada vez mejores recomendaciones y necesitando menos datos para conseguirlos. Problemas como realizar recomendaciones contando con unos pocos datos son comunes a todos los nuevos sistemas, por lo que se buscan también soluciones comunes. Estos problemas están muy identificados y existe mucha documentación acerca de cómo evitarlos.

Una de las vías de mejora de mayor actualidad es la incorporación de un nuevo tipo de datos de entrada como es el contexto, tanto físico como social, de los usuarios. Para obtener este contexto de forma automática sin requerir interacción con los usuarios se pueden utilizar varios métodos, entre otros el uso de la tecnología de flujos de trabajo. Con este contexto se enriquece en gran medida la recomendación ya que se conoce el estado del usuario en tiempo real pudiendo identificar necesidades concretas, anticiparse a peticiones e incluso detectar, reconocer y aprender patrones de comportamiento.

Esta tecnología de flujos de trabajo se está incorporando principalmente al mundo empresarial para automatizar y optimizar la ejecución de procesos de negocio. Los flujos de trabajo son hoy una realidad en gran cantidad de grandes entidades; aunque se utilizaron en un principio para modelar procesos de negocio estáticos y poco cambiantes con el objetivo de mejorar los tiempos de ejecución y necesitar una menor cantidad de recursos para llevarlos a buen puerto, hoy existen mejoras de la tecnología para conseguir flujos de trabajo de definición dinámica que permitan una mayor flexibilidad. Con una extensión concreta de la tecnología y aprovechando su potencial se pueden obtener datos de contexto social dentro de grupos de trabajo que pueden beneficiar a los usuarios y servir para mejorar sistemas como los motores de recomendación.

Por otro lado, al igual que existe en Internet una cantidad desmesurada de información, la red se está poblando de servicios cada vez más numerosos. Cada uno de estos servicios supone actualmente para los usuarios un registro con su consecuente par de usuario y contraseña (o cualquier otro tipo de credenciales) que genera una peor usabilidad y un rechazo a la hora de incorporar nuevos servicios a los ya utilizados habitualmente, entre otras cosas por no poder manejar tantas credenciales y percibir que se está poniendo en riesgo la seguridad. Para cambiar este paradigma ha surgido la idea de la identidad digital, entendida como una identidad única por cada usuario que le sirva para acceder a todos sus servicios. El principal desarrollo es OpenID, un sistema que permite el uso de un único par de usuario-contraseña basándose en la aparición de un nuevo ente: el proveedor de confianza. Este sistema tiene muchos detractores debido a sus problemas de seguridad, muchos de ellos solucionables pero no planteados en la propia definición del protocolo. Por ello están surgiendo otros sistemas evolucionando el mencionado para superar los problemas de seguridad y alcanzar un protocolo que suponga en el futuro la funcionalidad que permitirá un crecimiento mayor y de más valor de los servicios finales en Internet..

A lo largo de este documento se realiza un repaso del estado del arte de la idea de identidad digital, la tecnología de flujos de trabajo y los sistemas recomendadores. En la parte de desarrollo, se explica el profundo estudio realizado de los problemas de seguridad y soluciones propuestas de OpenID, la extensión de la tecnología de flujos de trabajo llevada a cabo para conseguir una herramienta de obtención de información de contexto y la utilización de estas dos tecnologías para potenciar los sistemas recomendadores actuales.

2 Estado de la tecnología

En esta sección se realiza un intenso repaso del estado de la tecnología en tres dominios a priori independientes que en siguientes secciones serán ligadas para llegar a un fin común. En primer lugar se introduce la idea de la identidad digital haciendo hincapié en sus beneficios e inconvenientes. Posteriormente se define la idea de contexto y su utilización en un caso concreto, el desarrollo distribuido de software. La tecnología de flujos de trabajo es inspeccionada en la sección 2.3 para acabar haciendo un repaso profundo del estado de los sistemas de recomendación en la actualidad.

2.1 Identidad Digital

La evolución de Internet en los últimos años ha llevado a una red de servicios de diferente ámbito coexistiendo. La utilización por parte de los usuarios se encamina cada vez más al acceso a un conjunto de estos servicios para obtener la funcionalidad proporcionada por cada uno hasta satisfacer una serie de necesidades.

Un número importante de estos servicios tienen que ver con las redes sociales y de compartición de contenidos. Cada servicio implica la necesidad de un registro y con ello se hace incómodo para el usuario final el acceso por conllevar la gestión de distintos pares de usuario y contraseña.

Para solventar este problema han surgido distintos métodos para poder contar con una identidad única que de acceso a todos los servicios, utilizando una sola autenticación ante un servidor de confianza. Con ello se consigue, no sólo mejorar la experiencia de usuario en cuanto a la comodidad de acceso, sino también permitir la aparición de nuevas funcionalidades relacionadas con la identidad única y la compartición de información y contenidos entre servicios.

En este aspecto, OpenID aparece como la opción de mayor adopción y madurez, pero su extensión no se está realizando con la rapidez esperada por contener una serie de problemas de seguridad de distinta índole. Es esencial que estos problemas queden solventados para poder universalizar el sistema de identidad única, ya que la confianza en este tipo de sistemas es lo más complicado de conseguir.

Los problemas de seguridad identificados están siendo intensamente estudiados y se está llegando a soluciones, tanto técnicas como en forma de buenas prácticas, que deberían conseguir hacer de OpenID un sistema robusto y convertirlo en sistema de identificación única global [1].

Además de solventar los agujeros de seguridad, la investigación en este entorno está dirigiéndose hacia la integración de OpenID con otros sistemas de seguridad o con

distintos tipos de aplicaciones, encontrando modelos de uso no identificados en el primer desarrollo.

En este documento se realiza un repaso al estado actual del campo de investigación de la identidad única, poniendo especial interés en los problemas de seguridad de OpenID en su versión 2.0 y sus soluciones, y recopilando las vías de investigación de mayor actualidad. La estructura del documento es como sigue: en primer lugar se introduce el concepto de identidad digital única, pasando después a definir el protocolo OpenID; se analizan en profundidad los problemas de seguridad encontrados y las soluciones identificadas, pasando en las últimas secciones a comentar la adopción del protocolo y una serie de conclusiones.

2.1.1 Identidad Digital Única

La gestión de la identidad digital, se centra en la creación de una única identidad digital que podría usarse en cualquier lugar en el que se requiera un inicio de sesión login para acceder a un servicio o sitio web. No se trata de una tecnología única, sino de un grupo de tecnologías e ideas asociadas. En términos simples, una forma de crear una identidad digital única que pueda ser utilizada en cualquier sitio en el que sea necesario acceder al sistema mediante un inicio de sesión. Se están desarrollando distintos sistemas, y aunque todos tienen el mismo propósito general de crear un sistema de registro que sea conveniente y seguro para un individuo más que para una compañía u organización, las ideas sobre lo que define con precisión un sistema de identidad centrado en el usuario y cómo podría implementarse siguen siendo muy heterogéneas. Tanto Google como Facebook están posicionando sus sistemas para ser el "hogar" de la identidad digital de los usuarios.

El manejo de una identidad única proporciona también la capacidad de que el usuario final defina los datos que quiere compartir con los diferentes sistemas. Una forma de hacerlo es introducir sus datos sólo una vez y establecer después cuántos de esos datos permite consultar a según qué servicio; de este modo se consigue un repositorio centralizado de datos, propios de una identidad. De la misma manera, en el proveedor de identidad se consigue un perfil rico de usuario al estar este registrado en distintas aplicaciones que aportan diferentes datos de perfil acerca de dicho usuario.

Las soluciones existentes para reducir la complejidad del uso y gestión de las credenciales de usuario en diferentes servicios pueden ser divididas en federadas (como SAML) y centradas en usuario (como OpenID) [2]. Por un lado la gestión federada de identidad es segura y más frecuente (especialmente en comunidades científicas) mientras que la centrada en usuario ofrece mejor usabilidad y mantenibilidad.

En la actualidad hay muchos usuarios que disponen de cuentas OpenID pero ni siquiera son conscientes de ello, como puede ser el caso de los usuarios de Yahoo. OpenID es un buen paso hacia la resolución de algunos de los problemas clave de la identidad digital a través de un estándar abierto que no trata de resolver todos los problemas a la vez, sino que, en su lugar, trata de resolver y manejar los distintos problemas y necesidades conforme van surgiendo.

2.1.2 Beneficios

Los beneficios [3] que se pueden obtener a partir de sistemas de gestión de identidad única son, entre otros, los siguientes:

- Mayores tasas de registro; dado que el usuario percibe que no es necesario un registro ya que puede utilizar su identidad única para utilizar un servicio concreto, los registros en servicios que dan estas facilidades aumentan considerablemente.
- Login con un solo click; el usuario no tiene que recordar un nombre de identificación y una contraseña para cada servicio, por lo que la autenticación se realiza de una forma más rápida y natural. La satisfacción del usuario aumenta y se reducen los gastos relativos al olvido de contraseñas.
- Datos de usuario actualizados; si un usuario ha decidido utilizar toda la potencia de estos sistemas aportando sus datos y decidiendo qué servicios pueden consultar qué grupo de datos, éstos siempre se mantendrán actualizados en todos aquellos servicios. Si el usuario actualiza cualquiera de sus datos, ésta actualización puede ser transmitida a todos los servicios consumidores.
- Importación de amigos y contactos; si entre la información a compartir en su perfil el usuario ha introducido sus contactos, éstos pueden ser comunes en todas las redes sociales usadas. De esta manera se evita tener que mantener actualizadas las listas de amistades de diferentes redes sociales o, por ejemplo, tener que añadir a una misma persona a todas las redes en las que el usuario participe.
- Flujos de datos bidireccional; la información compartida por el usuario fluye entre el proveedor de identidad y las aplicaciones o servicios autorizados. De esta manera, estos servicios pueden utilizar el flujo para compartir o monitorizar actividades del usuario de una forma natural en otras aplicaciones.
- Login federado entre aplicaciones asociadas; las aplicaciones pueden hacer que el registro de un usuario en un servicio sirva como autenticación ante otros asociados. De esta manera es posible, por ejemplo, que una empresa de transportes, tras vender un viaje, redirija y autentica a un usuario a una web de reserva de hoteles asociada, con recomendaciones directas de utilidad.

2.1.3 Inconvenientes

Es importante remarcar que no todo son ventajas en el uso de OpenID, también tiene una serie de problemas muy importantes, entre los que podemos destacar los siguientes:

- Perfiles de usuarios y pérdida de anonimato; actualmente, cada sitio en el que se tiene una cuenta sabe sobre el usuario lo que éste quiera contarle. Con OpenID, incluso aunque puedan mantenerse múltiples identidades, se están relacionando de manera inherente muchos servicios y en consecuencia perdiendo cierta parte del anonimato del usuario.
- Seguridad; sin entrar en detalle, ya que los apartados 4 y 5 de este mismo documento recogen en detalle los problemas de seguridad encontrados y sus posibles soluciones, hay que remarcar que existen una serie de compromisos en la adopción de OpenID que pueden redundar en la pérdida de seguridad. En el caso de que las credenciales de un usuario fueran interceptadas por un atacante, y al ser éstas únicas para todos los servicios utilizados por la víctima, el atacante tendría la capacidad de hacer un daño mayor que en un sistema de claves individuales por servicio.
- Usabilidad; para el usuario medio, OpenID es aún demasiado confuso para crear una cuenta y usarla. Simplemente el hecho de encontrar una página en la que poder crear su propia identidad de OpenID puede ser difícil; y el proceso de utilizarlo en un página que afirme aceptar este tipo de identidad es tosco y difícil. Por ejemplo, ¿cómo elige un nuevo usuario a su proveedor de OpenID? Además el proceso de autenticación puede ser confuso, pues implica que debemos dirigirnos a otro sitio web.

2.2 Contexto en el Desarrollo Distribuido de Software

El desarrollo de software a través de grupos de trabajo distribuidos es una práctica cada vez más extendida en la actualidad, debido a la emergencia de proyectos de carácter colaborativo en que las diferentes partes del diseño e implementación son encargadas a distintos conjuntos de desarrolladores, a menudo físicamente alejados.

En este tipo de proyectos es imprescindible contar con herramientas que consigan mantener la conciencia de grupo, a menudo denominada contexto, que garantice la coordinación y evite problemas como el trabajo duplicado. En la mayoría de los grupos de trabajo distribuidos este contexto es conseguido mediante la utilización de herramientas cotidianas no orientadas a tal propósito como las listas de correo o la mensajería instantánea; las soluciones existentes hoy en día no han tenido la acogida esperada.

En cualquier tipo de trabajo distribuido colaborativo la conciencia de grupo es esencial a la hora de desarrollar las diferentes tareas de una forma óptima. Mantener el contexto para conseguir que cada participante sea consciente de las actividades de sus compañeros es crítico para conseguir efectividad y coordinación.

El contexto de grupo es entendido como la capacidad de conocer en tiempo real quién está trabajando en el proyecto, qué está haciendo cada participante y cómo las acciones de un participante afectan al trabajo del resto [4]. Es de gran utilidad a la hora de conseguir coordinación, distribuir el trabajo, tomar decisiones, anticipar acciones y encontrar ayuda [5].

En el caso concreto del desarrollo de software, con su complejidad e interdependencia, es muy aconsejable el tomar medidas o usar herramientas que aseguren el mantenimiento del contexto.

Habitualmente existen tres mecanismos para mantener el contexto en ámbitos no distribuidos: la comunicación explícita mediante conversaciones entre participantes, la comunicación consecuente en la que un participante observa lo que hace su compañero, y la vigilancia de los cambios que se realizan en el repositorio de artefactos.

En el caso del ámbito distribuido, tal abanico de comunicaciones no se da con tal facilidad [6], provocando una falta de coordinación que repercute en el coste final del proyecto. La más potente de las tres variedades de comunicación es la explícita, la cual se intenta mantener en el proceso real de desarrollo distribuido mediante herramientas cotidianas y no optimizadas a tal efecto como el correo electrónico, la mensajería instantánea o el chat. Otro tipo de comunicación, como la que deriva de observar lo que hace el resto de participantes, se intenta añadir al caso distribuido mediante herramientas que muestran de forma gráfica, a menudo mediante códigos de colores y uniones entre nodos, las interacciones de cada cual con el resto y con el repositorio de artefactos; entre ellas se encuentra Tukan [7] y Augur [8]. Estas últimas herramientas no están teniendo una acogida real entre la comunidad de desarrolladores, por su dificultad de uso y sus escasos beneficios.

2.3 Flujos de Trabajo

La tecnología de flujos de trabajo [9] está creciendo en importancia y aceptación en diferentes entornos por aportar una serie de ventajas en optimización de procesos. Posibilitan la definición de diferentes procesos propios de las distintas fases de un proyecto de desarrollo software, especificando el tipo de profesional a intervenir en cada momento y el tipo de artefacto a modificar o crear, entre otras muchas opciones disponibles gracias a la potencia de esta técnica.

2.3.1 Definición y Evolución

Un flujo de trabajo, de forma genérica, enmarca el estudio de los aspectos operacionales de una actividad de trabajo: la forma en que se estructuran las tareas, cómo se realizan, su orden correlativo, su sincronización, la manera en que fluye la información que da soporte a las tareas y cómo se lleva a cabo el seguimiento del cumplimiento de las mismas.

Llevando la definición del flujo de trabajo al campo que nos ocupa, la tecnología de la información, éste se trata de una parte esencial del funcionamiento de aplicaciones de trabajo colaborativo. En este sentido un flujo de trabajo comprende la automatización de un proceso de trabajo, en todo o en parte, durante la que documentos, información o tareas son pasados de un participante a otro por acción, según una serie de reglas de procedimiento. La dirección del proceso representado por el flujo de trabajo es definida en un "Proceso de Definición" que identifica los procesos de actividades, reglas de procedimiento y datos de control asociados.

A los flujos de trabajo se les denomina workflows. Al utilizar un workflow, el elemento que fluye por el camino establecido por este workflow es denominado workitem. Cada uno de estos workitems tiene una serie de propiedades (campos) que podrán variar o no a medida que recorra el flujo de trabajo, ya que cada paso de dicho flujo está orientado a actuar sobre el workitem para procesarlo y generar un valor añadido. Este enriquecimiento será fácilmente apreciable al comparar el estado del workitem al término del flujo con su estado inicial.

En la actualidad esta tecnología está despertando un gran interés por conseguir mejoras de carácter económico basadas en la automatización de procesos. Los recursos humanos y materiales y el tiempo de proceso son optimizados gracias a esta automatización, lo que es muy ventajoso en el mundo empresarial actual. Otra razón para la adopción de esta tecnología es la aparición de herramientas capaces de integrar todas las aplicaciones que interactúan con el desarrollo de un flujo de trabajo, aplicaciones antes independientes.

Existe hoy en día un amplio grupo de aplicaciones en torno a la tecnología de los flujos de trabajo. Gran parte de ellas están orientadas a la creación y edición de los flujos. Las más complejas son soluciones integradas que permiten, además de la creación y edición, la ejecución del flujo de trabajo como servicio mediante la inclusión de motores de ejecución.

Dada la importancia de esta tecnología, se han desarrollado estándares para formalizar tanto los lenguajes de definición de flujos de trabajo como la representación

gráfica de los mismos. Entre los lenguajes más extendidos se encuentran XPDL [10] o BPEL [11], y la notación gráfica estandarizada es BPMN [12].

Existe también un amplio número de motores de flujos de trabajo sobre los que ejecutar los procesos. Estos motores están programados en diferentes lenguajes mayoritarios de programación como Java. En 2001 surge el proyecto de creación de un nuevo motor, OpenWFE [13], creado por John Mettraux. En 2006 pasa a llamarse OpenWFEru, siendo las últimas letras “ru” una referencia al cambio de tecnología que experimenta el proyecto, pasando de desarrollarse en Java a usar Ruby. Este cambio suscita mucho interés por tratarse Ruby de un lenguaje ágil y de rápida implementación; esta es la razón por la que muchos desarrolladores se acercan a este proyecto de código abierto, cuyo nombre actual es Ruote.

2.3.2 Sistemas de Flujos de Trabajo

El propósito de los sistemas de flujos de trabajo, también denominados “Sistemas de Gestión de Procesos de Negocio” o BPMS, es acercar personas, procesos y máquinas, con el objeto de reducir tiempo y acelerar la realización de un trabajo. Estos sistemas permiten la colaboración en equipo desde diferentes lugares físicos, facilitan la automatización de procesos mediante flujos de trabajo y permiten integrar los diferentes procesos de un proyecto, rediseñados de acuerdo a esta nueva estrategia.

En los últimos años y a nivel mundial, los sistemas de flujos de trabajo han despertado gran interés por dos razones principales: una económica y otra tecnológica. La económica se basa en el reconocimiento por parte de diversos sectores de la economía de que, debido a la globalización de los mercados, el ambiente competitivo de negocios actual necesita la automatización de todas las actividades y procesos de trabajo, y no solo la automatización de algunas tareas individuales. La razón tecnológica resulta de la aparición de nuevos ambientes computacionales, capaces de integrar varias aplicaciones que antes operaban de manera independiente. Debido a estas razones, las expectativas hacia la adopción de sistemas de flujos de trabajo de manera global son grandes.

Las exigencias del mercado y la presión de la competencia obligan a las organizaciones a ser más eficaces y eficientes en todas las áreas de servicio al cliente, producción, servicios internos y control. Los ordenadores personales y las aplicaciones de ofimática (procesadores de texto, hojas de cálculo, etc.) han incrementado la eficacia y desempeño de los individuos en las organizaciones de manera extraordinaria. Ahora, gracias a la interconexión de los puestos de trabajo mediante las redes locales y la tecnología Web, es posible incrementar también los niveles de eficacia y desempeño de los grupos de trabajo mediante las redes internas empresariales o intranets.

La realización de actividades coordinadas en las que participan dos o más miembros de un equipo de acuerdo con reglas de negocio establecidas son costosas y demandan importantes recursos organizacionales tanto a nivel de personal, materiales y equipos de oficina como logística. La automatización de los procesos que hasta el momento se realizaban transfiriendo formularios en papel de una persona a otra, puede representar a la organización ahorros económicos anuales muy importantes con inversiones moderadas en la implantación de sistemas de flujos de trabajo.

Existe en la actualidad un conjunto de nuevas herramientas que permiten conseguir esta automatización. Clasificadas dentro del paquete genérico de “herramientas de trabajo en grupo” o “software colaborativo”, incluyendo como piezas clave las herramientas de automatización de procesos administrativos y las de gestión de la documentación. Como se comentó anteriormente, los sistemas de flujos de trabajo son una pieza fundamental de estas aplicaciones colaborativas.

2.3.3 Motores y Lenguajes

Un motor de flujos de trabajo es un software capaz de dotar de control del entorno de ejecución de cada instancia de un flujo de trabajo. Existe en la actualidad un gran número de desarrollos de este tipo de software, tanto de carácter libre como propietario, diferenciándose en las necesidades de cada usuario final y en el dominio de aplicación para el que se requieran.

Entre los motores de pago existentes destacan Microsoft’s BizTalk Server, Microsoft’s Windows Workflow Foundation (WF), Oracle’s BPEL Progress Manager, BizAgi y Symantec Workflow. En cuanto a las soluciones libres destacan jBPM, OpenWFE y Enhydra Shark.

En cuanto a los lenguajes de definición de procesos o flujos de trabajo [14], existen, como se ha comentado previamente, una serie de lenguajes extendidos como son BPEL y XPDL.

BPEL es un lenguaje ejecutable con soporte para XML e intercambio de mensajes SOAP [15] para sus operaciones. Su objetivo principal es la orquestación de servicios web, y la secuencia de interacción y flujo de datos. Sin embargo, carece de dos importantes características: soporte gráfico –los flujos no contienen información acerca de los diagramas de definición que los generaron- e interacción humana –que sólo se consigue mediante extensiones-. XPDL, por el contrario, soporta no sólo la representación gráfica de los procesos, sino que cada paso puede incluir la descripción de la actividad, temporizadores, llamadas a servicios web, o roles para diferentes tipos de participantes (incluyendo humanos). Su portabilidad garantiza la fácil conversión a BPEL en aquellos casos en los que no se soporta de manera nativa.

2.4 Recomendadores

Desde el comienzo de la expansión de Internet el crecimiento del número de fuentes de información existentes en la red ha seguido un ritmo exponencial; tanto es así que en la actualidad es posible encontrar una ingente cantidad de información en la Web, haciendo compleja la tarea de identificar la información y el conocimiento relevantes para cada usuario en cada situación. Este problema pone de manifiesto la necesidad de contar con sistemas de extracción capaces de filtrar información antes de ser presentada al usuario mediante una predicción que lleve a una medida de cuán interesante será ese contenido para el usuario. Estos sistemas son los llamados sistemas recomendadores. En los últimos tiempos, los sistemas recomendadores se han convertido en parte de la solución a la sobrecarga de información sufrida por los consumidores finales. Proveen a los usuarios de sugerencias acerca de qué contenido es el más apropiado o interesante en cada caso.

Existen aplicaciones comerciales en funcionamiento que cuentan con una fuerte dependencia de un sistema de recomendación. Como ejemplo se puede encontrar Amazon que recomienda diferentes tipos de productos como libros o discos a sus usuarios en función de las últimas compras realizadas, Netflix que trabaja en la recomendación de películas o Last.fm con gran éxito a la hora de presentar al usuario canciones que casan con sus gustos musicales. Todas estas aplicaciones han demostrado que tanto la predicción como el consumo de cualquier tipo de elemento sigue el modelo de Long Tail, con un reducido grupo de elementos interesantes para la mayoría de los usuarios y un enorme grupo restante de interés para usuarios muy concretos. Del mismo modo se ha identificado que es esencial que estos sistemas no desestimen estos elementos finales por ser de gran utilidad para conseguir predicciones exactas.

Varios tipos de recomendadores conviven en la actualidad, siendo cada uno el más apropiado para según qué caso de uso. Existen varias clasificaciones que expresan las distintas maneras en las que cada tipo obtiene las predicciones correspondientes. De la misma forma, desde que surgieron los primeros sistemas de recomendación, los algoritmos encargados de realizar las aproximaciones matemáticas de cálculo de similitudes y aplicación de pesos a las posibles soluciones han ido sucediéndose y evolucionando.

Lejos aún de ser una solución idílica, estos sistemas son vulnerables a una serie de problemas que los hacen ser, en la mayoría de los casos, poco fiables e inexactos. Algunos problemas están relacionados con las condiciones de arranque, otros con los usuarios poco definidos o grises, y otros con las dificultades derivadas de la necesidad de procesamiento de grandes cantidades de datos y la realización de complejas

consultas a bases de datos en tiempo real. Un importante problema es el relacionado con la privacidad a la hora de compartir datos de usuarios. También supone una complicación la evaluación de estas aplicaciones, obligando en muchos casos a probarlos contra bases de datos ficticias que poco tienen que ver con la realidad del sistema en producción.

Este documento está organizado de la siguiente manera: en la siguiente sección se introduce una clasificación actualizada de los sistemas de recomendación más habituales; en la tercera sección se hace mención de los últimos algoritmos desarrollados; en la cuarta, se detallan los problemas más habituales y las soluciones aplicadas; la quinta sección trata los métodos de evaluación que se llevan a cabo; la sexta sección comenta algunos de los casos de éxito más notables y la séptima versa acerca de los conceptos más novedosos en el campo de la recomendación. El documento termina con un apartado de conclusiones.

2.4.1 Tipos de Recomendadores

En esta sección se enumeran los tipos de sistemas recomendadores surgidos a lo largo de la historia, siendo lo más frecuente en la actualidad el uso de uno de los dos enumerados en primer lugar, o bien un sistema híbrido como mezcla de estos dos. La clasificación presentada no es única [16], existiendo diferentes tipos clasificación en función de las características estudiadas y la profundidad requerida.

1. Basados en contenido

Este tipo de sistemas operan sobre el perfil del usuario activo y la información que se tiene acerca de los elementos o ítems. Además, tratan de recomendar al usuario elementos similares a los que le interesaron en el pasado.

Este enfoque nace a partir de las ideas de los sistemas de recuperación de información y adopta muchas de sus técnicas, por ejemplo, el análisis de documentos de texto mediante técnicas de minería de texto para poder hacer recomendaciones en base a su contenido.

Sus principales desventajas radican en que tienen un completo desconocimiento acerca del entorno social del usuario, por lo que se dificulta la recomendación de objetos novedosos y desconocidos para éste, y que dada la naturaleza del formato de los ítems puede ser difícil analizar su contenido de forma automática, por lo que debería especificarse manualmente mediante algún tipo de representación.

2. Filtros Colaborativos

Esta aproximación opera sobre el usuario activo y su entorno social, es decir, usuarios con características similares. Es muy diferente a la aproximación anterior ya que en vez de recomendar a un usuario ítems similares a los que le resultaron de

interés un tiempo atrás, recomienda ítems que resultan interesantes a usuarios con características y gustos similares, de modo que en vez de computar la semejanza entre ítems, computa la semejanza entre usuarios.

Este método resuelve los problemas del enfoque basado en contenido ya que puede tratar cualquier tipo de contenido, sea cual sea su naturaleza, y puede ofrecer al usuario objetos totalmente novedosos y desconocidos. No obstante, esta aproximación introduce dos nuevos problemas conocidos como first-rater y cold-start, que serán estudiados en profundidad en las siguientes secciones.

3. Modelos de Grupos

Este enfoque es una generalización del filtro colaborativo. Se introduce el concepto de granularidad de los grupos o clusters ya que con una granularidad muy fina, es decir, cuando los grupos son de tamaño uno, el sistema será de filtro colaborativo. En enfoque de modelos de grupos trata de encontrar usuarios similares al usuario activo para hacer las recomendaciones en base a los intereses de éstos. Para ello divide los usuarios en grupos o clusters y trata el problema como si se tratase de una tarea de clasificación. El objetivo del algoritmo es asignar el usuario activo a uno de los grupos que se han creado. Una vez asignado a un grupo, el sistema utiliza las acciones y características de los usuarios de ese grupo para realizar las recomendaciones pertinentes.

Este enfoque tiene mejor escalabilidad y rendimiento que un enfoque de filtro colaborativo porque compara el usuario activo con los usuarios del conjunto asignado, ahorrándose el tener que manejar todos los usuarios del sistema.

4. Basados en Búsquedas

Este enfoque se basa en realimentar la consulta que el usuario hace sobre el sistema con información que ya se conoce acerca de sus intereses, para así recomendar ítems similares o de categorías similares. Por ejemplo, en el comercio electrónico se tienen en cuenta las compras y valoraciones que hace un cliente.

Si el cliente ha hecho pocas compras o ha valorado pocos productos, el sistema escala y desempeña de forma eficiente su función. Pero si el usuario ha comprado o valorado miles de productos, es muy ineficiente ejecutar una consulta con información sobre todos los productos en cuestión, por lo que el algoritmo debería construir la consulta con un subconjunto de productos, reduciendo así la calidad de las recomendaciones. De todos modos, este tipo de sistemas no son muy utilizados por que sus recomendaciones son pobres debido a que no recomienda ítems novedosos ni desconocidos.

5. Filtros Colaborativos Objeto-a-Objeto

Es el enfoque utilizado por Amazon.com en su página Web. Este enfoque ha sido diseñado exclusivamente para manejar información de millones de ítems y usuarios. El algoritmo que implementan goza de gran escalabilidad, siendo capaz de trabajar con conjuntos de datos enormes, y produciendo recomendaciones de gran calidad en tiempo real. A diferencia del enfoque de filtro colaborativo común, que se basa en recomendar ítems que resultan de interés a usuarios con características similares al usuario activo, este enfoque recomienda ítems que han adquirido otros usuarios y que éstos a su vez tienen adquisiciones similares a las del usuario activo. Para determinar el ítem más similar a uno dado, el algoritmo construye una tabla de similitudes entre ítems basándose en los ítems que adquiere cada cliente. Afirman que “si un cliente C1 compró los productos X e Y, y otro cliente C2 compra el producto X, entonces el cliente C2 podrá estar interesado en comprar el producto Y”. El algoritmo que construye dicha tabla es muy costoso en términos de tiempo y recursos, pero puede ser ejecutado en modo offline.

2.4.2 Algoritmos

Existe una gran variedad de algoritmos que son aplicados a los sistemas recomendadores [17]. Cada uno tiene sus rasgos específicos, ya sea para aplicarse a un determinado enfoque de los vistos en la sección 2, o para manejar una cantidad determinada de información de usuarios e ítems. Cada algoritmo será utilizado en una o varias de las etapas típicas de cualquier sistema recomendador mostradas en la figura 1. A continuación se comentarán brevemente algunos de ellos.

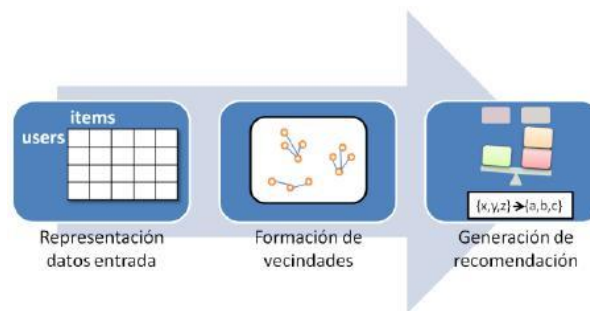


Figura 1. Partes de un sistema de recomendación.

1. Slope One

Este algoritmo es muy utilizado gracias a su eficiencia y facilidad de implementación. Combina las opiniones y gustos de los distintos usuarios para ofrecer recomendaciones personalizadas al usuario activo. Este algoritmo parte de la asunción de que existe una relación lineal entre los valores de preferencia expresados por parte de un usuario por un ítem y otro, y que se puede estimar la preferencia por un ítem Y a partir de la preferencia conocida por un ítem X mediante una función lineal de tipo recta de pendiente 1 con un valor de offset. Existen dos variantes: basado en usuario

(calcula la similitud entre usuarios) y basado en ítems (opera sobre valoraciones de los ítems y estadísticas de uso).

Este algoritmo predictor tiene una serie de evoluciones para solventar su carácter simplista. La primera de ellas es el uso de slope one con pesos; esto se realiza para establecer que serán tenidos en cuenta en mayor medida los ítems que más usuarios hayan evaluado, ya que la evaluación media total será mucho más fiable. Una segunda evolución es el llamado slope one bipolar; que utiliza el algoritmo en dos partes, siendo la primera para obtener los ítems que han gustado al usuario y la segunda para obtener los que no le han gustado. De esta manera se reduce el número de elementos a tener en cuenta a la hora de generar la predicción, si bien es cierto que se plantea la dificultad de establecer dónde se encuentra el límite, en cuanto a cuantía de la valoración, entre lo que gusta y lo que no gusta al usuario.

2. Rocchio

La búsqueda de información es muy dependiente de la experiencia del usuario a la hora de seleccionar las palabras clave con las que se hará la consulta. El algoritmo de Rocchio se utiliza en enfoques basados en búsquedas y trata de optimizar las consultas de manera incremental.

Para ello, el sistema debe permitir al usuario que valore los ítems que el sistema le devuelve. Esta valoración se hará en términos de lo interesantes que le resultan estos ítems al usuario en referencia a la consulta que hizo.

Gracias a esta información suministrada por los usuarios, el algoritmo puede modificar la consulta inicial de las búsquedas -ya sea estableciendo pesos a los términos, o añadiendo y eliminando términos de la búsqueda- para ofrecer ítems más interesantes.

3. Ripper

Ripper es un algoritmo de inducción de reglas que trabaja de forma similar a los algoritmos que construyen árboles de decisión: particionando recursivamente el conjunto de datos. Su eficacia se debe a un algoritmo de poda que optimiza el conjunto de reglas que se obtienen a partir de un determinado conjunto de datos. Además, soporta atributos multi-valorados, siendo esta característica muy interesante cuando se trabaja con textos semi-estructurados.

4. ID3

Los algoritmos de tipo ID3 construyen un árbol de decisión particionando recursivamente el conjunto de datos de entrenamiento hasta obtener clases puras. En el contexto de clasificación de textos, la pureza de una clase se mide en función de la presencia o ausencia de términos. Este tipo de algoritmos son muy potentes cuando el

conjunto de datos sobre el que trabajan son datos estructurados (por ejemplo tablas de bases de datos relacionales). Sin embargo, cuando se trata de datos no estructurados, es necesario preprocesar los datos mediante algún tipo de técnica (minería de textos si los elementos del conjunto de datos son textos o documentos digitales).

El uso de este algoritmo en los sistemas recomendadores se encuentra en la necesidad de clasificar los ítems que adquiere un usuario para poder aprender una función que represente su perfil y, de este modo, poder recomendar ítems de la misma clase.

1. *K-Nearest Neighbor*

Se trata de un algoritmo de clasificación, que puede ser adaptado a multitud de escenarios y áreas como visión por computador, reconocimiento de patrones, compresión de datos, sistemas recomendadores, etc.

Su nombre significa k vecinos más cercanos, donde la k es un valor entero positivo. El algoritmo trata de clasificar un elemento en función de la clase más común entre sus k vecinos más cercanos. La elección del valor de k no es trivial y depende de los datos, por lo que existen heurísticas para elegirlo. Normalmente una k grande reduce el efecto del ruido de la clasificación, pero hace más difusa la frontera o separación entre clases. En la figura 2 se muestra un ejemplo donde los cuadrados azules representan una

clase $C1$ de ítems, mientras que los triángulos rojos representan una clase distinta $C2$. El ítem a clasificar es el círculo verde. Con $k=3$, el ítem en cuestión pertenecería a la clase $C2$. Por el contrario, con $k=5$, el ítem pertenecería a la clase $C1$.

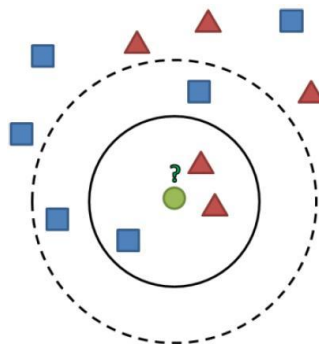


Figura 2. Ejemplo de k -nearest neighbor.

2.4.3 Problemas y Retos

A la hora de realizar un sistema recomendador se presentan un conjunto de retos que solventar de muy distinto ámbito. Muchos de ellos son comunes a todo sistema recomendador. Los más habituales se listan a continuación:

- First-Rater. Ocurre cuando un nuevo ítem es introducido en el catálogo ya que no se tiene ningún tipo de información acerca de él. Este ítem no podrá ser

recomendado hasta que algún usuario especifique algunas de sus características o un conjunto de ítems similares.

- **Cold-Start.** Ocurre cuando un usuario se da de alta en el sistema y, como éste no conoce nada acerca de sus gustos e intereses, no se puede hacer ningún tipo de recomendación. Existe un problema similar cuando un usuario tiene intereses que no son comunes entre el resto de usuarios del sistema.
- **Grey-Users.** Trata la problemática de encontrar similitudes entre usuarios cuyos gustos no están definidos por ser totalmente similares a los de la media del sistema. Las recomendaciones a estos usuarios suelen ser poco precisas.
- **Data-Sparseness.** Se da el caso cuando se tienen pocos datos, ya sea de usuarios o elementos, que estos datos están muy poco relacionados y es difícil obtener “vecindades” entre usuarios o elementos.
- **Malicious-Users.** Usuarios utilizando de forma maliciosa los sistemas de valoración hacen que el sistema pierda efectividad. Ante un grupo grande de usuarios, este efecto es mínimo; el problema real surge con sistemas de relativamente pocos usuarios.
- **Privacidad.** Para realizar buenas recomendaciones se necesitan datos de los usuarios para poder establecer un perfil y relacionarles con otros usuarios. Esto plantea problemas legales relacionados con la privacidad. A la hora de utilizar mecanismos de última generación como el contexto social en los recomendadores, este sistema cobra especial importancia por necesitar la interrelación de bases de datos heterogéneas [18][19].
- **Datos cambiantes.** Es difícil para los sistemas recomendadores actuar de forma acorde a la evolución de los datos. Las valoraciones de los usuarios para un mismo elemento a menudo varían con el tiempo y es complejo identificar esa variación y tenerla en cuenta a la hora de recomendar.
- **Elementos impredecibles.** Existen, en algunos ámbitos, elementos cuya valoración es extremadamente buena o mala en función del usuario valore. Un ejemplo son las películas excéntricas o algunos grupos musicales que los usuarios “adoran” u “odian”. Es arriesgado recomendar este tipo de elementos.

La resolución de todos estos problemas y otros intrínsecos de cada tipo de sistema recomendador mantiene activa a la comunidad de investigadores. No existen soluciones totales pero sí pequeñas adaptaciones que pueden solventar alguno de los puntos anteriormente citados.

2.4.4 Métodos de Evaluación

Uno de los mayores problemas a la hora de desarrollar un nuevo sistema de recomendación es la forma de evaluarlo. Se han intentado desarrollar métricas [20] para cuantificar de forma objetiva cuánto de bueno es un sistema recomendador

[21][22]. La gran mayoría de estos sistemas están basados en el uso de conjuntos de datos de dominio público estudiados en profundidad de manera que al ejecutar el sistema recomendador sobre los mismos se conoce a priori el resultado que se debería obtener.

En esta sección se repasan las principales métricas y conjuntos de datos utilizados en la actualidad para evaluar los sistemas recomendadores.

1. Métricas de evaluación

Existen una serie de métricas utilizadas para establecer de forma objetiva y numérica la calidad de un sistema recomendador. A continuación se listan las principales métricas utilizadas.

- **Precisión**

La precisión es la principal característica medida a la hora de evaluar un sistema recomendador. Se trata más de un conjunto de métricas que de una sola; ha sido la forma de evaluación más estudiada a lo largo de la historia de este tipo de sistemas y sigue siendo hoy el principal indicador de calidad al realizar un desarrollo de este tipo.

Se asume que si un usuario pudiera examinar un conjunto de elementos dado, podría ordenarlo en función de sus preferencias. De esta manera, la precisión de un sistema recomendador mide empíricamente cuánto se acerca la predicción de la lista ordenada de elementos realizada por el sistema a la realizada por el usuario. Puede también referirse a cómo de bien es capaz de predecir el sistema la valoración exacta que daría un usuario a un elemento concreto.

Para cada tipo de recomendación existe una métrica concreta que establece la forma de medir la distancia de lo predicho por el sistema de lo real.

- **Cobertura**

La cobertura es la medida del dominio de elementos en el sistema sobre los que el recomendador puede realizar predicciones o recomendaciones. Cuanta mayor cobertura tenga un sistema mayor será el valor aportado al usuario final, ya que será mayor el ámbito de ayuda que pueden aportar.

La usual tarea de “encontrar todos los elementos buenos” necesita de una cobertura amplia ya que si existe un conjunto de elementos en el dominio estudiado que no son evaluados, no se podrán encontrar todos los buenos.

Una forma de definir esta medida es averiguando para qué porcentaje de elementos puede el sistema realizar predicciones (cobertura de predicción) o qué porcentaje de elementos recomienda alguna vez el sistema (cobertura de recomendación o catálogo).

Este tipo de medida permite comparar diferentes sistemas recomendadores de manera que se pueda detectar un sistema que, a pesar de tener una precisión alta, no es bueno, ya que siempre está recomendando un pequeño conjunto de elementos de aceptación consabida.

- Tasa de aprendizaje

Los sistemas recomendadores incorporan algoritmos de aprendizaje que operan sobre modelos estadísticos. Como resultado, su rendimiento varía en función de la cantidad de datos aprendidos disponibles. Con el aumento de los datos obtenidos mediante aprendizaje debería aumentar la calidad de las recomendaciones realizadas.

Diferentes sistemas recomendadores pueden obtener resultados aceptables con distintas tasas de aprendizaje, en función del entorno en el que operen. Algunos algoritmos tan sólo necesitan unos pocos datos de entrada para empezar a generar resultados apropiados mientras que otros requieren el uso intensivo de aprendizaje.

Se han definido tres tipos de tasas de aprendizaje: global, por usuario y por elemento. La primera define la calidad de las recomendaciones en función del total de valoraciones existentes en el sistema o el total de usuarios. La tasa por usuario toma la calidad de las recomendaciones para un usuario en función del número de valoraciones que ese usuario ha introducido en el sistema. Por último, la tasa por elemento es la calidad de las predicciones para un elemento en función del número de valoraciones disponibles sobre dicho elemento.

Es complejo evaluar este tipo de métrica dado que la mayoría de los conjuntos de datos utilizados para la experimentación cuentan con un gran número de entradas, siendo imposible evaluar el comportamiento del sistema con un conjunto reducido de datos.

- Novedad y Casualidad

Existe una realidad paradójica en los sistemas recomendadores: no es de utilidad para los usuarios finales que se les recomienden elementos que ya saben que existen y les gustan ya que no se les aporta ninguna información novedosa y por tanto ningún valor; sin embargo, estudios revelan que los usuarios agradecen encontrar recomendaciones de este tipo porque les permiten comprobar que el sistema funciona correctamente y que ha captado sus gustos.

Un ejemplo claro es el encontrado en una tienda de discos con elementos muy reconocidos como el White Album de The Beatles. Cualquier aficionado a la música conoce este álbum y muchos ya lo tienen; aquel que no lo tiene es porque ha decidido no comprarlo. Dar la recomendación de comprar este álbum no llevará, por tanto, a su

compra. Sería de mucho más valor para los usuarios una recomendación acerca de un grupo desconocido de que hagan música del gusto del usuario y que éste no conozca y de otra manera quizá nunca hubiese conocido.

Las recomendaciones obvias tienen dos desventajas: los usuarios interesados en esos elementos ya los han adquirido previamente y no es eficiente desarrollar o adquirir un motor de recomendación para obtener recomendaciones que todo el mundo conoce.

- **Confianza**

Los usuarios de los sistemas recomendadores se encuentran habitualmente con la situación de no saber cómo interpretar las recomendaciones obtenidas ya que existen dos dimensiones en conflicto y difíciles de diferenciar. La primera se refiere a cuánto predice el sistema que le gustará el elemento recomendado al usuario, y la segunda se refiere a cuánto de seguro está el sistema de que la recomendación es precisa.

De esta manera, los sistemas que presentan las recomendaciones junto a un marcado de puntuación (estrellas, valores, etc.) pueden confundir al usuario ya que para elementos con pocas valoraciones buenas la puntuación mostrada será mayor que para otros elementos, más adecuados o no, que tienen muchas más valoraciones y, por tanto, su valoración media tiende a ser menor. Sin embargo, valoraciones menores no implican una mayor base de conocimiento. Por ello se trata ésta de una métrica compleja.

La forma de presentar esta métrica de forma correcta es darle la posibilidad al usuario de navegar en los resultados de la recomendación en las dos dimensiones introducidas anteriormente.

- **Evaluación de los usuarios**

Las métricas comentadas hasta el momento involucran variables de medida que afectan la utilidad percibida por el usuario del sistema recomendador. Esta última sección trata la evaluación directa de la reacción de los usuarios finales ante el uso del sistema.

Esta medida es bastante más compleja que las anteriores ya que trata las opiniones personales de los usuarios, siendo éstas totalmente subjetivas y en ningún caso normalizadas.

Existen diferentes formas de obtener la evaluación de los usuarios; por un lado se encuentra la obtención explícita (preguntando) y la implícita (observando). Además existen los laboratorios o centro de investigación preparados con entornos controlados para evaluar distintas hipótesis. Se tendrá que evaluar el global del sistema y cada

proceso o tarea por separado. Por último, cada una de estas evaluaciones tendrá que realizarse en el corto plazo y también en el largo plazo.

Se han realizado un gran número de estudios acerca de la manera de evaluar tanto sistemas en general como sistemas recomendadores cuando se trata de obtener las opiniones de los usuarios finales.

2. Conjuntos de datos (Datasets)

Los investigadores y desarrolladores en el campo de los sistemas de recomendación utilizan una serie de conjuntos de datos o datasets para evaluar el rendimiento del sistema y sus algoritmos. A continuación se introducen una serie de sistemas abiertos, sencillos de acceder, maduros en cuanto a su utilización intensiva en el pasado y con muchas posibilidades de seguirse usando en el futuro.

- MovieLens

Los conjuntos de datos proporcionados por MovieLens [23] son muy populares y han sido objeto de numerosas investigaciones. Se tratan de datos totalmente públicos y accesibles que contienen valoraciones explícitas acerca de películas, información demográfica sobre los usuarios (edad, sexo, ocupación, dirección, etc.) y una breve descripción de cada película (título, año, género, etc.). Se trata de un conjunto de datos de gran densidad, en el que las valoraciones de películas van de 1 a 5 y han sido realizadas por 943 usuarios activos (con 20 o más valoraciones realizadas) sobre 1682 películas. Varios conjuntos de datos están disponibles: el primero, llamado U, contiene 100.000 valoraciones completas; las cinco siguientes, denominadas desde U1 a U5, contienen grupos de 10.000 valoraciones. Existen dos más conteniendo 10.000 valoraciones llamadas UA y UB.

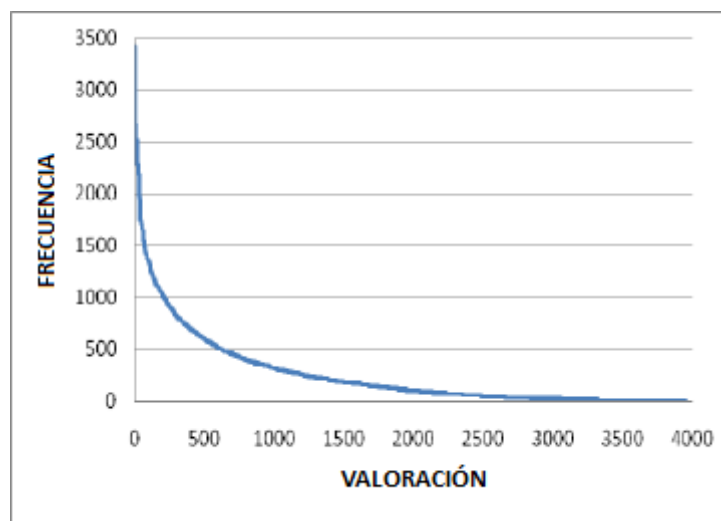


Figura 3. Distribución de datos de MovieLens.

La distribución que siguen los datos de este dataset es del tipo power law distribution ya que se da el caso de obtener un enorme número de películas con valoraciones muy bajas (parte izquierda de la figura 3) y un muy pequeño número de películas con valoraciones cercanas al máximo (parte derecha de la figura 3).

- Jester

Fue liberado por Ken Goldberg de Jester Joke Recommender System [24]. Contiene valoraciones anónimas de 76.496 usuarios sobre 100 chistes. Las valoraciones van desde -10 hasta +10. Gracias a que el contenido a valorar son chistes, muchos usuarios continúan leyendo y valorando hasta acabar la lista de chistes disponibles, de manera que se consigue un conjunto de datos mucho más denso que el aportado por otros proveedores.

En este caso se pueden encontrar tres conjuntos de datos: el primero, llamado Jester-data-1, contiene valoraciones de 24.983 usuarios que han valorado más de 36 chistes cada uno; la segunda, llamada Jester-data-2, contiene la evaluación realizada por 23.500 usuarios acerca de más de 36 chistes; por último, Jester-data-3 contiene los datos de 24.983 usuarios que han valorado entre 15 y 35 chistes.

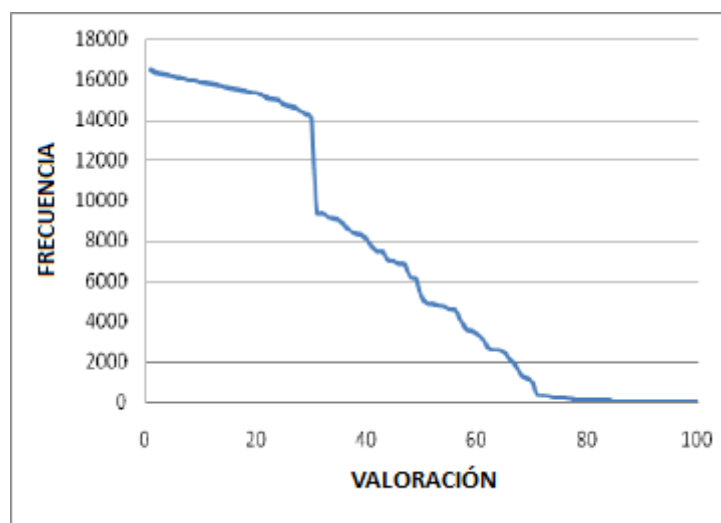


Figura 4. Distribución de datos de Jester.

Los resultados obtenidos para cualquiera de los tres conjuntos de datos son similares en cuanto a su distribución. En la figura 4 se muestra la distribución de Jester-data-1, y en ella se puede apreciar que con mayor frecuencia se suelen utilizar valoraciones bajas y tan sólo unos pocos chistes son valorados con el máximo de la escala.

- Netflix

Se trata, al igual que MovieLens, de un conjunto de datos muy utilizado y popular, y fuente de numerosos estudios. En 2006 se presentó el concurso The Netflix Prize [25] con el que se ofrecía un premio de un millón de dólares a la primera persona capaz de alcanzar una meta de precisión definida en las bases a la hora de recomendar películas. Se liberó un inmenso conjunto de datos incluyendo más de 100 millones de valoraciones de películas valoradas entre 1 y 5. Los usuarios, anónimos, alcanzaban la cifra de 480.159 y las películas eran 17.770. Se proporcionaban fechas de cada valoración y nombres de películas.

A partir de este concurso este conjunto de datos ha sido utilizado para la evaluación de sistemas de recomendación.

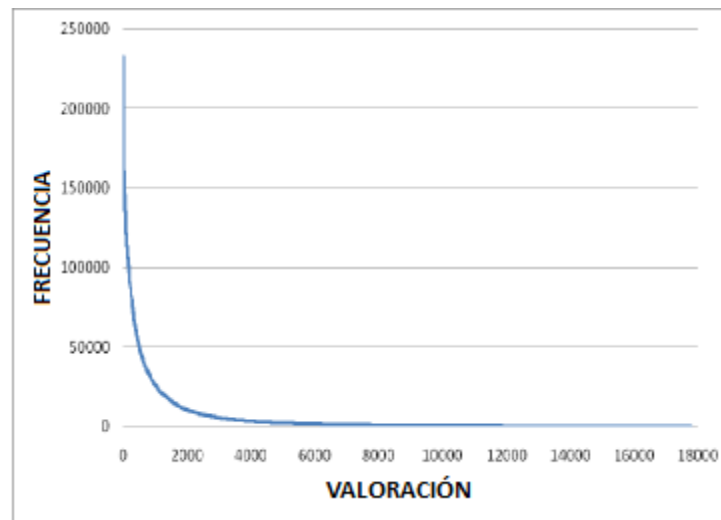


Figura 5. Distribución de datos de Netflix.

La distribución de estos datos, como puede observarse en la figura 5 es similar a la seguida por los datos de MovieLens incluyendo el concepto de Long Tail.

- BookCrossing

Este conjunto de datos fue recogido por Cai-Nicolas Ziegler de la comunidad Book Crossing [26] en 2004. Contiene 278.858 usuarios anónimos con 1.149.780 valoraciones acerca de 271.379 libros, expresados mediante una escala entre 1 y 10. Este conjunto de datos, tal y como ocurría con MovieLens, contiene información demográfica acerca de los usuarios y una breve descripción de los libros valorados (título, autor, año de publicación, editorial, etc.).

La distribución que siguen estos datos sigue, al igual que MovieLens y Netflix, la power law distribution. Esto puede observarse en la figura 6.

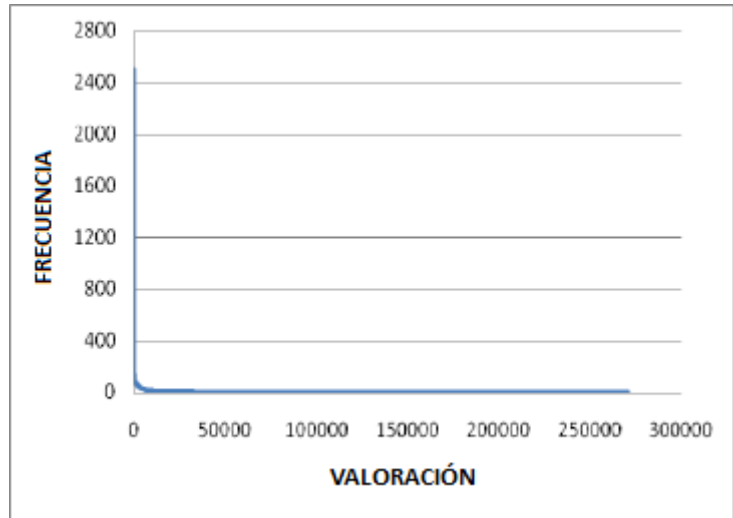


Figura 6. Distribución de datos de BookCrossing.

3 Desarrollo Realizado

En esta sección, y una vez repasado el estado de distintas tecnologías que cooperan en el objetivo de este proyecto, se detallan los desarrollos realizados en los distintos aspectos tratados. En primer lugar se explican los resultados obtenidos de un profundo estudio del protocolo OpenID y de su problemática y las soluciones identificadas hasta el momento. Posteriormente se detalla la extensión realizada a la tecnología de flujos de trabajo que permite obtener información de contexto de usuarios trabajando en un mismo proyecto utilizando como soporte los flujos de trabajo. Por último se llega a la unión de todos los conceptos para conseguir un sistema de recomendación potente que utiliza el contexto de sus usuarios para mejorar las recomendaciones de salida.

3.1 Análisis en Profundidad de OpenID y su problemática

OpenID es un estándar de autenticación, desarrollado de forma inicial en 2005, que puede ser usado para el control de acceso, permitiendo a un usuario acceder a diferentes servicios con una misma identidad digital (una URL o XRI como se comentará más adelante). Es un método que utiliza una única autenticación contra un proveedor de confianza. De este modo no es necesario que el usuario cree nuevas cuentas ni que se registre en otras páginas web. La primera vez que se accede a un servicio web se le tiene que explicitar permiso al proveedor de confianza para que comparta parte o toda la información del perfil con ese nuevo servicio. Internamente el proveedor de confianza confirma al nuevo servicio que el usuario es quien dice ser, ya que se realizó la autenticación previa con el proveedor de OpenID. Los proveedores de confianza son cada vez más, algunos de ellos de gran calibre como son Google, Yahoo! o Microsoft.

3.1.1 Funcionamiento

El funcionamiento del protocolo OpenID es de una complejidad bastante reducida. Se puede ilustrar en una serie de pasos:

1. El usuario abre una cuenta en un servidor de confianza, por ejemplo, OpenId.es, Google, Microsoft, etc. Es esencial confiar en el proveedor del servicio, ya que el nivel de seguridad del protocolo depende en gran medida de la forma en que se gestiona desde el servidor de confianza. Se obtiene, tras el registro, la URI o XRI que servirá como identificador único.

2. Opcionalmente se puede utilizar la URL de un dominio propio del usuario como identificador, mediante la inserción de unas líneas de HTML en el código fuente del sitio web bajo dicho dominio. Si el host de dicho sitio web no dispone de un servidor de OpenID, el usuario podrá delegar dicha funcionalidad en un servidor de OpenID de confianza mediante nuevas sentencias HTML.

3. Para acceder en un servicio web que soporte OpenID, solo tiene que introducir para su autenticación la dirección URL o XRI de su perfil OpenID o, si ha seguido el paso 2, la URL de su web personal.

4. La URL OpenID se enviará hasta el servidor elegido de confianza, mostrándole al usuario un formulario de entrada en el que tendrá que introducir su contraseña, y acto seguido autorizar los datos que quiera compartir con el servicio al que está accediendo.

5. Después de esto el servidor devuelve el control al servicio web, que ya conoce quién es el usuario.

La próxima vez que inicie sesión en ese u otro servicio web que acepte OpenID (proceso "OpenID Authentication") no tendrá que volver a introducir su contraseña, ya que esos datos se guardarán temporalmente en su navegador, evitándole más identificaciones innecesarias. En la Fig. 1 se muestra un esquema resumido del funcionamiento del protocolo OpenID en su proceso de autenticación. Conociendo este procedimiento se puede observar una primera premisa de seguridad cumplida: el servicio web no tiene acceso a la contraseña del usuario en ningún momento, al contrario de los sistemas tradicionales. Solo tiene acceso a los datos que el usuario quiere y en el momento en que lo desea.

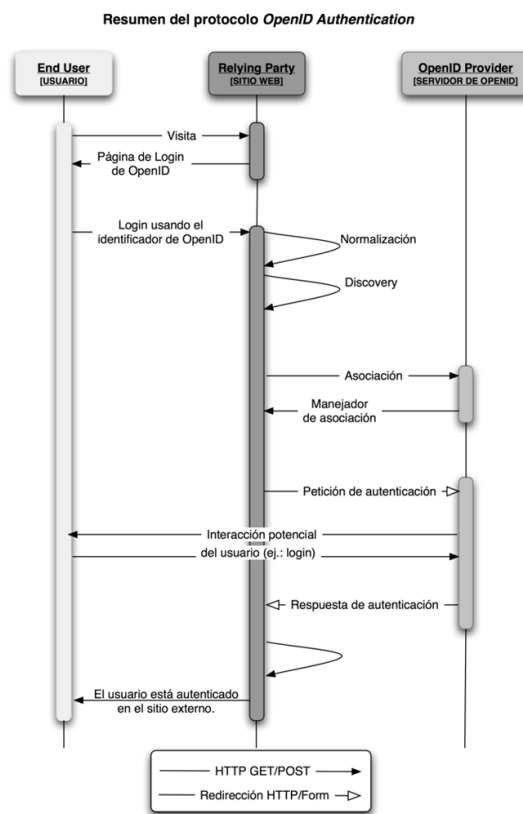


Figura 7. Resumen del protocolo OpenID Authentication

3.1.2 URL o XRI

A partir de la versión 2.0 de OpenID, se añade la posibilidad de utilizar XRI en lugar de una URL como identificador único. Para ello los servidores de OpenID tienen que soportar esta funcionalidad añadida; en la actualidad, el soporte de autenticación mediante URL ronda el 97%, mientras que el soporte a XRI está en torno al 41%. Por tanto, la norma general es que los servidores que soportan XRI soportan también URLs. La inclusión de XRI al proyecto OpenID parte de los inicios de la investigación en la concreción de gestión de identidad digital mediante identificadores. Las propuestas OpenID y XRI eran disjuntas, pero tendieron a converger con el paso del tiempo. La adopción de XRI añade una serie de ventajas al protocolo OpenID:

- Añade soporte al formato de descubrimiento de metadatos de XRDS y al protocolo de resolución . Estos mecanismos están relacionados con el origen de XRDS como herramienta para publicitar servicios o recursos.
- Protección ante la reasignación de identificadores OpenID entre usuarios, mediante el uso del mapeo de un XRI entendible para los usuarios (y, por tanto, con peligro de reasignación) llamado i-name a un XRI persistente, no reasignable, llamado i-number.
- Capacidad de diferenciación entre los i-names o i-numbers pertenecientes a usuarios concretos de los pertenecientes a organizaciones.
- Fuerte protección de privacidad en el registro de los i-names o i-numbers debido a la no existencia de un servicio de Whois, datos de contacto requeridos mínimos, y privacidad lograda mediante anonimidad de los datos de contacto.

El descubrimiento del servidor de OpenID varía levemente en función de la opción elegida. Una URL puede resolver directamente a un servidor de OpenID, llevar a un HTML con unas líneas que definan la ubicación del mismo o bien tener asociado un documento XRDS. Por otra parte, todas las XRIs tienen asociados un documento XRDS en el que se especifica la ubicación del servidor de OpenID entre otros servicios o recursos indexados.

El formato de cada una de las dos opciones es la que sigue:

- OpenID URL:

`http://example.name o example.name`

`https://example.name o example.name`

`http://example.com/user o example.com/user`

- OpenID XRI:

=name

=first.last

@organization

Aunque en su arranque el uso de XRI parecía no tener buena acogida, en la actualidad es un formato que está tomando fuerza tanto en este ámbito como en su propósito inicial: el descubrimiento de recursos.

3.1.3 Herramienta complementaria: OAuth

OpenID no soporta la compartición de contenido entre diferentes servicios, sólo solventa por tanto el problema de la identidad en el momento del acceso a diferentes páginas web. Para conseguir compartir este contenido, que es uno de los puntos clave para conseguir el nuevo paradigma de red útil para el usuario comentada anteriormente, es necesario utilizar otros esquemas como OAuth.

OAuth es un protocolo abierto que permite autorización segura de un API de modo estándar y simple para aplicaciones de escritorio, móviles, y web. Para desarrolladores de aplicaciones, OAuth es un método de interactuar con y publicar datos protegidos. Para desarrolladores y proveedores de servicio, OAuth proporciona a los usuarios un acceso a sus datos al mismo tiempo que protege las credenciales de su cuenta. En otras palabras, OAuth permite a un usuario del un servicio concreto compartir su información presente en ese servicio (proveedor de servicio) con un servicio diferente (llamado consumidor) sin compartir toda su identidad.

Una vez diferenciada la funcionalidad que aporta cada una de las tecnologías, es necesario reseñar que su utilización complementaria llevaría a la realidad la idea de identidad única con compartición de contenidos a lo largo de todos los servicios a los que un usuario quiera acceder. Esta interacción es ya soportada gracias a desarrollos como el realizado por Google, que proporciona ya a los desarrolladores de aplicaciones un API con el que conseguir la autenticación mediante el sistema de OpenID utilizando como proveedor de confianza el propio Google, y al mismo tiempo la autorización para la utilización de contenido propietario del usuario mediante el protocolo OAuth.

3.1.4 Problemas de Seguridad

En cuanto a seguridad, OpenID tiene una serie de problemas que hace que su adopción se ralentice [27].

Como ya se ha introducido anteriormente, si el par “nombre de usuario/contraseña” cayera en manos de un potencial adversario, el número de sitios

web en los que la identidad digital del usuario se vería comprometida sería mucho mayor. En las especificaciones de OpenID se contemplan distintos aspectos de seguridad, como los referentes al establecimiento de claves, firma de mensajes y mecanismos de verificación, así como con el uso de canales protegidos TLS/SSL; sin embargo, el empleo de estos sistemas de seguridad es completamente opcional. Este aspecto supone una de las principales diferencias que se establecen con otros estándares usados en infraestructuras federadas, sea éste por ejemplo el caso de SAML, donde se establecen robustos sistemas de seguridad .

Estos problemas se dan tanto en el proceso de la autenticación básica, cuando el principal peligro es el phishing, y durante el mantenimiento de la funcionalidad de Single-Sign-On (SSO) la información de la sesión puede ser robada.

En las siguientes secciones se presentan los detalles de cada amenaza de seguridad presente en OpenID en función de su origen.

Amenazas inherentes al protocolo

Debido a las propias especificaciones de OpenID, una serie de agujeros de seguridad han quedado abiertos, posibilitando diferentes ataques, y alimentando la creencia de un sector de los investigadores y desarrolladores de que OpenID no es una solución de futuro al problema de la identidad.

A continuación se detallan los problemas de seguridad cuyo origen son las propias especificaciones:

- Navegador como Man-in-the-middle; los mensajes y aserciones viajan sin cifrar entre el servidor de OpenID y el sitio web, también llamado Relying Party (RP). De esta manera, el navegador se presenta como una parte muy interesante que atacar. Se podría, por ejemplo, alterar o copiar mensajes o aserciones debido a su falta de cifrado y a que la mayoría de los mensajes viajan sin firmar.

Además, los mensajes propios del protocolo están muy poco ligados unos a otros y al funcionamiento o ejecución del protocolo en sí mismo, siendo esta una medida que haría más pequeño el campo de ataque relacionado a este agujero de seguridad.

- Intercambio de sesiones; un atacante puede forzar al navegador de una víctima a hacer login en un RP controlado por el atacante. Esto es posible ya que el mensaje de "aserción positiva" no está atado al navegador de origen. El funcionamiento de este ataque es como sigue: el servidor de OpenID autentica a M (el atacante); M puede hacer que el navegador de A (la víctima) envíe el mensaje de aserción a un RP elegido por M. Las consecuencias son variadas: desde hacer que A entre con la cuenta de M en el buscador preferido de A, pudiendo registrar y por tanto espiar las búsquedas de la

víctima, hasta engañar a la víctima para que ingrese los datos de su tarjeta de crédito en la cuenta de compras online de M.

Amenazas inherentes a navegadores / HTTP / Web

A parte de las amenazas surgidas a partir del propio protocolo, otras derivan del uso de este servicio en la web. De esta manera, y debido al funcionamiento propio de los navegadores, el protocolo HTTP y, en general, la Web, se pueden encontrar estas debilidades:

- Cross site request forgery (CSRF); aprovechando la confianza de un sitio web hacia un usuario validado por el servidor de OpenID, el atacante puede forzar al navegador de la víctima a transmitir una petición maliciosa a la aplicación.

- Cross site scripting (XSS); basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado. En el caso del ataque a OpenID toma la forma de código malicioso inyectado en sitios web que puede llevar, incluso, al robo aserciones.

- Phising; se trata de una de las amenazas más preocupantes de OpenID, de la que se ha estudiado en profundidad [28]. Cuando el usuario usa su identidad única para autenticarse en un RP cualquiera, y no está autenticado en su servidor de OpenID, se le muestra una pantalla de su servidor de OpenID para que ingrese su par "usuario/contraseña" propio de su cuenta de OpenID. Justo en ese momento, un RP maléfico puede mostrar al usuario una pantalla similar a la del servidor de OpenID, pero falsa, en la que el usuario ingresará su par "usuario/contraseña" entregando estos datos al atacante. Es sencillo para un atacante mostrar a la víctima la pantalla de autenticación correcta (asociada al servidor de OpenID que utiliza el usuario) ya que, accediendo a la URL proporcionada por el usuario como identidad única, se obtiene directamente qué servidor de OpenID tiene asociado.

- End-entitiy Man-in-the-middle; es un problema que todos los protocolos de Single-Sign-On de la Web tienen que afrontar. Se trata de la posibilidad de tener servidores de OpenID y/o RPs maléficos. El RP redirigirá al navegador de la víctima a un servidor de OpenID falso para obtener sus credenciales.

Estas amenazas pueden ser mitigadas mediante mejoras en las especificaciones del protocolo o de los perfiles de uso.

Confianza del usuario

Como se ha explicado anteriormente, el sistema OpenID está basado completamente en la confianza que tenga el usuario hacia su servidor o proveedor de identidad. Por ello, a pesar de existir la posibilidad de que cualquiera, sin necesidad de ser un ente

reconocido, pueda implementar su propio servidor de OpenID, los usuarios adscritos a este sistema sólo seleccionan servidores de renombre debido a la necesidad de confiar en los mismos.

Este es un inconveniente, ya que dificulta para el usuario habitual de internet la adopción de OpenID por la confusión que genera el no tener un referente principal o tener que elegir entre multitud de posibilidades. Además, a pesar de que un servidor sea confiable, puede dejar de serlo en el momento que sufra un ataque o que su seguridad haya sido vulnerada. Es en ese momento en el que entran en juego las diferencias entre proveedores de OpenID, siendo las políticas de seguridad de cada uno diferentes; a la hora de elegir es necesario tener muy en cuenta la forma de actuar de cada servidor ante un ataque, siendo lo más óptimo la existencia de un compromiso por parte del servidor de informar de forma eficaz, rápida y transparente de cualquier posible vulnerabilidad encontrada o ataque filtrado. Del mismo modo, en función de las políticas de seguridad propias de cada opción, el usuario se sentirá más o menos protegido y, del mismo modo, tendrá más facilidad para comprobar o detectar posibles problemas de ataques en su propia cuenta de OpenID.

3.1.5 Soluciones Identificadas

Los aspectos de seguridad en OpenID introducidos en el apartado anterior se están tratando de manera activa {Document Not In Library}, pero las soluciones propuestas aún están en fases iniciales de desarrollo y cada OpenID tiene la responsabilidad de elegir su propia solución. Hasta que esta situación no se resuelva, los usuarios y compañías están optando por no utilizar este protocolo en comunicaciones que necesiten una alta privacidad, como bancos o sistemas sanitarios.

A continuación se detallan las principales soluciones en las que se está trabajando:

- Evolución del protocolo; el protocolo OpenID tiene mucho que mejorar en cuestión de seguridad. Los requisitos de funcionamiento son, en este aspecto, demasiado permisivos. Por ejemplo, no es obligatorio el uso de HTTPS, ni tampoco está definida una política que haga expirar las sesiones en distintos RPs. Los problemas de seguridad inherentes al protocolo sólo pueden ser solventados modificando o mejorando dicho protocolo.

- Mecanismos de confianza; para evitar los problemas de seguridad provocados por la existencia de RPs o servidores de OpenID maléficos, podrían usarse mecanismos de mantenimiento de confianza que consiguen establecer qué servidores son confiables y cuáles no.

- Técnicas anti-phising; dado que este ha sido uno de los problemas más importantes, no sólo en el ámbito de la identidad digital sino también la Web en

general, se han realizado muchos estudios acerca de las técnicas para evitarlo [29]. Algunas de estas técnicas son las siguientes: uso de una imagen personal introducida por el usuario en el sistema, que será presentada en la página vulnerable de introducción de credenciales; soluciones propias de cada navegador (Verisign and IE7's Extended Validation Certificate [30], Verisign and Firefox's seatBelt [31]); solución de clave de paso como imagen por parte de Vidoop [32]; autenticación de Jabber mediante SMS o Messenger [33]. A pesar de existir todos estos métodos, ninguno es infalible, por lo que continúa la investigación en este ámbito.

- Integración con otros sistemas de seguridad; una de las posibilidades que está tomando fuerza es la integración de OpenID con otros sistemas de seguridad para mejorar dicha seguridad. Es el caso de la integración con tarjetas SIM [34] que podrían habilitar los operadores de telefonía haciendo posible el acceso de sus usuarios a diferentes servicios sin necesidad de registro y garantizando la seguridad.

- Buenas prácticas; desde OpenID se insta a todas las partes implicadas en el proceso a cumplir una serie de buenas prácticas que redunden en una máxima garantía de seguridad.

A los usuarios finales se les recomienda utilizar como clave única una clave difícil de adivinar y mantenerla privada; también se les recomienda hacer uso de opciones de autenticación seguras que pueda implementar su servidor de OpenID, como el uso de clientes certificados; por otro lado se les insta a vigilar el phishing comprobando la URL de la página de login, utilizar su identidad con el prefijo de seguridad "https", sólo utilizar OAuth o similares para compartir contenido entre aplicaciones y evitar así la compartición de credenciales y hacer log out del proveedor de OpenID y de cada RP visitado tras su uso.

A los RPs les invita a protegerse contra ataques de tipo XSS y CSRF; también se les indica que no escriban la implementación de OpenID desde cero, sino que partan de las librerías ya existentes de código libre. Por otro lado se establece que las sesiones abiertas en los RPs no deberían durar más de lo que dura una sesión abierta en el servidor de OpenID. Cada RP debería implementar RP Discovery, un mecanismo que permite legitimizar los RPs y hacerlos confiables y, del mismo modo, emplear técnicas de confianza ante los servidores de OpenID presentados por los usuarios. Además, una serie de peligros son puestos de manifiesto proponiendo otras posibles salvaguardas.

Por último, los proveedores de OpenID son también llamados a ciertas prácticas. Entre ellas se encuentra la necesidad de abrir la ventana de login en una ventana nueva, y de educar a sus usuarios en la prevención y detección del phishing. Además, el uso de HTTPS, la utilización de métodos para evitar los intentos de fuerza bruta para la

extracción de contraseñas, la repartición de identificadores con formato HTTPS, y el uso de políticas varias de seguridad es introducido.

A pesar de todas estas soluciones, y dado el carácter poco alentador de algunas como el establecimiento de buenas prácticas, que no obliga a su cumplimiento y deja, por tanto, la puerta abierta a los problemas de seguridad, un sector amplio de la comunidad en Internet rechaza el uso de OpenID en aplicaciones críticas.

3.1.6 Adopción

Por lo comentado anterior, la adopción no se está dando de la forma en que se esperaba, si bien es cierto que ha tenido un importante crecimiento. Este crecimiento está asociado a la acogida del protocolo por parte de grandes compañías de internet. Alguno de los casos más relevantes es comentado a continuación:

- VeriSign es una empresa de seguridad informática famosa por ser una autoridad de certificación reconocida mundialmente. Emite certificados digitales RSA para su uso en las transmisiones seguras por SSL, principalmente para la protección de sitios en Internet en su acceso por http. Esta empresa de reconocido prestigio lanza ya en 2006 su primera implementación de servidor de OpenID.

- Yahoo! decide en 2008 hacer los identificadores de sus 250 millones de usuarios válidos como identificadores de OpenID, siendo el propio Yahoo! el proveedor de dicho servicio. De esta manera triplica de un golpe las cuentas válidas existentes de OpenID en aquel momento.

- Más tarde, también en 2008, Google se convierte en proveedor de OpenID, abriendo un API en Febrero en su versión de pruebas para permitir a los desarrolladores comenzar a hacer pruebas con el mismo. En 2009, Google hizo que el perfil de cada usuario pueda funcionar como su URL identificativa en el protocolo OpenID, de manera que evita la necesidad de crear la cuenta y ofrece ésta con los datos ya actualizados.

- MySpace del mismo modo en 2008 se une a la iniciativa de lograr una identidad única, haciéndose proveedor de OpenID y permitiendo a sus usuarios hacer login en todos aquellos servicios que soporten este protocolo con su cuenta de MySpace.

AOL, Orange, la mayor operadora de Japón NTT docomo y muchas otras compañías se han ido añadiendo a la lista de proveedores de OpenID, mientras que una inmensa cantidad de aplicaciones y servicios web [35] permiten hoy la autenticación mediante este protocolo.

3.2 Generación de Contexto a partir de Flujos de Trabajo

En este proyecto se propone la tecnología de flujos de trabajo como una alternativa de fácil adopción y gran flexibilidad a las herramientas actuales que buscan el mantenimiento de contexto en grupos de trabajo distribuidos. En concreto se detalla el proyecto ItecDesk, desarrollado para el caso de uso concreto de desarrollo de software en un entorno bancario, pero cuya base es aplicable a cualquier otro tipo de entorno, sirviendo además este caso de uso para realizar el estudio de viabilidad del que queda constancia en este documento.

El proyecto ItecDesk es creado tomando como base la tecnología de flujos de trabajo para automatizar y optimizar procesos propios de un proyecto de desarrollo software distribuido. El requisito principal del diseño de esta herramienta es dotarlo de un carácter genérico, de manera que se puedan crear o modificar en tiempo real definiciones de procesos ejecutables en el motor de flujos de trabajo ubicado en la parte servidor de la herramienta. Además se incorpora una característica innovadora, inexistente en el resto de aplicaciones del mismo ámbito, que es la definición de formularios de forma dinámica mediante un lenguaje creado a tal efecto, incorporada en el cuerpo de la definición del flujo de trabajo. Esto servirá para presentar a cada usuario final vistas que faciliten su interacción con el flujo de trabajo en sus diferentes estadios. La parte cliente de la herramienta contiene un mecanismo de renderizado de vistas o formularios a partir de esta definición, e interactúa con el motor de flujos de trabajo mediante un interfaz REST [36].

Se presentan además en este documento desarrollos paralelos de apoyo a la herramienta ItecDesk que la hacen aún más potente como es un editor gráfico interactivo de flujos de trabajo que aísla al diseñador de procesos de su programación, tarea que implicaría de otra manera conocimientos avanzados en este campo. Además existe un editor gráfico de formularios que, al igual que el anterior, genera el código de definición de vistas facilitando esta labor.

El uso de esta herramienta provee al equipo de desarrollo de un conocimiento global de las acciones que se llevan a cabo, de los diferentes participantes que las realizan y de la evolución que supone para el proyecto, generando con ello el contexto necesario para facilitar la coordinación y optimizar la realización del trabajo, de forma más eficiente que el uso de listas de correo, y de adopción más sencilla que otros sistemas.

3.2.1 La Herramienta Desarrollada: ItecDesk

ItecDesk es un proyecto orientado al uso de la potencia de la tecnología de flujos de trabajo para automatizar, organizar y, sobre todo, optimizar tareas entre las que destaca el proceso de desarrollo software en un núcleo bancario, donde un número

significativo de profesionales distribuidos emplean la gestión de flujos de trabajo para coordinar sus actividades diarias.

Uno de los requisitos principales de este proyecto es que toda la interacción del usuario final se hiciera a través de tecnologías web, minimizando el impacto de despliegue de la solución en un entorno corporativo. Del mismo modo, los interfaces de comunicaciones entre los diferentes componentes de la aplicación debían permitir la interacción entre la web humana y la web de datos y servicios, utilizándose para ello protocolos basados en estándares como REST.

Este proyecto utiliza como motor de manejo de flujos de trabajo el proyecto OpenWFERu, tecnología que se describe en los siguientes apartados. Este motor se ejecuta en la parte servidor de la aplicación, programado con tecnología Ruby on Rails, mientras que el lanzamiento de los flujos y las acciones a desarrollar en cada paso del mismo se llevan a cabo mediante una aplicación web cliente programada en Flash mediante el entorno Flex.

Para utilizar la aplicación los usuarios acceden mediante un navegador web al cliente Flash. Una vez arrancado el cliente, los usuarios han de autenticarse contra un servicio de autenticación central CAS [37], utilizado para proveer seguridad al entorno y facilitar una posible integración futura con otras herramientas. Una vez aceptadas las credenciales del usuario, se le muestra una vista en la cual puede proceder a iniciar un nuevo flujo de desarrollo, o ejecutar acciones que tenga pendientes de flujos que hayan llegado a un punto de acción en el que una interacción humana es requerida, concretamente una del usuario conectado.

La forma de interactuar con los flujos de trabajo es mediante formularios renderizados a partir de datos comprendidos en el mismo flujo. Estos formularios son rellenados por el usuario y, al aceptarlos, los datos introducidos son recogidos y enviados hacia el motor de ejecución como resultado del paso en el que estuviera parado el flujo. El motor evalúa dichos resultados y, a partir de los mismos, continúa con la ejecución del proceso. Esta ejecución puede llevar al lanzamiento de funciones automáticas que dan valor añadido a los datos introducidos por el usuario, a un punto de espera para la interacción con un nuevo usuario o a la finalización del flujo, entre otras muchas posibilidades.

Durante la vida de un flujo de trabajo, entre su generación y la llegada a su punto final, se generan eventos entre los que se encuentran las notificaciones programadas en la propia definición del proceso. Estas notificaciones son también mostradas al usuario en su vista principal.

A continuación se detallarán las tecnologías utilizadas en el desarrollo de la aplicación, el motor de flujos de trabajo seleccionado junto con el lenguaje de definición de flujos utilizado, una breve introducción de los módulos cliente y servidor, y un comentario acerca de avances introducidos por esta solución como son la generación dinámica de vistas o formularios y la edición gráfica de flujos de trabajo.

3.2.2 Tecnologías de Desarrollo

Como se ha introducido en la sección anterior, esta herramienta tiene una arquitectura cliente-servidor. Las tecnologías de cada una de las partes son igualmente diferenciadas; la parte servidor en la que se enmarca el motor de ejecución de flujos de trabajo ha sido desarrollada utilizando un lenguaje de programación ágil y dinámico como es Ruby; por el contrario, la parte cliente ha sido implementada utilizando Flex para conseguir una mejor experiencia de usuario. Ambas tecnologías son brevemente introducidas a continuación.

Adobe Flex es un conjunto de tecnologías que permite el desarrollo y despliegue de RIA basadas en la plataforma propietaria de Adobe Flash. Actualmente, incluye un kit de desarrollo de software (Flex SDK) y un entorno de desarrollo o IDE llamado Flex Builder.

Los programadores tradicionales se encontraban con multitud de dificultades al trabajar con las herramientas de desarrollo Flash, al estar muy orientadas a diseñadores gráficos y animadores basando su interfaz en metáforas fácilmente entendibles por éstos como líneas de tiempos, cuadros, etc.

El objetivo de Flex es permitir a los desarrolladores de aplicaciones web construir rápida y fácilmente Rich Internet Applications (RIAs) [38] basadas en un modelo de varias capas, donde las aplicaciones Flex son el nivel de presentación.

Para ello, se introduce como lenguaje principal de esta tecnología MXML, un lenguaje de marcado al estilo de XML que permite definir interfaces de usuario bastante similares a los de las aplicaciones de escritorio. La interactividad de estos interfaces viene dada por ActionScript, el lenguaje propio del núcleo de Flash Player.

En el modelo de varias capas el cliente sólo carga la aplicación una vez, no siendo necesaria la recarga al realizar cambios en ella, mejorando por tanto el flujo de datos frente a aplicaciones web basadas en HTML como PHP, ASP o JSP, donde era necesario recargar la página tras cada cambio.

Ruby es un lenguaje de programación interpretado, orientado a objetos y distribuido bajo licencia de software libre. Combina sintaxis inspirada en Python y Perl con características de SmallTalk.

Creado en Japón en 1993, presentado en 1995, fue inicialmente diseñado y desarrollado por Yukihiro Matsumoto bajo la idea de dar lugar a un lenguaje de programación que se asemeje al natural.

Ruby es orientado a objetos: todos los tipos de datos son un objeto, incluidas las clases y tipos que otros lenguajes definen como primitivas, (como enteros y booleanos). Toda función es un método. Las variables siempre son referencias a objetos, no los objetos mismos. Ruby soporta herencia con enlace dinámico y métodos *singleton* (pertenecientes y definidos por una sola instancia más que definidos por la clase). A pesar de que Ruby no soporta herencia múltiple, las clases pueden importar módulos. La sintaxis *procedural* está soportada, pero todos los métodos definidos fuera del ámbito de un objeto son realmente métodos de la clase *Object*. Como esta clase es padre de todas las demás, los cambios son visibles para todas las clases y objetos.

Ruby es un lenguaje interpretado, es decir, no se compila, aspecto que permite una mayor flexibilidad a la hora de generar código de manera dinámica, cualidad que es de suma importancia para el desarrollo de aplicaciones web y que fue una de las principales razones para desarrollar el framework de Rails sobre él.

Entre las grandes ventajas del lenguaje, destacan su flexibilidad para funcionar en múltiples paradigmas, ya que permite a sus usuarios alterarlo libremente, pues las partes esenciales de Ruby pueden redefinidas a placer.

Ruby utiliza un sistema propio para gestionar paquetes que provee un formato estándar para distribuir programas escritos en Ruby y bibliotecas. El formato en el que los programas son contenidos recibe el nombre de “gema”. La herramienta que gestiona estas gemas es RubyGems. El entorno Rails se instala como una gema, y es una de las razones principales que ha hecho que Ruby haya aumentando su penetración en el mercado al ir de la mano de Rails.

3.2.3 Motor y Lenguaje de Definición de Flujos

OpenWF^{eru}, cuyo significado es “Open Source Ruby Workflow Engine” es un motor de código libre para el manejo y ejecución de flujos de trabajo programado en el lenguaje Ruby. Surge en 2001 como OpenWFE, creado por John Mettraux. En 2006 pasa a llamarse OpenWF^{eru}, siendo las últimas letras “ru” una referencia al cambio de tecnología que experimenta el proyecto, pasando de desarrollarse en Java a usar Ruby. Este cambio suscita mucho interés por tratarse Ruby de un lenguaje ágil y de rápida implementación; esta es la razón por la que muchos desarrolladores se acercan a este proyecto de código abierto, cuyo nombre actual es Ruote.

En torno a Ruote se encuentra una comunidad de desarrolladores de aplicaciones basadas en flujos de trabajo que utilizan como motor de ejecución el propio motor

Ruby. La definición de flujos de trabajo para ser ejecutados en este motor se realiza en un lenguaje propio del proyecto basado en XML, Ruby y Json. Este lenguaje no está orientado a la representación gráfica del flujo de trabajo definido, de manera que no existen editores o modeladores gráficos para la edición de flujos de trabajo ejecutables en el motor Ruote.

Como respuesta ante la falta de editores para la creación y edición de flujos de trabajo en el motor Ruote, y frente a la existencia de este tipo de editores en cualquier otro proyecto de flujos de trabajo, se lleva a cabo en paralelo al proyecto ItecDesk el desarrollo de un editor gráfico interactivo para aplicaciones genéricas de flujos de trabajo, orientado al lenguaje propio del proyecto Ruote.

En comparación con los motores comentados en apartados anteriores, OpenWFE ofrece un rango más amplio de características que otros como jBMP y Enhydra Shark. Estos dos soportan un conjunto limitado de operadores de cara al control del flujo de trabajo, lo que supone un bajo soporte para los patrones que se encuentran fuera de la categoría básica. OpenWFE ofrece un mayor rango de facilidades para tareas útiles como son la concurrencia. En cuanto a los datos, los tres ofrecen soporte a un limitado rango de enlaces de elementos de datos y se basan en gran medida en elementos de datos a nivel-case [39].

Actualmente el proyecto se divide en una serie de subproyectos. Ruote es el proyecto con el que se inició toda la actividad. Es el motor de flujos de trabajo de código abierto programado con tecnología Ruby. Aunque en ciertos entornos la afirmación de que Ruby es más lento que Java en ejecución es correcta, en este caso concreto fue acogido por acortar el tiempo de desarrollo de manera significativa, tratándose Ruby de un lenguaje de programación muy ágil. Además, el problema de velocidad es cada vez menos crítico por la evolución en potencia de los equipos en los que se ejecuta. Del mismo modo los intérpretes del lenguaje Ruby son cada vez más rápidos.

Como motor de flujos de trabajo, se basa en la extendida disciplina de manejo de procesos de negocio BPM, de manera similar al resto de motores existentes en la actualidad.

La relación entre el lenguaje Ruby y los flujos de trabajo es, en principio, débil, siendo el primero muy conocido por su integración en el entorno Rails para el desarrollo ágil de aplicaciones web. Para los desarrolladores que utilizan esta dupla, los flujos de trabajo no suponen, en principio, más que “máquinas de estados” accesibles mediante una extensión de Ruby on Rails. Sin embargo, el objetivo de Ruote es Ruby, no solo Rails, por lo tanto el tratamiento de los desarrolladores ante esta

nueva herramienta se aleja de los mecanismos habituales para la generación ágil de aplicaciones web.

El motor en sí mismo es un intérprete de flujos de trabajo o procesos; concretamente la definición de dichos procesos es lo que se interpreta. Además el proyecto define un lenguaje propio de definición de dichos procesos. El lenguaje contiene cuatro constructores principales: process-definition, participant, sequence y concurrence

Se trata de un lenguaje conciso que no incorpora ningún tipo de información orientada al renderizado automático de los procesos o flujos, por tanto evita el ruido de datos de posicionamiento de cajas y flechas presente en otros lenguajes, ruido que distrae la atención del contexto real del proceso. Al evitarlo se dificulta la representación gráfica de los procesos, dificultad que ha sido parte importante de la problemática solventada en el desarrollo del editor gráfico de flujos de trabajo que se comentará en los próximos apartados.

En el proyecto ItecDesk se realiza una extensión del lenguaje para cubrir dos aspectos principales: la generación de formularios web, para permitir la implementación flexible de vistas desde el propio proceso de diseño, y operaciones básicas desde el lenguaje para acceder al modelo de datos.

Otros lenguajes de definición de procesos introducidos previamente, que sí incluyen información sobre visualización, son XPDL y BPML, pero esta información se hace compleja y crea confusión en el código de definición. Por ello se intentó desde el principio crear el nuevo lenguaje.

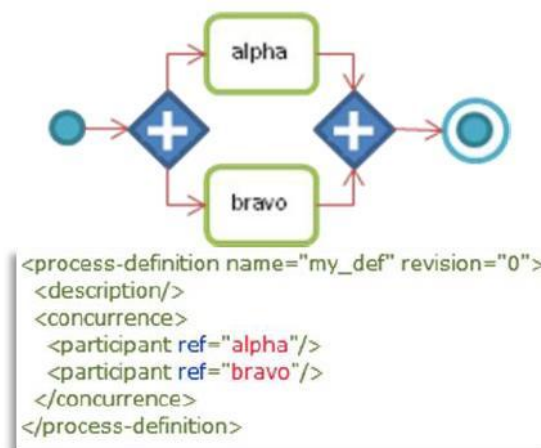


Figura 8. Definición de un flujo de trabajo concurrente simple.

Este proyecto no incluye un entorno de generación de código, ya que definir un proceso no es, siendo estrictos, programar, de la misma forma que no lo es el redactar una misión o una serie de tareas a ser ejecutadas dentro del proceso. Sin embargo sí se

trata de una labor propia de programadores el modelar los flujos de trabajo, ya que este tipo de profesional ha probado solventar mejor los problemas surgidos durante el modelado. Al tratarse estos flujos de trabajo de procesos orientados al mundo de la empresa, del negocio, parece ilógico el necesitar personal cualificado en programación para desarrollarlos; esta es la razón por la cual se buscó en el diseño del lenguaje un nivel de abstracción lo mayor posible. Esta abstracción no ha llegado al nivel de capacitar a cualquier persona para definir procesos, siendo necesaria una herramienta de definición gráfica de flujos de trabajo.

3.2.4 Servidor

En la arquitectura de ItecDesk, de estilo cliente-servidor, el servidor es el que proporciona un interfaz estándar de tipo REST para la comunicación externa con el motor de ejecución de flujos de trabajo. Además, es en el servidor donde se realiza la ejecución de funciones externas al motor declaradas en los propios flujos y mantiene comunicación con elementos como la bases de datos o el servicio de autenticación CAS. Está desarrollado utilizando tecnología Ruby anteriormente descrita.

3.2.5 Cliente

Siguiendo la arquitectura ya comentada de esta aplicación, en la que se realiza un recubrimiento del motor de flujos de trabajo con un interfaz REST que permite una interacción estándar con el mismo, la parte cliente de esta aplicación podría haberse implementado con diferentes tecnologías aprovechando la versatilidad de este tipo de interfaz.

En concreto, se optó por utilizar un cliente basado en web, ejecutable desde el navegador, para evitar al usuario final el proceso de instalación de una nueva aplicación en su máquina, y consiguiendo una característica multiplataforma. Como se ha comentado anteriormente, se seleccionó la tecnología Flash mediante el desarrollo en lenguaje de programación Flex, por permitir ésta la creación de un interfaz de usuario rico y completo, de fácil uso y de aspecto atractivo.

Las características más importantes del desarrollo en Flex son las encargadas de soportar la interacción con el interfaz REST del motor. Para ello, el cliente cuenta con la capacidad de interpretar toda la información de los recursos, que incluye datos sobre el flujo de trabajo en cada uno de sus estados e información sobre la presentación de las vistas. Toda esta información es intercambiada en formato XML. Con la información recogida el cliente es capaz de generar de forma dinámica vistosos formularios con los que el usuario final interactúa para continuar con el proceso de ejecución del flujo de trabajo. Tras cada interacción, se reenvía información con el mismo formato al interfaz REST y el motor actualiza el estado del flujo de trabajo y continúa con su ejecución.

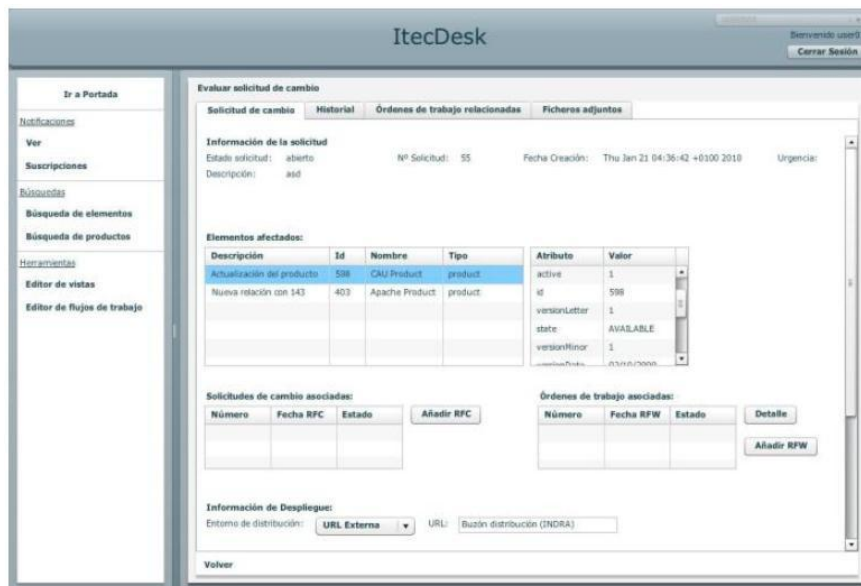


Figura 9. Captura de pantalla de la herramienta ItecDesk.

Cabe mencionar que la arquitectura de ItecDesk daría solución a otros escenarios; por ejemplo, si el encargado de recibir la información fuese una aplicación cliente basada en JavaScript, sería posible optimizar la transmisión de la información y el tratamiento de la misma, utilizando en la interfaz con el cliente que serializara los datos en JSON, en lugar de en XML. La razón es que JavaScript puede tratar de forma muy eficiente el formato JSON. Por otro lado, también es posible eliminar toda inteligencia del cliente, y enviar las vistas usando HTML. Sin embargo, esto aumenta la complejidad en el servidor y elimina la posibilidad, dentro de la vista, de ejecutar funciones anidables relacionadas con los workflows.

3.2.6 Generación Dinámica de Vistas

En el proyecto ItecDesk se lleva a cabo la extensión del lenguaje manejado por el proyecto OpenWFE para, entre otras mejoras, permitir la definición en los propios flujos de trabajo de los formularios o vistas con las que el usuario final interactuará.

Para conseguirlo se lleva a cabo la creación de un nuevo lenguaje, el Lenguaje Sencillo de Definición de Formularios (Simple Form Definition Language, SFDL) [40]. Éste fue diseñado con una serie de requisitos: soportar una amplia variedad de elementos de formularios web como selectores, tablas, desplegados y campos de entrada, ser autocontenido, capaz de soportar múltiples pantallas por vista, definible en diferentes lenguajes de marcado como XML, JSON e YAML, e incluir soporte para la ejecución de funciones de manejo de datos en la parte servidor.

SFDL no es el primer lenguaje de definición de formularios existente, conviviendo en este ámbito con otros. XForms [41], por ejemplo, fue descartado por carecer de soporte en los clientes web actuales y ser dependiente de marcado CSS. HTML 5.0

Forms [42] empieza a tener un soporte generalizado pero sus formularios son difícilmente serializables para tratarse con JavaScript, careciendo además de llamadas a funciones.

La definición de vistas mediante SFDL se realiza siguiendo un sencillo esquema basado en etiquetas que indican la posición, tipo, valor y parámetros extra de cada elemento.

```
screens:
- id: faqSearch
  layout_manager:
    type: grid
  tabs:
    1:
      id: main
      name: "%incident.default%"
    2:
      id: history
      name: "%history%"
    3:
      id: related
      name: "%incident.related%"
    4:
      id: attachments
      name: "%attachments%"
  fields:
    - id: "0102"
      type: label
      value: "%solution.selectFromTable%"
      params:
        weight: ""
        style: ""
        halign: left
        width: 750
    - id: "0202"
      type: dynamic_table
      value:
        headers:
          1:
            id: id
            name: "%id%"
          2:
            id: description
            name: "%description%"
          3:
            id: solution
            name: "%solution.default%"
          4:
            id: related_element_ids
            name: "%related_element_ids%"
        params:
          hint: ""
          width: 750
          height: 200
          filter_by_status: %status.closed%
          selectable: one
    - id: "2502"
      type: button
      value: change_view
      params:
        button_text: %button.accept%
        screen id: "default"
```

Figura 10. Ejemplo de definición de vista mediante SFDL.

El acceso al modelo de datos se realiza mediante la adición de funciones ejecutables en tiempo de procesamiento del flujo o bien en tiempo de presentación del formulario.

Para ayudar a la generación de vistas se ha desarrollado de forma paralela al proyecto ItecDesk un editor gráfico de vistas que, utilizando tecnología Flex y aprovechando su capacidad drag&drop, facilita enormemente la labor de definir nuevos formularios que incluir a un flujo de trabajo.

3.2.7 Editor Gráfico de Flujos de Trabajo

Como apoyo a la herramienta ItecDesk y para facilitar la creación y definición de flujos de trabajo, se realizó el desarrollo de un editor gráfico interactivo para aplicaciones genéricas de flujos de trabajo. En más detalle, se trata de una aplicación web que permite al usuario generar de forma gráfica y a partir de una paleta de elementos un flujo de trabajo para obtener a continuación su definición en código soportado por el motor de flujos de trabajo Ruote. Esta definición puede ser salvada en el disco local del usuario para su posterior uso. También permite el sentido contrario de la acción; un usuario puede cargar en la aplicación un flujo de trabajo definido en

lenguaje SFDL y observar su forma gráfica con la consiguiente opción de editar el flujo mostrado.

Con la existencia de este editor gráfico de flujos de trabajo se consigue aislar al usuario final de la problemática de programar un flujo de trabajo. Con ello se posibilita a usuarios no expertos a participar en la definición de flujos de trabajo, siendo estos usuarios muy habituales en ámbitos de negocio en los que comienzan a ser muy populares los flujos de trabajo.

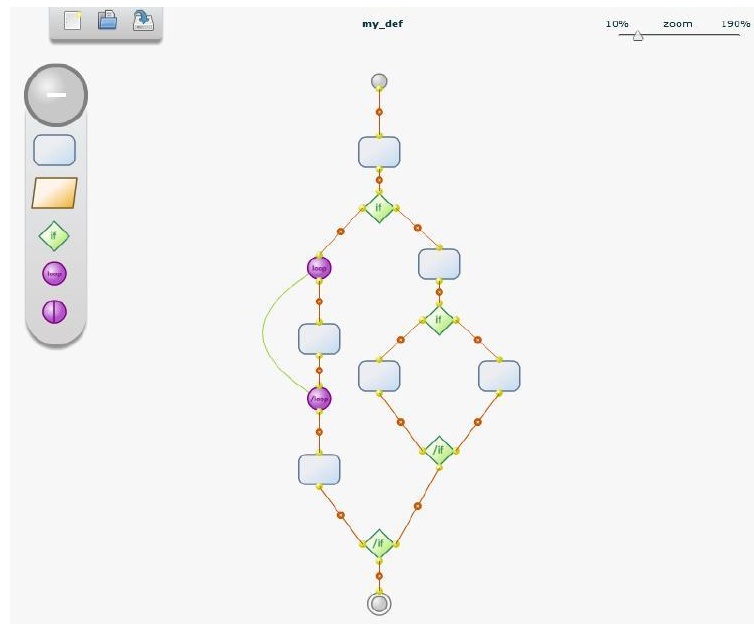


Figura 11. Captura del editor gráfico de flujos de trabajo.

3.2.8 Contexto Conseguido

La herramienta ItecDesk está orientada al desarrollo de software distribuido apoyado en la interacción organizada entre participantes mediante la acción de un flujo de trabajo.

El tipo de contexto conseguido es el contexto orientado a tareas, ya que cada tarea, entendida como un desarrollo software o una parte concreta del mismo, es modelada como un proceso y ejecutada en el motor de flujos de trabajo.

Una vez arrancado un nuevo proceso o flujo de trabajo, en este caso por la necesidad de una nueva tarea de desarrollo, el lanzador del proceso queda registrado y estará al tanto de los eventos relacionados con la evolución del desarrollo. Es el motor de flujos de trabajo el encargado de, posteriormente, trasladar la acción a un participante que, por su especialidad, sea capaz de aportar el trabajo e información necesarios para que la acción continúe hacia un nuevo participante. Todos los implicados en el flujo de trabajo serán notificados, con un nivel de notificación completamente configurable, de los eventos relevantes que se den durante el proceso.

De este modo se mantiene el contexto o conciencia de todo el proceso por parte de todos los participantes, a pesar de que no estén relacionados por su tipo de actividad o su localización.

Además, todo flujo de trabajo lleva asociadas anotaciones acumulativas en forma de comentarios que permiten la comunicación explícita textual entre participantes de forma óptima en comparación con la habitual comunicación por correo electrónico mediante listas de correo. Este aumento en eficiencia de la comunicación, y por tanto en el mantenimiento del contexto, está relacionada con el hecho de que se produce en el propio ámbito de desarrollo en el que, además, se puede acceder directamente a los artefactos propios de la implementación, ya que, al igual que los comentarios, se asocian al flujo de trabajo todos los documentos que sean relevantes en forma de ficheros adjuntos.

Con la inclusión de la capacidad para ejecutar funciones externas al motor de flujos de trabajo desde los propios procesos, se consigue mejorar el contexto dando la capacidad a los distintos participantes de escribir en base de datos o de añadir los artefactos de desarrollo en un buzón de compilación entre otras acciones.

La coordinación del trabajo entre participantes de un proceso de desarrollo distribuido es por tanto conseguida mediante la inclusión de la herramienta ItecDesk, que añade además mecanismos para mantener el contexto, consiguiendo la optimización del proceso en tiempo y trabajo.

3.3 Recomendadores basados en Contexto

Los sistemas de recomendación, como se ha observado a lo largo de este documento, no son un tipo de sistemas novedoso: su desarrollo ha llegado a un momento de relativa madurez por albergar un largo período de investigación.

Sin embargo, se ha apuntado que existen una serie de problemas aún por resolver y mucho por refinar, tareas que siguen haciendo de éste un campo de activa investigación.

Una de las vías de investigación de mayor actualidad es la inclusión de nuevas entradas como alimentación de los sistemas recomendadores para conseguir mejores resultados. Estas entradas vienen de diferentes ámbitos como son: la web semántica, el contexto de los usuarios y las redes sociales [43][44].

A lo largo de este apartado se introducen las ideas relativas a cada una de las líneas de investigación mencionadas para culminar en la explicación de un proyecto de motor de recomendación avanzado llevado a cabo de forma paralela a las tareas descritas en estas páginas.

3.3.1 Web Semántica

La tecnología de web semántica está volviendo al entorno actual a pesar de haber sufrido un decrecimiento en su actividad en los últimos años. En el caso de los sistemas recomendadores esta tecnología puede soportar el monitorizado de las acciones llevadas a cabo por los usuarios del sistema y, en general, facilitar la recolección y gestión de información a lo largo de la web.

Existen proyectos como Linking Open Data para hacer disponibles conjuntos de datos útiles para la realización de recomendaciones mediante su definición en formato RDF, y su valor frente a soluciones centralizadas queda demostrada al poder utilizar mecanismos automatizados para la interconexión de los distintos conjuntos de datos.

Un caso de estudio es Ifanzy que recoge contenido de forma personalizada para cada usuario usando como base un sistema que utiliza las tecnologías de la web semántica para establecer y gestionar las estructuras

de conocimiento con las que trabajar. El contenido procesado tiene que ver con programación televisiva personalizada. Una de las características propias de esta tecnología utilizadas en este proyecto es la capacidad de utilizar diferentes fuentes de información heterogéneas.

Una de las fuentes que están dotando de un importante éxito a los sistemas recomendadores es el uso de etiquetas o tags [45]. El etiquetado es utilizado para conseguir una conexión entre usuarios y elementos generada por los propios usuarios; de esta manera se consigue también una clasificación de los elementos a recomendar en función del significado de las etiquetas, que puede ir orientado a representar opiniones o características. Para utilizar estos sistemas ha de establecerse la diferencia entre la relevancia de una etiqueta y la preferencia del usuario que la utiliza (el modo en que la utiliza, en función de la valoración positiva o negativa que la acompaña).

3.3.2 Contexto Físico

En la actualidad, y gracias a la utilización cada vez más masiva de dispositivos capaces de captar el contexto físico de los usuarios como son los Smartphones con sistemas como GPS, cámara, brújula, etc., la utilización del contexto en aplicaciones de distinto carácter es un tema de investigación muy intenso.

El contexto no ha sido una entrada habitual en los sistemas recomendadores, pero ya existen estudios para la inclusión de este aspecto junto al resto de inputs. En la representación multidimensional habitual en la que se simbolizan los usuarios y los elementos en una misma tabla, es posible incluir el contexto como un nuevo eje pasando de la representación bidimensional típica a una tridimensional, como puede observarse en la figura 7.

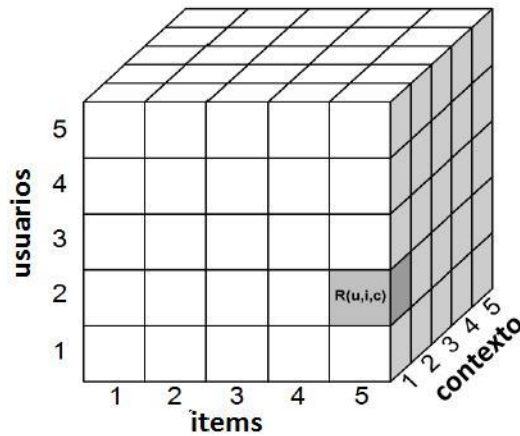


Figura 12. Dimensiones de un sistema recomendador con contexto.

Es posible nombrar algunos ejemplos que clarifican la utilidad del contexto en los sistemas recomendadores. En primer lugar, conocer el momento concreto en el que el usuario está solicitando una recomendación o la hora para la que la desea es interesante de cara a proponerle, por ejemplo, un restaurante óptimo para cenar. Otro ejemplo es la obtención de la localización del usuario para recomendarle comercios que se encuentren cerca de él.

3.3.3 Redes Sociales

La utilización de los contenidos sociales de los usuarios finales para mejorar las recomendaciones es otro tipo de contexto. El contexto social es una clave determinante a la hora de identificar las acciones y la necesidad de información de los usuarios finales.

Una forma de utilizar el contexto social es disminuyendo el peso que se le da a la obtención automática de “vecinos” para identificar usuarios de gustos similares y pasar a un modelo mixto que incluya las relaciones sociales de un usuario con otros patentes en distintas redes sociales.

Otra manera de incluir esta nueva herramienta es darle mayor relevancia a las valoraciones hechas por usuarios que tengan relación con el usuario a recomendar.

Se están llevando a cabo gran cantidad de estudios en este aspecto y las primeras evaluaciones llevan a la idea de que la inclusión del contexto social en las recomendaciones mejora los resultados obtenidos, llegando a utilizar recomendadores híbridos [46].

Sin embargo existe un problema ya comentado en secciones anteriores relacionado con la privacidad [47]. Para utilizar datos de contexto social de los usuarios es necesario acceder a las bases de datos de distintas redes sociales con los problemas legales que esto implica.

3.3.4 Ontologías, Taxonomías y Folksonomías

Todas las técnicas comentadas en los apartados anteriores están soportadas por una serie de tecnologías subyacentes de ordenación, clasificación y representación de información que tienen sus raíces en la web semántica y los sistemas inteligentes.

Por un lado se encuentran las ontologías, que son esquemas conceptuales de un dominio concreto y que sirven para facilitar la comunicación e intercambio de información entre distintos sistemas y entidades. En el ámbito de los sistemas recomendadores, esta tecnología es usada para realizar categorizaciones con sentido semántico y para habilitar el uso de la web semántica como entrada de datos de fuentes heterogéneas. La captura de contexto, no sólo físico sino también lógico relacionado con las preferencias de cada usuario, es a menudo representada mediante ontologías [48].

Taxonomías y folksonomías son utilizadas para generar clasificaciones. Las primeras tratan las clasificaciones más habituales en forma de árbol con agrupaciones por similitud; el contar con este tipo de organización de los elementos es útil ya que aligera la tarea de la recomendación en sí misma y permite tratar los elementos o ítems en un espacio de dimensiones más manejable, de manera que potencia estos sistemas [49]; además, permite solventar algunos problemas identificados como el de cold start [50]. Las folksonomías son otro tipo de clasificación que está tomando mucho impulso en la actualidad; se trata indexación social o clasificación colaborativa, en la que, mediante el uso de etiquetas o tags [51] en un espacio de nombres a menudo no controlado o restringido, sin relaciones preestablecidas, se consigue una categorización de calidad y con información añadida acerca de los usuarios que etiquetan. Por esta última capacidad explicada, las folksonomías son parte hoy de un buen número de sistemas recomendadores que instan a sus usuarios al uso de etiquetas sobre los elementos recomendados para conseguir feedback, establecer relaciones entre elementos e identificar la manera en que cada usuario valora dichos elementos.

3.3.5 Otras Posibilidades

Son muchas las líneas en las que se está avanzando, y es diferente la metodología estudiada en cada caso concreto de uso.

Una de las líneas, ya introducida anteriormente, es la utilización de un modelo de recomendación denominado “de grupos”. Está tomando un renovado impulso el desarrollo de métodos y sistemas de clustering que consigan asociar a usuarios en función de su perfil y generar grupos de “gustos” similares. De esta manera, y una vez obtenidos los “grupos”, se reduce de manera muy significativa el tiempo de cómputo para obtener las recomendaciones, ya que basta con extraer los elementos mejor valorados por parte del total de los usuarios en un “grupo”.

Para conseguirlo es esencial el perfeccionamiento de los algoritmos de agrupación existentes en la actualidad (K-Means, Canopies, EM, etc.) y el estudio de la optimización de sus parámetros de entrada.

A parte del plano técnico, si se quiere llegar a dividir a la población a la que va dirigida un servicio recomendador concreto en grupos, es necesario contar con la ayuda de personal de formación sociológica, de manera que se establezca la corrección de un número u otro de grupos, del tamaño de los mismos y de su contenido, empresa nada sencilla pero clave en el funcionamiento final del sistema recomendador.

3.3.6 El Proyecto Galactus

Todos los conocimientos adquiridos al realizar el estado del arte de recomendadores plasmado en la primera parte de este documento, así como la investigación acerca de las líneas de investigación más activas y avanzadas en la mejora de este tipo de sistemas, comentadas en los apartados anteriores, han sido clave en la creación de un proyecto de motor de recomendación avanzado: Galactus.

Se encuentra en sus primeras fases de desarrollo y su objetivo principal es incluir en la creación de las recomendaciones diferentes técnicas punteras ya introducidas en este documento para conseguir unas recomendaciones de gran riqueza y calidad.

Por ejemplo, el acceso a la aplicación de recomendación va a utilizar la potencia de los sistemas de identidad única, tanto OpenID como otros que están en pleno apogeo como puede ser FacebookConnect o Twitter. De esta manera se consigue una mayor tasa de registros y la percepción del usuario es la de no tener que realizar un nuevo registro para disfrutar de esta nueva funcionalidad, ya que está utilizando una cuenta ya creada previamente. Con ello se asegura una mejor usabilidad y se alinea el proyecto con la línea de las aplicaciones y servicios más avanzados.

Por otro lado, la generación de recomendaciones parte de la utilización de algoritmos de especial interés por soportar grandes cantidades de usuarios y ofrecer grandes posibilidades de optimización en el tiempo. En concreto se ha decidido utilizar algoritmos de clustering que agrupen a los usuarios en grupos de gustos similares para que la exploración previa a cada recomendación pueda realizarse de una forma rápida. Además, se incluye la idea del contexto a la hora de elegir la recomendación a mostrar; este contexto es tanto físico, obteniendo información de localización, hora, y entorno físico en general, hasta social, teniendo en cuenta los mecanismos de generación de contexto explicados previamente en este mismo documento.

De esta manera, y aunque aún está en sus primeros estadios de desarrollo, se observa ya una mejora frente a otros sistemas de recomendación, habiéndose evitado problemas tan comunes como el cold-start.

4 Conclusiones y Trabajos Futuros

En esta sección se detallan las conclusiones a las que se ha llegado tras realizar cada uno de los desarrollos explicados a lo largo de este documento y se introducen los posibles trabajos futuros a realizar en cada una de las áreas. De esta manera, y al tratarse de desarrollos concretos, relacionados pero disjuntos en su implementación, se describirán las conclusiones y trabajos futuros de cada uno de ellos siguiendo el esquema respetado a lo largo de todo el escrito.

4.1 Identidad Digital y OpenID

Parece evidente la utilidad de un sistema distribuido de gestión de identidad única que permita a los usuarios de diferentes servicios mantener un solo par de “usuario y contraseña”, ya que aumenta la usabilidad de los sistemas e implementar nuevas funcionalidades relacionadas con la compartición de información entre servicios y la presencia de datos actualizados de usuario.

El protocolo referente en cuanto a adopción para conseguir esta meta es OpenID. Surgido como solución ligera, su implementación no tiene en cuenta una serie de problemas de seguridad que tienen mucha importancia cuando se plantea la utilización de este sistema como medio de autenticación a nivel global. Servicios críticos como los bancarios o sanitarios pueden verse comprometidos por estos problemas, lo que está frenando la extensión de OpenID.

Las soluciones no son a priori sencillas pero existe una activa vía de investigación para hacer de OpenID una solución real de futuro. Los desarrollos actuales para solventar los problemas identificados pasan por la realización de evoluciones del protocolo, la inclusión de una serie de mecanismos para garantizar la confianza que se puede tener en uno u otro proveedor, el establecimiento de técnicas para luchar contra el phishing por tratarse de uno de los ataques a los que es más vulnerable OpenID y la adopción de una serie de buenas prácticas por parte de los proveedores de este servicio.

En la actualidad el futuro inmediato de este tipo de sistemas no está claro, ya que algunos desarrollos como FaceBook Connect o el importante esfuerzo por la imposición de OAuth y su crecimiento, hacen plantearse a algunos grupos de trabajo la dirección a seguir. De cualquier manera la funcionalidad buscada está clara y es la de conseguir un entorno Internet más cómodo de utilizar para que los usuarios finales adopten de forma sencilla un mayor número de servicios.

4.2 Generación de Contexto a partir de Flujos de Trabajo

La herramienta presentada para ayudar al mantenimiento del contexto y mejorar la coordinación en el desarrollo de software en entornos de trabajo distribuido ha sido

desplegada en un entorno bancario. Para conseguir implementaciones futuras orientadas a la mejora del mantenimiento de contexto, sería necesario realizar un seguimiento a los beneficios obtenidos del uso de la herramienta, introduciendo la misma en un número mayor de entornos y realizando un estudio detallado de sus efectos en cada una de las fases del proyecto. Una forma acertada de enfocarlo sería realizando entrevistas a los diferentes participantes del proceso y estudiar el uso que le dan a la herramienta tal y como se hace en ejercicios previos similares [52].

Sería interesante estudiar la posibilidad de incluir en la herramienta componentes orientados al mantenimiento de conciencia de grupo como puede ser el chat o la videoconferencia. Una funcionalidad frecuentemente demandada por los desarrolladores en grupos distribuidos es el acceso rápido, sencillo, e integrado con el resto de componentes al repositorio de artefactos. Esto es ya parcialmente incluido mediante la capacidad para mantener un acumulado de archivos adjuntos en un flujo de trabajo, pero podría aumentarse creando un acceso directo a repositorio mediante un componente integrado, limitado por los permisos propios de cada usuario.

Un trabajo futuro de especial interés pero de gran complejidad es la integración de los principales entornos de desarrollo (IDEs) con las herramientas de mantenimiento de contexto como la presentada en este documento o viceversa. Este trabajo contribuiría a conseguir un entorno completo.

Por último, sería posible generar una serie de plantillas de flujos de trabajo orientados a cubrir las tareas fuertemente definidas y estáticas que aparecen en todo proceso de desarrollo de software, en mayor o menor medida, en función del modelo de desarrollo que se esté utilizando. Con estas plantillas sería mucho más rápido el diseño de nuevos flujos de trabajo aplicables a ámbitos concretos, y sería totalmente compatible con el editor gráfico de flujos de trabajo comentado previamente.

Es necesario incorporar una herramienta flexible, potente y de fácil adopción a los grupos de desarrollo de software distribuidos, de manera que se gestione la coordinación del grupo, la toma de decisiones, y la repartición de tareas de forma eficiente.

Todas las alternativas existentes no han sido aceptadas por conllevar una excesiva complejidad en comparación con los beneficios obtenidos a partir de ellas.

La tecnología de flujos de trabajo es un campo maduro, de importancia emergente, que proporciona un amplio abanico de ventajas en diferentes ámbitos. Su aplicación en el campo del mantenimiento de contexto en grupos de trabajo distribuidos no ha sido explotada hasta el momento debido a que el uso habitual de los flujos de trabajo ha

sido automatizar y optimizar procesos rígidos, muy bien definidos y poco cambiantes, características que no comparten los procesos propios del desarrollo software.

Se ha realizado la implementación de una herramienta, ItecDesk, que permite la gestión de flujos de trabajo editables en tiempo real, evitando así el problema de excesiva rigidez inherente de esta tipo de tecnología. De este modo se pueden definir diferentes procesos para cada caso concreto de uso.

Cada proceso o flujo de trabajo será gestionado por el motor de flujos de trabajo haciendo que en cada momento sea el usuario más adecuado, por su experiencia, el encargado de realizar una acción sobre el mismo. De este modo se aceleran las tareas y se evitan confusiones o trabajos duplicados.

Además, ItecDesk aporta un importante contexto orientado a tareas, en la que cada participante de un proceso es consciente de las acciones que realizan sus compañeros y de la evolución que sufre el grupo de artefactos en los que está trabajando.

Por último, esta herramienta ha sido desplegada en un entorno bancario, lo que implica que se ha tenido muy en cuenta el aspecto de seguridad e identidad. También supone que el grado de madurez de la herramienta es alto, siendo ésta fácilmente trasladable a otros ámbitos de ejecución.

4.3 Recomendadores Basados en Contexto

El entorno de los sistemas de recomendación es un medio de gran actividad a pesar de ser ésta una tecnología ya muy estudiada y relativamente madura. La aplicación de este tipo de sistemas en el ámbito de negocio hace que se intenten mejorar continuamente, ya que son la base de algunos modelos de negocio, siendo un ejemplo claro el comercio electrónico de música o literatura.

La clasificación de los sistemas de recomendación en función de su naturaleza es necesaria para establecer las distintas formas de “atacar” un problema según sea el resultado deseado [53]. Esta clasificación puede tomar varias formas, desde una sencilla que los divide entre “basados en usuario” y “basados en ítem” hasta otras más completas como la presentada en este documento.

Los algoritmos utilizados son la pieza de los sistemas recomendadores. Existen algoritmos para realizar la asociación de usuarios o elementos en “vecindades” y los algoritmos orientados a calcular la recomendación. Generan un campo de investigación en sí mismos.

Los sistemas actuales se encuentran con una serie de los problemas con los que ya se encontraban los primeros desarrollados, muchos de ellos relacionados con la propia naturaleza de los datos tratados. A nivel conceptual no es sencillo realizar una

recomendación precisa a no ser que se tengan muchos datos, y esto no es a menudo garantía de éxito.

Uno de los problemas más habituales es la dificultad de evaluar la calidad del sistema recomendador desarrollado, existiendo un gran número de métricas u objetivos a cumplir y conjuntos de datos públicos y muy estudiados sobre los que se pueden aplicar distintos sistemas para intentar establecer cuánto de buenos son.

Por último, quizá lo más interesante sea la nueva corriente de trabajo que se está llevando a cabo para mejorar los sistemas recomendadores a partir de datos de entrada que hasta ahora no habían sido tenidos en cuenta. La utilización de la potencia de la web semántica para automatizar la interrelación de datos y mejorar la explotación de fuentes de información, y la inclusión de contexto físico y social de los usuarios finales a la hora de recomendar, han probado ser vías para optimizar dicha recomendación.

4.4 Conclusiones Generales

El trabajo realizado y plasmado en este documento ha servido para analizar una serie de dominios a priori disjuntos para llegar a conjugarlos y conseguir con la potencia de cada uno mejorar unos sistemas tan importantes y activos en el entorno actual como son los motores de recomendación.

La idea final obtenida es que la unión de estas tecnologías puede abrir una nueva línea de investigación para la optimización no sólo de sistemas de recomendación, sino de cualquiera que pueda beneficiarse de la identidad digital única y de información acerca del contexto físico y social de sus usuarios.

De forma personal, este trabajo me ha servido para aumentar mis conocimientos sobre los temas tratados y para orientar mi ámbito de estudio futuro. He encontrado interesante el estudio de documentación de investigación que presenta el estado de las investigaciones más novedosas y de los primeros desarrollos realizados utilizando tecnologías de nueva aparición.

Bibliografía

- [1] D. Mashima and M. Ahamad, "Towards a User-Centric Identity-Usage Monitoring System," *2008 The Third International Conference on Internet Monitoring and Protection*, Bucharest : IEEE, 2008, pp. 47-52.
- [2] S. Rieger, "User-Centric Identity Management in Heterogeneous Federations," *2009 Fourth International Conference on Internet and Web Applications and Services*, Venice-Mestre: IEEE, 2009, pp. 527-532.
- [3] W. Bin, H.H. Yuan, L.X. Xi, and X.J. Min, "Open Identity Management Framework for SaaS Ecosystem," *2009 IEEE International Conference on e-Business Engineering*, Macau: IEEE, 2009, pp. 512-517.
- [4] C. Europarc, P. Dourish, and V. Bellotti, "Awareness and Coordination in Shared Work Spaces," *Conference on Computer-Supported Cooperative Work*, ACM, Toronto: 1992, pp. 107-114.
- [5] C. Gutwin and S. Greenberg, "A Descriptive Framework of Workspace Awareness for Real-Time Groupware," *Computer Supported Cooperative Work*, vol. 11, 2002.
- [6] J. Herbsleb and R. Grinter, "Architectures, coordination, and distance: Conway's law and beyond," *IEEE Software*, vol. 16, 1999, pp. 63-70.
- [7] T. Schümmer, "Lost and Found in Software Space," *HICSS*, 2001.
- [8] J. Froehlich and P. Dourish, "Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams," *International Conference on Software Engineering*, Washington, DC, USA: IEEE Computer Society, 2004, pp. 387 - 396.
- [9] "Workflows." [Online]. Available: <http://www.ccee.edu.uy/ensenian/catsistc/docs/Workflow.pdf>. [Accessed: 1/03/2010].
- [10] "XPDL. Workflow Management Coalition." Available: <http://www.wfmc.org/xpdl.html> [Accessed: 1/03/2010].
- [11] "BPEL. Oasis." Available: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel [Accessed: 2/03/2010].
- [12] "Eclipse BPMN Modeler." Available: http://www.eclipsecon.org/summiteurope2006/presentations/ESE2006-EclipseModelingSymposium1_BPMS&Eclipse.pdf [Accessed: 1/03/2010].
- [13] "Ruote." Available: <http://ruote.rubyforge.org/> [Accessed: 2/03/2010].
- [14] "Ruote." Available: <http://ruote.rubyforge.org/> [Accessed: 2/03/2010].
- [15] P. Groth and Y. Gil, *Analyzing the Gap between Workflows and their Natural Language Descriptions*, IEEE, 2009.
- [16] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, 2005, pp. 734 - 749 .
- [17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," *Electronic Commerce*, Minneapolis, Minnesota, United States: 2000, pp. 158-167.

- [18] J. Zhan, C. Hsieh, I. Wang, T. Hsu, C. Liau, and D. Wang, "Privacy-Preserving Collaborative Recommender Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. PP, 2010, pp. 1-6.
- [19] C.A. Hsieh, J. Zhan, D. Zeng, and F. Wang, "Preserving Privacy in Joining Recommender Systems," *2008 International Conference on Information Security and Assurance (isa 2008)*, Busan : IEEE, 2008, pp. 561-566.
- [20] E. Campochiaro, R. Casatta, P. Cremonesi, and R. Turrin, "Do Metrics Make Recommender Algorithms?," *2009 International Conference on Advanced Information Networking and Applications Workshops*, Bradford : IEEE, 2009, pp. 648-653.
- [21] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, 2004, pp. 5 - 53 .
- [22] Z. Zaier, R. Godin, and L. Faucher, "Evaluating Recommender Systems," *2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*, Florence : IEEE, 2008, pp. 211-217.
- [23] "MovieLens - GroupLens." <http://www.grouplens.org/node/73> [Accessed: 12/01/2010].
- [24] "Jester Collaborative Filtering Dataset." <http://goldberg.berkeley.edu/jester-data/> [Accessed: 5/03/2010].
- [25] "Netflix." www.netflix.com/ [Accessed: 1/02/2010].
- [26] "Book-Crossing Dataset." www.informatik.uni-freiburg.de/~chiegler/BX/ [Accessed: 5/02/2010].
- [27] H. Oh and S. Jin, "The Security Limitations of SSO in OpenID," *2008 10th International Conference on Advanced Communication Technology*, Gangwon-Do : IEEE, 2008, pp. 1608-1611.
- [28] H. Oh and S. Jin, "The Security Limitations of SSO in OpenID," *2008 10th International Conference on Advanced Communication Technology*, Gangwon-Do : IEEE, 2008, pp. 1608-1611.
- [29] H. Lee, I. Jeun, K. Chun, and J. Song, "A New Anti-phishing Method in OpenID," *2008 Second International Conference on Emerging Security Information, Systems and Technologies*, Cap Esterel : IEEE, 2008, pp. 243-247.
- [30] "VeriSign Personal Identity Portal.": <https://pip.verisignlabs.com/> [Accessed: 1/03/2010].
- [31] "VeriSign and Firefox's seatBelt.": <https://pip.verisignlabs.com/>[Accessed: 12/04/2010].
- [32] "Vidoop - Confident Technologies.": www.confidenttechnologies.com [Accessed: 12/05/2010].
- [33] "Jabber.": <http://www.jabber.org/> [Accessed: 15/03/2010].
- [34] I. Jorstad, T.A. Johansen, E. Bakken, C. Eliasson, M. Fiedler, and v.T. Do, "Releasing the potential of OpenID & SIM," *2009 13th International Conference on Intelligence in Next Generation Networks*, Bordeaux : IEEE, 2009, pp. 1-6.
- [35] ICAM, "Federal Identity, Credentialing, and Access Management - OpenID 2.0 Profile," 2009.

- [36] N. Yan, "Make the Consumable Services via REST," 2008 IEEE International Conference on Web Services, IEEE, 2008, pp. 12-12.
- [37] Y. Zhang, C. Yang, J. Liu, and J. Tian, "An authentication scheme for three-party mobile Conditional Access system(CAS)," 2009 Fourth International Conference on Communications and Networking in China, IEEE, 2009, pp. 1-5.
- [38] "Usability for rich internet applications. Digital Web Magazine.." Available: http://www.digital-web.com/articles/usability_for_rich_internet_applications/ [Accessed: 1/03/2010].
- [39] E.A. Wohed P., Andersson B., "Patterns-based Evaluation of Open Source BPM Systems: The Cases of jBPM, OpenWFE, and Enhydra Shark," 2007.
- [40] D. Moreno, E. García, S. Aguirre, and J. Quemada, "Extensiones de Lenguaje de Workflow para la Generación Dinámica de Vistas," Cita 2009, Gijón: 2009, pp. 45 - 52.
- [41] "W3C XForms 1.0 (Third Edition)." Available: <http://www.w3.org/TR/xforms/> [Accessed: 20/02/2010].
- [42] "W3C HTML 5." Available: <http://dev.w3.org/html5/spec/> [Accessed: 2/03/2010].
- [43] W. Woerndl and G. Groh, "Utilizing Physical and Social Context to Improve Recommender Systems," 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, Silicon Valley, CA : IEEE, 2007, pp. 123-128.
- [44] P. Bellekens, G. Houben, L. Aroyo, K. Schaap, and A. Kaptein, "User model elicitation and enrichment for context-sensitive personalization in a multiplatform tv environment," European Conference on Interactive Television, Leuven, Belgium: 2009, pp. 119-128 .
- [45] S. Jian, H. Liang, and L. Xin, "Improving the accuracy of Tagging Recommender System by Using Classification," Advanced Communication Technology (ICACT), 2010 The 12th International Conference on, Phoenix Park : 2010, pp. 387-391.
- [46] M. Ghazanfar and A. Prugel-Bennett, "A Scalable, Accurate Hybrid Recommender System," 2010 Third International Conference on Knowledge Discovery and Data Mining, Phuket : IEEE, 2010, pp. 94-98.
- [47] K. Pussep, S. Kaune, J. Flick, and R. Steinmetz, "A Peer-to-Peer Recommender System with Privacy Constraints," 2009 International Conference on Complex, Intelligent and Software Intensive Systems, Fukuoka : IEEE, 2009, pp. 409-414.
- [48] R.F. Jr., B.C. Furtado, F. Maximino, R.G. Cattelan, and M.d. Pimentel, "Context information exchange and sharing in a peer-to-peer community: a video annotation scenario," ACM Special Interest Group for Design of Communication, The Evolution of TV Systems, Content, and Users Toward Interactivity: 2009, pp. 265-272 .
- [49] C. Ziegler, G. Lausen, and J.A. Konstan, "On exploiting classification taxonomies in recommender systems," AI Communications, vol. 21, 2008, pp. 97-125 .
- [50] L. Weng, Y. Xu, Y. Li, and R. Nayak, "Exploiting Item Taxonomy for Solving Cold-Start Problem in Recommendation Making," 2008 20th IEEE International Conference on Tools with Artificial Intelligence, Dayton, OH: IEEE, 2008, pp. 113-120.
- [51] B. Sigurbjörnsson and R.v. Zwol, "Flickr tag recommendation based on collective knowledge," International World Wide Web Conference, Beijing: 2008, pp. 327-336 .

- [52] C. Gutwin, R. Penner, and K. Schneider, "Group awareness in distributed software development," *Computer Supported Cooperative Work*, Chicago, Illinois, USA : ACM New York, NY, USA , 2004, pp. 72 - 81.
- [53] M. Montaner, B. López, and J.L. de la Rosa, "A Taxonomy of Recommender Agents on the Internet," *Artificial Intelligence Review*, vol. 19, 2003, pp. 285-330.