

Soporte al inicio de colaboración en el desarrollo distribuido de software mediante esferas de trabajo colaborativas

Ramón René Palacio Cinco¹, Alberto L. Morán y Solares²
y Víctor Manuel González y González³

¹ Instituto Tecnológico de Sonora, Unidad Navojoa,
México

² Facultad de Ciencias, Universidad Autónoma de Baja California,
México

³ Instituto Tecnológico Autónomo de México,
México

ramon.palacio@itson.edu.mx, alberto.moran@uabc.edu.mx, victor.gonzalez@itam.mx

Resumen. Este artículo estudia los entornos de trabajo donde grupos de personas interactúan de manera síncrona y remota (distribuida), con el propósito de crear y desarrollar software dentro del marco institucional de una organización, en lo que se conoce como desarrollo distribuido de software (DSD, por sus siglas en inglés). En este tipo de esquemas colaborativos, los desarrolladores requieren trabajar en grupos que están geográficamente distribuidos y su interacción generalmente es realizada con el apoyo de tecnología de información y comunicación. Es común que las tecnologías de colaboración no estén diseñadas para apoyar lo que llamamos *inicio de colaboración informado*, es decir los escenarios donde el iniciador de la colaboración pueda contar con la información de la actividad que realiza la persona buscada, con la cual, el iniciador pueda inferir si el momento para iniciarla es óptimo y apropiado para ambos. Para lograr esto, es necesario conocer el contexto de la actividad de la persona buscada en un momento determinado. Para apoyar el inicio de colaboración informado se propone la conceptualización y caracterización tecnológica de esferas de trabajo colaborativas, la cual aporta ideas de diseño para el desarrollo de una herramienta prototipo. A esta herramienta le llamamos CWS-IM (Collaborative Working Spheres – Instant Messaging). La herramienta es un mensajero instantáneo extendido con soporte para inicios de colaboración informados, la cual se introdujo a las actividades reales de DSD en una fábrica de software con la finalidad de evaluarla mediante un estudio de caso. Los resultados de esta evaluación proporcionan evidencia que muestra la aceptación favorable de CWS-IM por parte de los

participantes en términos de utilidad, facilidad de uso, apoyo al inicio de interacción y gestión del nivel de interrupción.

Palabras Clave: Sistemas Colaborativos, Desarrollo Distribuido de Software, Manejo de Interrupciones, Ingeniería de Software

Support for Starting Collaboration in Distributed Software Development through Collaborative Working Spheres

Abstract. This paper studies work environments where groups of people interact synchronously and remotely (distributed) with the purpose of creating and developing software within the institutional framework of an organization, in what is known as distributed software development (DSD). In this type of collaborative schemes, developers are required to work in groups that are geographically distributed and their interaction is usually conducted with the support of information and communication technologies. It is common that collaborative technologies are not designed to support what we call informed collaboration initiation, i.e., scenarios where the initiator of the collaboration can have the information of the activity that is being performed by the person he/she is looking for, with which the initiator can infer whether the time to start is optimal and appropriate for both of them. To achieve this it is necessary to know the context of the activity of the person that is being looked for at a given moment. To support the initiation of informed

collaboration, the conceptualization and technological characterization of collaborative working spheres is proposed, which provides design ideas for the development of a prototype tool. We call this tool CWS-IM (Collaborative Working Spheres - Instant Messaging). The tool is an extended instant messenger with support for the initiation of informed collaboration, which was introduced to the actual activities of DSD in a software factory in order to evaluate it through a case study. The results of this evaluation show the favorable acceptance of CWS-IM by participants in terms of usefulness, ease of use, support for the initiation of informed interaction and disruption level management.

Keywords: Collaborative systems, distributed software development, interruption management, software engineering.

1. Introducción

Actualmente, la industria del software se está enfrentando a un cambio en la forma de desarrollar software, el cual es conocido como desarrollo distribuido de software (DSD, por sus siglas en inglés). Este cambio crea situaciones de las cuales las organizaciones pueden beneficiarse, y retos a los que deben adaptarse (p. ej. la falta de oportunidades para la interacción informal). En esencia, esta tendencia implica la separación de los procesos de desarrollo y sus colaboradores en diferentes localizaciones geográficas, lo cual conlleva a que los desarrolladores deban enfrentarse a dificultades en la interacción y coordinación de sus actividades [29; 33; 62]. Esto se debe a la poca o nula interacción cara a cara entre los desarrolladores, que es causada por la distancia entre los desarrolladores que participan en proyectos conjuntos [13; 17, 35, 39, 47]. Para hacer posible la interacción remota y síncrona en DSD, los trabajadores tienen que hacer uso de herramientas de comunicación tales como la mensajería instantánea y el teléfono. Sin embargo éstas han sido reportadas como fuentes constantes de interrupciones [12; 22], las cuales causan efectos negativos en la realización de tareas [36].

Este trabajo aborda de manera particular lo que llamamos *inicios de colaboración informados* en entornos de trabajo de DSD. Con esto se

propone que el iniciador de la interacción o colaboración cuente con información acerca de la actividad que está realizando el colaborador buscado. Con dicha información, el iniciador puede inferir si el momento para iniciar la colaboración es apropiado. Para ello se propone la conceptualización y caracterización tecnológica de esferas de trabajo colaborativas. Este concepto trata de ofrecer una visión parcial y personal de la información relativa a las unidades de trabajo compartidas con otros colaboradores, con lo cual se aportan ideas de diseño para el desarrollo de una herramienta prototipo. Esta herramienta fue llamada CWS-IM (Collaborative Working Spheres – Instant Messaging). Dicha herramienta prototipo es un mensajero instantáneo extendido con soporte para inicios de colaboración informados, la cual se introdujo a las actividades reales de DSD en una fábrica de software con la finalidad de evaluarla mediante un estudio de caso. Los resultados de esta evaluación proporcionan evidencia que muestra la aceptación favorable de CWS-IM por parte de los participantes en términos de utilidad, facilidad de uso, apoyo al inicio de interacción y el nivel de interrupción en sus actividades de DSD.

1.1. Planteamiento del problema

Con base a la problemática planteada previamente se requiere responder: ¿Qué es necesario para tener un inicio de colaboración informado en el desarrollo distribuido de software? Con lo anterior, será necesario responder ¿Cómo se puede proporcionar la perspectiva de actividad grupal a los miembros del equipo de trabajo de DSD? y finalmente es importante conocer ¿Cuál es el impacto de introducir apoyo para los inicios de colaboración informados en DSD?

1.2. Objetivos de la investigación

El objetivo general de este trabajo es modelar, caracterizar y evaluar el impacto de facilitar el inicio de colaboración entre personas que realizan desarrollo distribuido de software a través de la evaluación de esferas de trabajo colaborativas en su entorno de trabajo real.

Como objetivos particulares:

1. Caracterizar la forma de trabajo de un grupo de desarrolladores distribuidos de software.
2. Determinar los requerimientos de diseño que debe contener una herramienta de software para soportar inicios de colaboración informados a partir de la caracterización del grupo de estudio.
3. Crear un diseño que proporcione apoyo adecuado a los inicios de colaboración informados basado en los requerimientos.
4. Desarrollar una herramienta tecnológica prototipo que ofrezca soporte a los inicios de colaboración informados basados en el diseño propuesto.
5. Evaluar la percepción de utilidad, y de facilidad de uso para determinar la intención de adopción de la herramienta por un grupo de prueba en DSD real.

1.3. Metodología

El enfoque de esta metodología está basada en la creación de un prototipo que ayude a probar el concepto de esferas de trabajo colaborativas. Por lo que es importante que este prototipo sea evaluado por un grupo de trabajadores del DSD en su lugar de trabajo. La metodología de investigación utilizada para el desarrollo de este trabajo consta de diez (10) etapas:

1. Revisión bibliográfica. En esta etapa se realizó un análisis bibliográfico extenso para identificar los trabajos previos realizados y

fundamentar el marco teórico para el presente trabajo.

2. Estudio de usuarios clave. Esta etapa consistió en realizar un análisis de las actividades que realizan los trabajadores del DSD.
3. Crear escenarios iniciales. Se estudiaron distintos escenarios que representaban la forma en que se realizan las actividades del DSD.

Estas primeras 3 etapas permiten obtener un mejor entendimiento para caracterizar la forma de trabajo de los grupos en DSD (Objetivo particular 1) y empezar a establecer un conjunto de requerimientos para apoyar el inicio de interacción (Objetivo 2).

4. *Determinar ideas de diseño.* Tomando como base la revisión de literatura, el estudio de usuarios clave y los escenarios planteados, se determinaron las características y limitaciones de la situación actual de trabajo del DSD.
5. Diseñar herramienta (1ª. iteración). Esta primera iteración del diseño se llevó a cabo con base en las ideas de diseño de la etapa anterior.
6. Evaluación preliminar. Teniendo una primera versión de los diseños de las interfaces de usuario, se procedió a realizar una evaluación basada en escenarios para obtener la percepción de utilidad y facilidad de uso del funcionamiento de una herramienta con soporte para esferas de trabajo colaborativas por parte de los trabajadores de DSD.

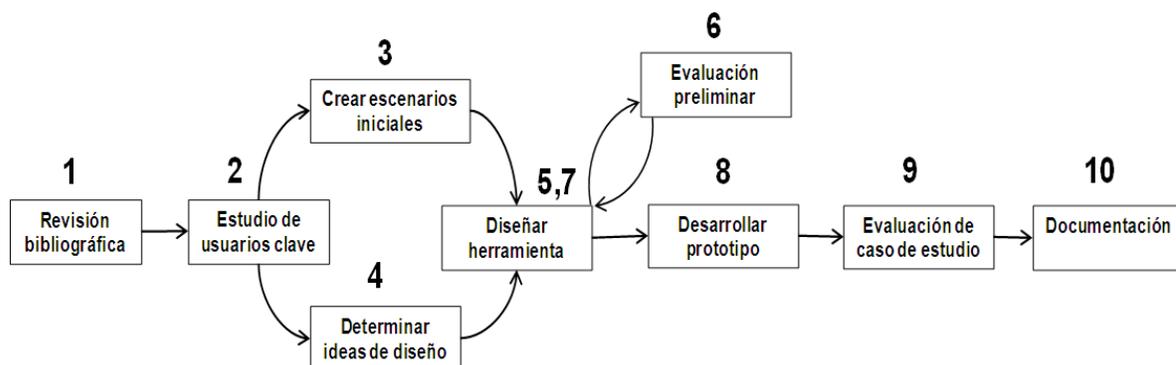


Fig. 1. Metodología seguida durante el proceso de investigación

7. Diseñar herramienta (2ª iteración). Con base en los resultados de la evaluación preliminar se informó el rediseño de la herramienta.

Estas etapas intermedias (4 – 7) permiten continuar con el establecimiento de los requerimientos (Objetivo 2) y establecer un diseño de una herramienta que ofrezca soporte a un inicio de colaboración informado (Objetivo 3).

8. Desarrollar prototipo. Esta etapa consistió en el desarrollo de la herramienta prototipo basado en el diseño refinado de la herramienta. Esta etapa permite alcanzar el objetivo 4.

9. Evaluación de estudio de caso. Dentro de esta etapa se realizó la planeación y ejecución del caso de estudio en una fábrica de software, donde se introdujo el prototipo desarrollado. Esta etapa permite alcanzar el objetivo 5.

10. Documentación. Esta etapa consistió en la documentación de los resultados y conclusiones del artículo.

1.4. Contenido

En la sección 2 se presenta el fundamento teórico donde se justifica la problemática abordada y se logra un entendimiento inicial de la misma en el entorno de trabajo. En la sección 3 son descritos los trabajos relacionados con la temática propuesta para este artículo. En la sección 4 se presenta y describe el concepto de esferas de trabajo colaborativas como parte de nuestra propuesta. En la sección 5 se describen las características de CWS-IM, el prototipo que considera los elementos de las esferas de trabajo como soporte para el inicio de interacciones informadas. En la sección 6 se presentan los resultados del estudio de caso donde se introdujo el prototipo. Finalmente en la sección 7 se presentan las conclusiones y recomendaciones derivadas de este artículo.

2. Marco teórico

En esta sección se presenta el entorno de trabajo del desarrollo de software distribuido y se establece la necesidad de contar con tecnologías

que permitan inicios de interacción remota de manera informada.

Para abordar esto, se requirió el entendimiento de los distintos conceptos y tópicos relacionados con las actividades de DSD. Para ello, se propuso y describió un modelo de trabajo, el cual fue creado a partir del análisis de literatura abordando tópicos de administración de proyectos, ingeniería de software, gestión de tareas y la teoría de articulación de trabajo.

Este modelo (ver Figura 2) aporta elementos importantes para el entendimiento, definición y relaciones que existen entre instancias que están involucradas en el trabajo del DSD. Este modelo es complementado con la perspectiva de la teoría de articulación de trabajo propuesta por Strauss (1985) [56], ya que esta teoría ayuda a explicar cómo se da el acoplamiento necesario para sustentar la instancia de Unidad de Trabajo. Esta teoría plantea tres niveles:

- Nivel 1. Se considera la visión amplia de la trayectoria o proyecto (vista como una planeación basada en las experiencias pasadas de trabajos ya realizados). Visualiza las principales características del arco de trabajo (totalidad de tareas ordenadas de manera secuencial y simultánea). Este nivel coincide con la parte del modelo de trabajo que denominamos especificaciones, debido a que la literatura confirma cómo los trabajos realizados bajo el enfoque de proyectos [21, 30, 38, 57] tienen su planeación inicial basada en la experiencia de los miembros de la organización y la asignación de tareas es realizada con base en el perfil o rol de los miembros involucrados (ver entidades en color azul de la Figura 2).
- Nivel 2. Se refiere al trabajo que las personas organizan, definen, supervisan y monitorizan con base en lo solicitado del primer nivel. Este nivel coincide con la parte de Ejecución del modelo de trabajo, en el sentido de que las personas a las que se les asigna un grupo de tareas, tienen el compromiso de realizarlas de la mejor manera buscando cumplir con las fechas límite previamente calendarizadas [2, 36, 57, 60]. En caso de contingencias, las personas improvisan para

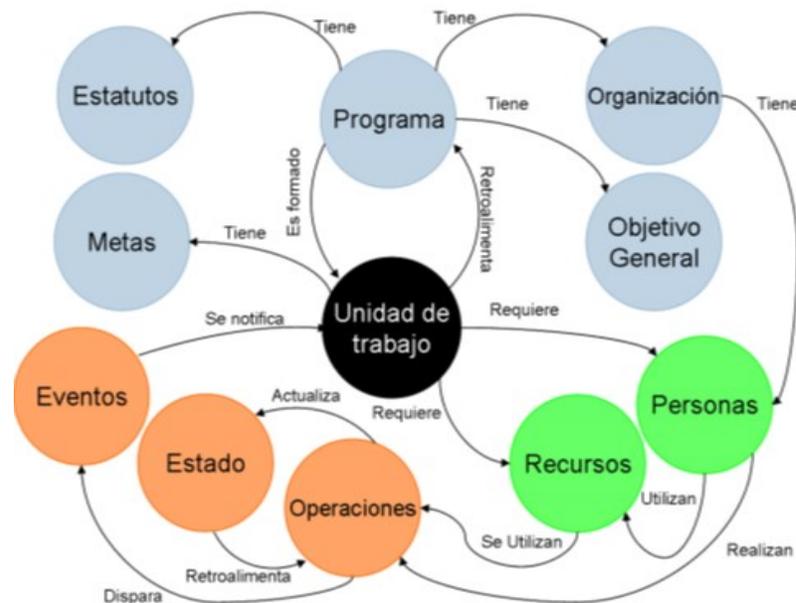


Fig. 2. Modelo de interrelación de instancias centrado en la unidad de trabajo para DSD

cumplir con sus tareas asignadas (ver entidades en color naranja de la Figura 2).

- Nivel 3. Se considera el agrupamiento y secuencia de tareas. Este nivel lo relacionamos con la parte de Esfera de trabajo [59], dado que las personas administran su trabajo de manera individual para lograr cumplir con el objetivo del programa (ver entidades en color verde de la Figura 2).

Con este modelo se obtuvo una visión teórica de los elementos y sus relaciones que fueron reportados como parte de la organización y división de las unidades de trabajo. Además con esto fue posible determinar los primeros requerimientos para una herramienta que soporte inicios de interacción informados.

4. Trabajos relacionados

Este apartado se divide en 2 grandes temas: manejo de interrupciones y desarrollo distribuido de software. Esto es en primera instancia para entender los elementos que pueden ayudar a

minimizar los costos negativos de las interrupciones y en segunda para comprender cómo se realizan los inicios de colaboración entre el grupo de trabajo, e identificar los aspectos clave para determinar los inicios de colaboración informados.

En cuanto al *manejo de interrupciones*, diversos investigadores han reportado cómo los trabajadores son interrumpidos en sus labores [25, 36, 43] y cómo la complejidad de la tarea (duración y tipo) [22, 36] y la cantidad de interrupciones impactan de manera negativa al tratar de reanudar la tarea interrumpida (p. ej. falla de memoria prospectiva) [15, 22, 25, 27].

Para que el inicio de colaboración pueda ser apropiado, desde la perspectiva de manejo de interrupciones, se deben tomar en cuenta dos aspectos importantes: el tipo de tarea que motiva la colaboración y la relevancia que tiene para los involucrados.

Con esto se puede inferir la voluntad para interactuar por parte del receptor [63]. Estos trabajos nos indican que los esfuerzos por proveer inicios de colaboración informados pueden estar en función de encontrar mecanismos o servicios que promuevan el

Tabla 1. Herramientas para coordinación de actividades en el desarrollo de software

Tipo de Actividad	Herramienta	Descripción
Localización de expertos	<i>Expertise Browser</i> [4]	Localiza expertos deseados mediante los datos del sistema de administración de cambios.
	<i>Step_IN</i> (socio-technical platform for in-situ networking) [63]	Localiza a colaboradores que cumplan con el nivel de conocimientos sobre la información de interés y que tengan cierta voluntad de ofrecer ayuda.
Codificación	<i>CodeSaw</i> [23]	Visualiza las contribuciones de códigos fuentes (líneas de código agregadas) al reunir el artefacto compartido (código) con la charla que rodea tal artefacto (por ejemplo, correos electrónicos).
	<i>CollabDev</i> [52]	Esta herramienta extrae datos de los códigos fuentes y cambia los datos del sistema de administración de versiones.
	<i>Tesseract</i> [5]	Es un ambiente interactivo donde los desarrolladores pueden investigar las relaciones del código fuente, los errores o fallas, el correo electrónico y las discusiones en torno al error.
Documentación	<i>XSDoc Wiki</i> [1]	Obtiene un conjunto de artefactos de software de múltiples fuentes (código fuente, los modelos, y los documentos).
	<i>Galaxy Wiki</i> [61]	Es un ambiente de desarrollo basado en Wiki y Java. El usuario puede iniciar o importar un proyecto java, crear o modificar una clase y ejecutar o depurar un programa.
Modelado	<i>D-UML</i> [41]	Es una implementación de un framework en JAVA el cual es consciente de la colaboración simultánea entre usuarios con la aplicación.
	<i>XCHIPS</i> [3]	La característica es que los usuarios pueden usarla en cualquier momento para acceder al modelo del proceso de manera concurrente.
	<i>SYSIPHUS</i> [10]	Permite la exploración, búsqueda, análisis y filtrado de modelos de sistemas y artefactos realizados por usuarios de la organización.
Pruebas	<i>ADAMS</i> [11]	Soporta la administración de calidad a través de la asociación de cada tipo de artefacto con una plantilla estándar y una lista de inspección de acuerdo al manual de la organización.
	<i>SoftFab</i> [26]	Habilita proyectos para automatizar la construcción y el proceso de pruebas. Esta infraestructura permite controlar los proyectos de manera remota.

suministro de información de la actividad del contacto potencial al iniciador (emisor).

Lo anterior con la finalidad de que el iniciador tenga elementos suficientes para identificar el momento apropiado para hacerlo (relevancia), y

de esta manera se pueda disminuir o equilibrar el costo o efecto negativo causado por la interrupción al receptor [36, 43].

Referente al *desarrollo distribuido de software*, se han realizado esfuerzos para tratar el

problema de la comunicación en los grupos de trabajos distribuidos en DSD, los cuales han sido abordados por los investigadores como comunicación inadecuada desde distintas perspectivas (p. ej. coordinación, diferencias culturales, diferencias horarias, etc.) [13; 19], ya que a partir de estas perspectivas se requiere que los desarrolladores realicen menos esfuerzos adicionales para lograr que la comunicación sea efectiva. Es por ello, que para entender de mejor manera la relación de estos trabajos con los inicios de colaboración informados, éstos fueron divididos con base en los requerimientos de: coordinación [35, 45], distribución de información de conciencia [18, 44] y entrada en colaboración [7].

En cuanto a *coordinación* existen muchas herramientas que han sido reportadas. Para este trabajo hemos recopilado las propuestas que se considera que aportan elementos para los inicios de colaboración en GSD. Para esto se han separado en cinco partes basados en las siguientes actividades: localización de expertos, codificación, documentación, modelado y pruebas. En la Tabla 1 se muestran las herramientas encontradas que apoyan a la coordinación de actividades en el desarrollo de software. Además en dicha tabla se describen y categorizan las herramientas de coordinación encontradas por el tipo de actividad en el desarrollo de software.

La conciencia según Dourish y Bellotti [44] es definida como la comprensión de las actividades de otros, lo que proporciona un contexto para la propia actividad. En este sentido se han propuesto un conjunto de características para soportar conciencia [18], las cuales demandan en general información acerca del estatus del proyecto y de la administración de cambios. ProjectWatcher [31] recopila información acerca de los artefactos del proyecto y las acciones del desarrollador sobre los artefactos en el ambiente de desarrollo Eclipse. StickySpots [32] es un sistema de mensajería basado en localización que mantiene informados a los miembros de un grupo mediante mensajes digitales cortos a varias ubicaciones del edificio. Community Bar [42] es un groupware diseñado para soportar interacciones casuales mediante el uso de video

(con webcam) y de pequeñas notas. FASTDash [28], utiliza una representación del código fuente compartido, el cual resalta las actividades actuales de los miembros del equipo. YooHoo [48] es un sistema para mantener informados a los desarrolladores con respecto a los cambios en los códigos fuentes. Esto es realizado mediante notificaciones.

Para este trabajo se considera que contar con soporte para *inicio informado de colaboración* debe ser visto desde la perspectiva que se reporta en [7], pues documenta el modelo base para entender de mejor manera los inicios de colaboración.

Se ha demostrado que dependiendo del tipo de interacción que se vaya a realizar, se debe tomar en cuenta el tipo de comunicación que debe ser ejecutada. Es decir se tienen que orientar las interacciones síncronas sobre los temas que no pueden ser resueltos durante una discusión asíncrona y que durante la interacción se requiere reducir incertidumbre de cierta actividad o tarea [17]. Por lo que también se ha afirmado que es importante mantener interacciones cortas de comunicación asíncrona, las cuales en ocasiones pueden servir como un sustituto suficiente para la comunicación síncrona [34].

Algunas herramientas para iniciar colaboración están diseñadas para propósitos distintos al de iniciar colaboración de manera informada en DSD, por lo que no necesariamente están pensadas para realizar una interacción alineada tanto al interés del emisor como del receptor. Sin embargo ofrecen una serie de elementos de información interesantes para el grupo de trabajo. Por ejemplo: Taskmaster [58] ofrece la lista de actividades asignadas e historial de actividad; Hubbub [24] incluye presencia e identidad; y ProjectView-IM [51] distribuye información de presencia, lista de actividades asignadas, identidad y notificaciones con base en recordatorios de tareas pendientes.

Después de haber analizado la literatura se han encontrado una serie de elementos que pueden ser útiles para un desarrollo que soporte inicio informado de colaboración, estos son:

- Los temas que motivan las interacciones deben ser convenientes tanto para el emisor

como para el receptor en el momento dado. Estos es por la importancia para definir cierto nivel de voluntad para interactuar [63].

- Según sea el tipo de información que se notifica debe ser el medio y la forma para iniciar la interacción [34].
- La herramienta para iniciar colaboración debe contar con acceso rápido y no invasivo a la información de la actividad de los posibles colaboradores [32, 58].
- La herramienta debe informar la actividad de manera transparente para el grupo de trabajo [31, 42, 51].
- Para poder distribuir información sobre la actividad del grupo, es necesario determinar la actividad individual de cada desarrollador [8, 42].

Sin embargo, también se encontró que las herramientas de comunicación a pesar de sus avances no ofrecen los siguientes aspectos:

- Información de la actividad actual del receptor.
- Estado de disponibilidad dependiendo de la actividad actual tanto del emisor como del receptor.

Como se puede notar, las herramientas descritas en este apartado están diseñadas para propósitos diferentes a los inicios de colaboración informados, por lo cual no permiten inferir la actividad de los colegas de trabajo a un nivel de acción u operación (en tiempo real) de las actividades del proyecto. Es por ello que con estos hallazgos, se adquieren ideas que podrán sustentar el diseño de un nuevo concepto particularmente para entornos de DSD. Por lo que, esta nueva forma de interactuar debe permitir a los colaboradores del proyecto la identificación de la actividad actual de los colegas, de tal manera que les permita saber de manera rápida y con mínimo esfuerzo quiénes de su grupo de contactos realizan actividades similares a la propia en un determinado momento. Así, hasta este punto se han abordado las preguntas de investigación establecidas: ¿Qué es necesario para determinar la oportunidad de colaboración? y ¿Qué es necesario para identificar y determinar el

momento adecuado para iniciar colaboración de manera informada?

6. Esferas de trabajo colaborativas

De acuerdo con [46], el desarrollo de software es una actividad intelectual compleja que consiste en generar una aplicación de computadora a partir de una necesidad planteada o identificada por un cliente, y que en principio, debe garantizar que la aplicación desarrollada satisfaga dicha necesidad. Sin embargo, cuando se cuestiona cuáles son las características de las actividades del desarrollo de software no existe una caracterización ampliamente aceptada (si es que hay una). Existen diversas propuestas que se centran en: i) las características del software mismo (producto), ii) las características de las actividades que se realizan como parte del desarrollo (proceso) o iii) las características de la organización.

6.1. Caracterización del DSD

Para este trabajo se consideran los siguientes elementos para caracterizar el desarrollo de software, inspirados por las propuestas de [37, 47], las cuales se describen a continuación:

- Escala: se refiere a los diversos valores que un desarrollo puede tomar en términos de:
 - Substrato social: el desarrollo puede ser realizado de manera individual, entre individuos de un grupo, en un grupo, entre grupos, dentro de una organización y entre organizaciones [40].
 - Distribución geográfica: los miembros del equipo pueden estar ubicados de manera co-localizada, localmente distribuida y remotamente distribuida o global [50].
 - Duración del esfuerzo de desarrollo: el tiempo para realizar un desarrollo puede estar expresado en días, semanas, meses, años. Cabe mencionar que la escala del desarrollo está relacionada con el tamaño del producto (pequeño, mediano, grande y muy grande).
- Incertidumbre: se refiere a la poca certeza que pueden tener los trabajadores respecto a

conocer el avance real de una tarea, meta, objetivo o incluso del proyecto mismo. Usualmente esto es generado por problemas de coordinación asociados a la escala del proyecto (cuanto mayor sea la escala, mayor tiende a ser la incertidumbre) y a la naturaleza cambiante del mundo (p. ej. los deseos del mismo usuario y las especificaciones del software cambian, el mundo externo para lo que el software fue diseñado cambia, las necesidades de negocios cambian, etc.). De esta manera, la incertidumbre depende tanto de la naturaleza del mundo real, como de las posibilidades técnicas que se tengan para enfrentar los problemas que de ésta se generan [6, 47].

- Interdependencia: esta característica se refiere a las dependencias que se presentan entre las diferentes actividades que realizan los desarrolladores. Según [37] estas dependencias pueden ser por: i) recursos compartidos, ii) asignación de tareas, iii) relación productor-consumidor, iv) restricciones con requisito previo, v) transferencia, vi) utilidad, vii) restricciones simultáneas, y viii) relación tareas/subtareas.
- Comunicación: se refiere a la forma en que la información se intercambia entre los miembros o participantes del proyecto con el fin de informar o comunicar avances, resultados, problemas, soluciones a problemas, justificaciones, etc. La comunicación puede ser tanto informal como formal [49]. La comunicación informal se refiere a las interacciones que se dan sin planearse la fecha o el lugar, donde el intercambio de información es breve pero interactivo y rico en contenido (p. ej. las interacciones en los pasillos). Por otro lado, la comunicación formal comprende las interacciones que se planean, y además tienen una agenda predefinida. Normalmente en este tipo de comunicación, un trabajador es el que se dirige a un grupo y son poco interactivas (p. ej. una reunión de avance de proyecto).

6.2. Gestión de actividades

Con esta caracterización de las actividades de los desarrolladores distribuidos de software se argumenta como los desarrolladores manejan sus actividades personales desde su espacio de trabajo y cómo esto fue conceptualizado como esferas de trabajo [60]. Tomando como fundamento lo anterior, se propone extender el concepto de esfera de trabajo a *esferas de trabajo colaborativas*. Algunas de las motivaciones para crear este concepto provienen de la integración del trabajo individual de los desarrolladores con el concepto de espacios de colaboración potencial [7], con lo cual se permita a los colaboradores obtener una vista parcial y personal de la información relacionada con las unidades de trabajo que son compartidas con otros trabajadores. En este caso los colaboradores y las unidades de trabajo despliegan información (p. ej. presencia y estado). Así pues, las esferas de trabajo colaborativas podrían permitir a los colaboradores descubrir, identificar o crear oportunidades de colaboración entre ellos, basados en la información manejada en sus unidades de trabajo individuales.

Para esto, la Esfera de Trabajo Colaborativa proveerá a los colaboradores con información suficiente y pertinente no sólo de la situación actual de trabajo (individual o grupal) sino también de las situaciones que suceden a su alrededor (periférica). Esto permite que se identifiquen, generen, establezcan o re-establezcan inicios de colaboración en momentos más adecuados conduciendo a minimizar el efecto negativo de las interrupciones [36].

Además, la Esfera de Trabajo Colaborativa permitirá a los colaboradores tener un punto de reunión con sus colaboradores potenciales, donde realmente se les ofrece iniciar una interacción adecuada y desde allí realizar la reunión de trabajo con los colaboradores y las unidades de trabajo implicadas, que consecuentemente dispararán el trabajo en grupo (colaboración actual).

Como primer paso para soportar la Esfera de Trabajo Colaborativa, se requiere que ésta sea caracterizada. Para plantear la caracterización

para Esferas de Trabajo Colaborativas se identificaron un conjunto de implicaciones de diseño que condujeron a las características que deben ser cubiertas por una herramienta para contar con este tipo de soporte.

6.3. Implicaciones de diseño

Basados en la caracterización de las actividades de desarrollo de software previamente descrita, la literatura de procesos de desarrollo de software [46], los conceptos de esfera de trabajo [60] y espacios de colaboración potencial [7], se identificaron un conjunto de implicaciones de diseño que deben ser cubiertos para iniciar colaboración de una manera más informada en DSD (ver Tabla 1). En la Tabla 1 se puede notar que las implicaciones están centradas en la necesidad de interactuar pero de una manera más informada. Es decir, que la interacción sea lo más oportuna tanto para el que inicia la interacción (emisor) como para el que está siendo buscado (receptor).

En lo referente a la característica de *Escala* se desea que la tecnología posea (I1) servicios de comunicación que permitan la interacción entre personas que se encuentran distribuidas o colocalizadas. Con esto se pretende agregar a todas las personas involucradas en el desarrollo de la actividad. Este servicio permitirá a los miembros del grupo de trabajo interactuar de manera remota, aun cuando el número de involucrados en el proyecto se incremente y estén ubicados en diferentes ubicaciones geográficas.

En cuanto a la *incertidumbre*, se requiere (I2) un mecanismo para compartir información de los proyectos entre los colegas o personas que estén relacionados, ya que de esa forma se conocerá el grado de avance de las personas en sus actividades y del proyecto mismo. Este mecanismo apoyará al usuario a tener conocimiento de lo que sucede en torno a alguna o algunas actividades del proyecto en que se tenga particular interés.

En el caso de *Interdependencia*, se necesita en primer lugar, (I3) un mecanismo que permita conocer el grado de avance en las actividades de cada uno de los miembros del proyecto, y en

segundo lugar, (I4) un mecanismo que permita localizar e interactuar con los miembros relacionados con las unidades de trabajo. Estos mecanismos permitirán informar lo que sucede en torno al receptor potencial, cuando existe el interés de articular una unidad de trabajo común.

Finalmente, referente a la *Comunicación*, se demanda en primer lugar, (I5) un mecanismo que permita conocer el estado de los miembros del proyecto, así como la tarea que están realizando para tener una idea más acertada de la conveniencia de llevar a cabo la interrupción. En segundo lugar, (I6) un mecanismo que identifique cuando un usuario puede interactuar con otro, basado en la necesidad de interacción, estado y unidad de trabajo que están realizando tanto del trabajador emisor como del receptor. Y en tercer lugar, (I7) se necesita un servicio de comunicación síncrona y asíncrona, dado que será conveniente tener una interacción continua (síncrona), pero también habrá situaciones donde se requiera que las notificaciones no sean intrusivas y se vayan acumulando para posteriormente ser revisadas (asíncrona). Estos mecanismos y el servicio implican una tecnología de comunicación que posea información de la actividad actual de los receptores potenciales.

Con en este apartado se aporta un conjunto de implicaciones de diseño para el logro de inicios de colaboración informados en DSD. Para obtener estas implicaciones fue necesario el análisis de literatura previo, lo cual permitió tener una perspectiva diferente de cómo iniciar una colaboración mediante el uso de una herramienta. Estas implicaciones aportan claramente nuevas ideas de diseño para la creación de nuevos mecanismos y servicios para iniciar colaboración de manera informada en DSD, los cuales pueden ser utilizados por desarrolladores que requieran crear herramientas de comunicación con este tipo de característica.

Con las implicaciones de diseño identificadas, se procede a describir el concepto de esferas de trabajo colaborativas en el siguiente apartado mediante un modelo conceptual, con el cual se ilustran los diferentes conceptos que están implicados en los inicios de colaboración informados.

6.4. Modelo conceptual

En este apartado se describe un modelo propuesto para esferas de trabajo colaborativas, con el cual se pretende entender como los participantes de una interacción a través de la realización de sus unidades de trabajo pueden lograr inicios de interacción apropiados.

La Figura 3 muestra a dos desarrolladores que trabajan en alguna actividad dentro de un proyecto, la cual está representada por un conjunto de elementos que desencadenan la interacción con una actividad. Una esfera de trabajo está representada por los eventos, personas, tareas, recursos y acciones que definen la forma en la cual la gente realiza una actividad en particular.

La Figura 3 también incluye el concepto de colaboración potencial (representado por una flecha). Esto permite a los usuarios o desarrolladores identificar oportunidades para iniciar colaboración a través de la monitorización del contexto de trabajo. Para lograr esto, se requieren llevar a cabo tres tareas principales:

1. Identificar la información necesaria de la actividad de los involucrados;
2. Identificar un momento adecuado para interrumpir a otros colaboradores; y
3. Iniciar colaboración, si es el momento adecuado

Estos tareas o pasos representan el proceso de colaboración potencial [7, 8]. Dicho de otra forma, durante la realización de actividades, el desarrollador 1 (emisor) requiere contactar al desarrollador 2 (receptor) para resolver algunas dudas respecto a una actividad común. De esta manera, la información referente a la actividad actual del receptor podría convertirse en conocimiento compartido para el grupo de trabajo.

Con esta información, el emisor crea conocimiento a través de su proceso cognitivo, que incluye:

- *Adquisición de información*, se obtiene mediante la consulta de la actividad actual del receptor potencial (p. ej.

obtener información sobre la actividad del proyecto que el receptor potencial está llevando a cabo, el rol que está desempeñando, etc.).

- *Integración de conocimiento*, cuando la información es obtenida de los receptores potenciales, se integra con los conocimientos adquiridos de otras fuentes de conocimiento (p. ej. el calendario de las actividades pendientes, la prioridad, documentos de diseño, información de proyectos, etc.).
- *Toma de decisión*, con la integración de las diferentes fuentes de conocimiento, el emisor sabe con más detalle en que actividad del proyecto está trabajando el receptor potencial. A través de esta integración podría ser posible determinar si el momento actual es adecuado para iniciar la colaboración.

Como se puede notar, esta caracterización de las esferas de trabajo colaborativas se fundamentó en la caracterización de las actividades de DSD propuesta y el análisis de literatura de gestión personal de actividades y del potencial de colaboración. Por lo que con las implicaciones de diseño identificadas en el apartado anterior y el modelo conceptual previamente descrito, es posible proponer una definición para las esferas de trabajo colaborativas, la cual es una aportación importante de este artículo debido a que este concepto no se encontró en un modelo que ayudará a la interpretación de los inicios de colaboración informados.

Además esta caracterización establece una forma nueva para iniciar colaboración en grupos de trabajo virtual desde la actividad individual y el descubrimiento o creación de inicios de colaboración oportunos. A diferencia de las propuestas para distribución de información de conciencia (p. ej. ProjectWatcher), las esferas de trabajo colaborativas distribuyen la información basado en la actividad que realiza cada individuo desde su espacio de trabajo, y no desde información que es capturada en sistemas de la organización (p. ej. sistemas de administración de cambios, sistema de control de versiones, etc.).

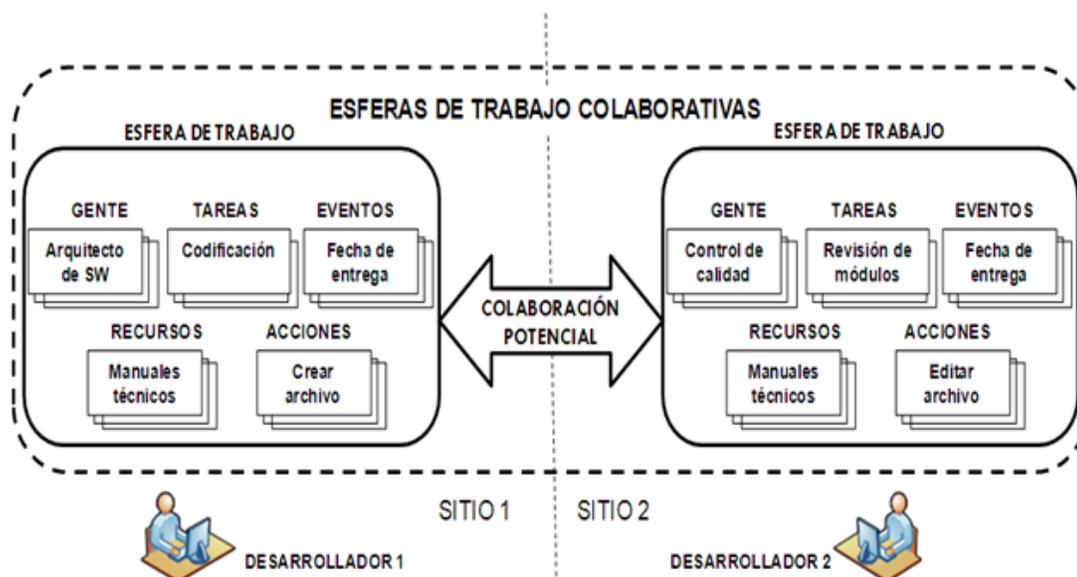


Fig. 3. Modelo conceptual de esferas de trabajo colaborativas

6.5. Definición

Tomando en cuenta el análisis realizado de las necesidades de interacción, la caracterización, el modelado previamente descrito, y la integración de las definiciones de esferas de trabajo y colaboración potencial, se propone la siguiente definición para esferas de trabajo colaborativas:

Es un conjunto de unidades de trabajo, recursos y personas que comparten una meta u objetivo común en un lapso de tiempo determinado, que proporcionan conocimiento de la actividad de los miembros del grupo de trabajo para el inicio de colaboración informado.

En cuanto a la definición, ésta se divide en 3 partes:

- La primera parte integra los elementos que componen el concepto de esfera de trabajo.
- La segunda parte de la definición considera el concepto de colaboración potencial, en el cual se argumenta que se proporciona el conocimiento de la actividad individual al grupo de trabajo.

- Finalmente, la tercera parte considera el inicio de colaboración informado, como resultado de la identificación o descubrimiento de oportunidades de colaboración [7, 8].

De la definición de este concepto se deriva una nueva forma para desplegar estatus de disponibilidad al grupo de trabajo. Esto se ha denominado *disponibilidad selectiva*, donde un usuario está disponible para interactuar con otros colaboradores cuya actividad está relacionada a la unidad de trabajo que él o ella está actualmente realizando, y por lo que está menos disponible (o no disponible) para otros colaboradores.

A continuación se presentan las siguientes propuestas para identificar el conjunto de estatus de usuario que pueden ayudar al diseño de interfaces de usuario que soporten el concepto de esferas de trabajo colaborativas. Tales estatus son:

- *Disponible* para aquellos colegas que están trabajando en la misma actividad y el mismo proyecto.

Soporte al inicio de colaboración en el desarrollo distribuido de software mediante esferas de trabajo colaborativas. 799

- *Alcanzable pero ocupado* para los colegas que trabajan en diferente actividad del mismo proyecto.
- *Ocupado* para quienes realizan actividades de distintos proyectos.

7. CWS-IM: un mensajero instantáneo con soporte para esferas de trabajo colaborativas

La aplicación propuesta para ilustrar el concepto de esferas de trabajo colaborativas se inspiró en la mensajería instantánea tradicional. Dicha herramienta fue llamada CWS-IM. La razón para diseñar este tipo de herramienta radica en que la mensajería instantánea tradicional es una herramienta de comunicación síncrona utilizada frecuentemente por los trabajadores del desarrollo distribuido de software. Sin embargo estas herramientas comúnmente no ofrecen una integración con las herramientas utilizadas para la administración de proyectos de software y tampoco es posible determinar estatus de usuario basados en la actividad actual. Por lo que estos aspectos fueron integrados en CWS-IM para proporcionar inicios de colaboración mejor informados.

Mediante el uso de CWS-IM los desarrolladores distribuidos de software son apoyados con información de las esferas de trabajo colaborativas, pues un desarrollador sabe QUIÉN está potencialmente disponible para colaborar y de manera opcional pudiera saber DÓNDE se encuentran, QUÉ están haciendo con respecto a las unidades de trabajo de la organización, PORQUÉ lo están haciendo, se puede inferir CUÁNDO es el momento más apropiado para contactarlos y CÓMO hay que integrar los esfuerzos realizados para las unidades de trabajo.

Para interpretar la presencia y función de los contactos potenciales desde el listado de contactos de CWS-IM (ver Figura 4), es necesario entender el código de colores y los íconos que se proponen para la herramienta. En cuanto a los íconos estos son representador por figuras alusivas al rol (p. ej., líder de proyecto = lentes, programador = teclado, etc.). En cuanto al

código de colores fue propuesto a partir de la nueva forma para desplegar estatus de disponibilidad denominado *Disponibilidad Selectiva*. Para esto las situaciones de trabajo identificadas por la interfaz de CWS-IM y mostradas a través del código de colores, deben ser interpretadas básicamente de la siguiente forma: 1) *Código verde*: Disponible para aquellos colegas que están trabajando en la misma actividad y el mismo proyecto. 2) *Código amarillo*: Alcanzable pero ocupado para los colegas que trabajan en diferente actividad del mismo proyecto. 3) *Código rojo*: Ocupado para quienes realizan actividades de distintos proyectos.

De esta forma un colaborador puede desplegar diferentes estatus a sus colegas dependiendo de la actividad que realicen, es decir para un colaborador A puede estar en VERDE y al mismo tiempo para un colaborador B en ROJO y para otros colaboradores puede estar en AMARILLO.

7.1. Componentes GUI

Para ilustrar la función de cada componente GUI a continuación se presenta una descripción (ver Figura 4):

Lista de contactos (4A): ésta es utilizada para mostrar los contactos que están relacionados con la actividad que se está realizando, con lo cual se aborda la implicación I6, que trata sobre un mecanismo que identifique cuando un usuario puede interactuar con otro, basado en un perfil de necesidades, estado y/o tarea que está realizando. Además en este listado se concentran los colaboradores potenciales de dicha actividad. Basta con dar doble click al nombre del colaborador deseado para iniciar una interacción por medio de chat. De esta manera se proporcionan servicios que permiten la comunicación entre todos los miembros del proyecto (I1) y el servicio de comunicación (I7). Cabe mencionar que sólo si el contacto está señalado como disponible se envía la petición. Si está en estatus ocupado se manda una advertencia de que es posible no ser atendido pero se envía la petición. Para que se inicie el chat, la contraparte debe aceptar la participación y es hasta entonces cuando se hace el intercambio de mensajes. También es posible

realizar un chat en grupo y compartir la conversación con todo el equipo de trabajo a pesar de que no estén conectados en ese momento.

Panel de Identificación (4B): es utilizado para presentar la información actual del usuario (colaborador), es decir muestra su nombre y el rol que desempeña en la actividad actual. Esta información es obtenida de los datos registrados en el repositorio de proyectos. Con esto se proporciona información concerniente a cuando un usuario puede interactuar con otro, basado en un perfil de necesidades, estado y/o tarea que está realizando (I6). La información que proporciona este componente permite compartir y filtrar datos relacionados al proyecto entre colegas (I2), ya que es a partir de esta información con la que se establecen los estatus de usuario (I5).

Vista de contactos (4C): cambia la vista del listado de contactos en dos niveles: proyecto y actividades. Este componente permite saber quiénes son los contactos que participan a nivel proyecto (ver Figura 4A) o más específicamente a nivel actividad (ver Figura 4F). Con esto se abordan las implicaciones I1 e I6 puesto que las diferentes perspectivas del listado de contactos permiten comunicarse con el colega e identificar cuando un usuario está en un momento oportuno para interactuar. Por defecto, el sistema filtra la vista por proyecto, pero eventualmente el usuario puede cambiar la vista mediante este componente.

ToolTip de *información del contacto* (4D): presionando el botón derecho del ratón sobre un contacto se proporciona información del colaborador potencial en el momento actual (p. ej. ubicación, documento que tiene activo, rol que está desempeñando, avance, etc.). Con este componente se proporciona un mecanismo que permite conocer qué actividad realiza el grupo de colegas (I5) y a su vez es posible conocer el grado de avance de manera individual (I3). Este componente recaba la información del repositorio de proyectos de la organización.

Etiqueta de ubicación (4E): este componente proporciona información de la ubicación geográfica (I4), con cual es posible identificar la zona horaria y ciudad donde está ubicado un

colega. Esta información es dada a partir de la dirección IP de la computadora que el usuario está utilizando. Esta información es vista por el resto de los colaboradores si es requerido. Asimismo el usuario puede cambiar el texto de la ubicación si lo cree conveniente.

7.2. Comparación con otras herramientas

Se han propuesto proyectos similares que se centran en el suministro de información sobre las actividades que se producen en un grupo de desarrollo. Por ejemplo: Palantir [9, 54] es un sistema que complementa los sistemas de configuración existentes al proporcionar conciencia distribuida de los avances del proyecto. Esto se hace a través de una pantalla gráfica que muestra las mediciones de gravedad y el impacto de los cambios en los artefactos, por lo que los desarrolladores pueden anticipar los problemas que puedan derivarse de dichos cambios. Este proyecto, sin embargo se centra sólo en la gestión de los cambios en los artefactos. ProjectWatcher [31] es un sistema diseñado para apoyar la conciencia con respecto a "quién es quién en general" y "qué trabaja en esta área del código". También existen mecanismos (por ejemplo Eclipse Plugins [53] para informar a los usuarios sobre la ubicación de un cambio concurrente a nivel de clase, el método y la línea. La mensajería instantánea tradicional como Skype [55], Microsoft Communicator [16], entre otros han ido evolucionando y han incluido servicios de comunicación mediante videoconferencia, llamadas y mensajes de texto, sin embargo no proporcionan presencia de los usuarios e identidad basados en su actividad de trabajo actual. Estos dos proyectos permiten mantener al equipo informado sobre el estado de las secciones de código específicas y archivos. Además el listado de contactos sólo es categorizado por favoritos, conectados u orden alfabético.

Por el contrario, nuestra propuesta ofrece una nueva manera de reflejar de forma automática el cambio de estado de un usuario en función de las actividades de proyectos asignados, así como el filtrado de estado de acuerdo con la actividad

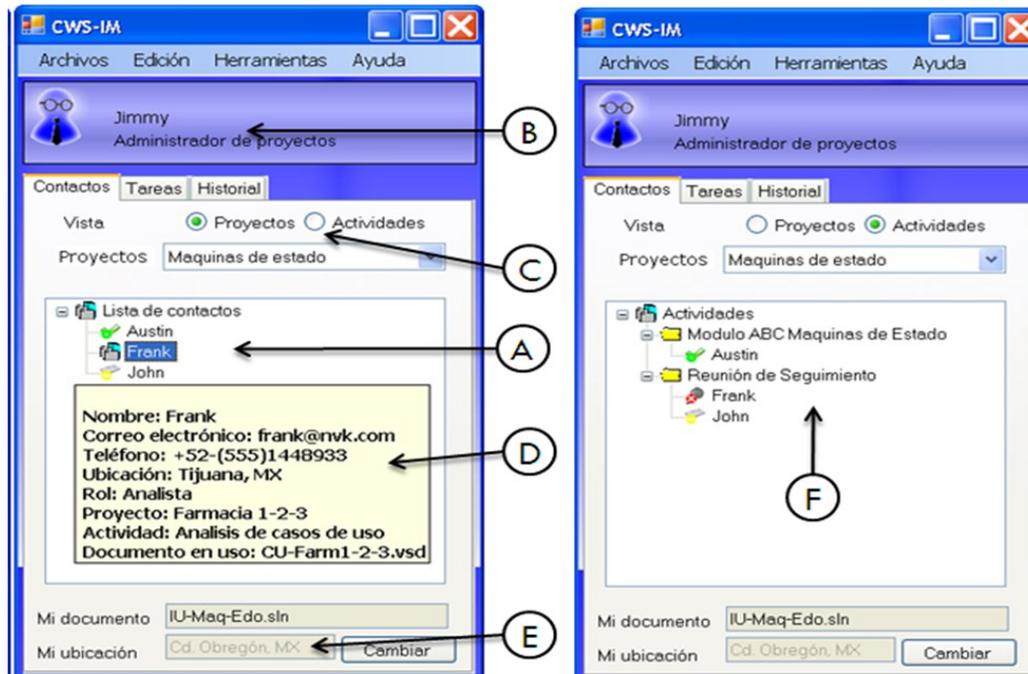


Fig. 4. Componentes GUI de CWS-IM: Vista de contactos

tanto del emisor como del receptor (Disponibilidad Selectiva). Además, ninguno de los trabajos citados se centra en informar al grupo con el nivel de dinamismo y granularidad con respecto a la actividad que realizan los individuos de su propio espacio de trabajo, los artefactos utilizados en esta actividad, y la información relacionada con el proyecto asociado, y los miembros del equipo involucrados.

Por último, es importante destacar que el enfoque que concierne al conjunto de elementos de información y los mecanismos utilizados para obtenerlos y presentarlos no pretende ser una propuesta concluyente ni completa, sino una propuesta de partida para explorar nuevas medidas y mecanismos que faciliten el inicio de la colaboración a partir de una interacción que está mejor informada y adecuada a las necesidades tanto del emisor como del receptor.

Como se puede notar, la herramienta CWS-IM es un prototipo basado en la funcionalidad de la mensajería instantánea con soporte para Esferas de Trabajo Colaborativas (CWS-IM). Uno de los aspectos importantes del objetivo de este trabajo,

es introducir un concepto nuevo al ambiente de trabajo del DSD. Por lo que este prototipo hace posible introducir los inicios de interacción informados a un grupo de estudio.

8. Evaluación

La evaluación de CWS-IM fue realizada a través de un estudio de caso, el cual consistió en introducir la herramienta en una fábrica de software.

Con la evaluación se pretende demostrar que los inicios de colaboración informados pueden auxiliar a los trabajadores a interpretar la actividad actual de los colaboradores con los que se comparte una unidad de trabajo y así determinar los momentos más adecuados para iniciar la colaboración síncrona. También la intención de ésta es comparar el prototipo CWS-IM con la mensajería instantánea tradicional (IM) en cuanto a utilidad y facilidad de uso durante la actividad laboral.

La evaluación se orientó a tres aspectos: i) Utilidad y facilidad de uso, ii) Apoyo a Inicio de colaboración, y iii) Nivel de interrupción de la herramienta. Estos aspectos permiten en primer lugar conocer el impacto en cuanto a si el inicio informado de colaboración (mediante el CWS-IM) pudiera ser adoptado por los trabajadores del DSD. En segundo lugar permite conocer si se proporciona información suficiente para determinar la actividad actual del colega. Finalmente en tercer lugar permite comparar una herramienta basada en el modelo de esferas de trabajo colaborativas con una herramienta de mensajería tradicional en cuanto a su nivel de interrupción con los colegas.

El paradigma de diseño para este estudio fue *Within Subjects*, con la participación de 16 trabajadores del DSD (4 administradores de proyecto, 2 ingenieros de control o aseguramiento de calidad, 4 programadores y 6 arquitectos de software) en una fábrica de software distribuida. Los trabajadores estuvieron distribuidos de la siguiente manera: 5 en Guaymas, Sonora, México, 8 en Ciudad Obregón, Sonora, México y 3 en Tucson, Arizona, USA.

Para llevar a cabo el estudio se realizaron 7 pasos: 1) reunión inicial, para informar el propósito del estudio; 2) entrevista personal, para conocer como los participantes organizan sus actividades de trabajo; 3) Aplicación de cuestionario de entrada, para obtener los datos del perfil de los participantes (edad, sexo, experiencia, rol, etc.); 4) instalación de sistemas; 5) entrenamiento personalizado de los sistemas a los participantes; 6) uso del sistema, una semana se usó CWS-IM y otra semana un mensajero tradicional; y 7) evaluación de usabilidad y facilidad de uso mediante la aplicación de cuestionarios TAM [20, 14] por cada sistema utilizado.

En cuanto a *utilidad y facilidad de uso* se estableció una hipótesis para determinar si existe diferencia en la utilización de las herramientas CWS-IM y la IM tradicional. Los resultados arrojados por SUS (Con rango de 0 a 100) indican que CWS-IM (media=82.5, desv. est.= 2.02892) es más útil que el IM (media=70.94, desv. est.= 3.05232). Para confirmar esto se realizó la prueba t de student a un nivel de

significancia $p < 0.01$. Dado que $t_{0.01} = -2.147$ se comprueba que esta diferencia es significativa. Asimismo, en forma paralela se aplicó el cuestionario TAM para verificar con 6 de sus reactivos el nivel de usabilidad de IM y CWS-IM y confirmó los resultados del cuestionario SUS. Por lo que los resultados arrojados tanto por SUS y TAM indican que la herramienta CWS-IM cuenta con elementos de información que les parecen útiles para el trabajo de DSD en comparación con la mensajería instantánea tradicional.

Con respecto a iniciar colaboración, se establecieron hipótesis en cuanto a si existe diferencia en la utilización de la información proporcionada por CWS-IM respecto a la proporcionada por el IM.

Para esto se evaluó la función de despliegue de información de usuarios con base en la actividad actual, dado que CWS-IM utiliza un código de colores para indicar la actividad actual. Los resultados del análisis indican que los usuarios prefirieron solicitar información adicional de los receptores cuando estaban en un estatus de "diferente actividad / mismo proyecto" o "diferente actividad / diferente proyecto" (media = 1.71, desv. est. = 0.442) que cuando estaban en el estatus "misma actividad / mismo proyecto" (media= 1.51, desv. est. = 0.502). Para confirmar esta información se realizó la prueba t de Student a nivel de significancia $p < 0.01$. Puesto que $t_{0.01} = 4.268$ se puede establecer que esta diferencia es significativa, por lo que hay evidencia suficiente para argumentar que los usuarios entendieron la información proporcionada por CWS-IM.

En cuanto a percepción de la interrupción con el uso de los sistemas IM y CWS-IM se establecieron hipótesis con respecto a si existe diferencia en la percepción de interrupción de los sujetos en cuanto al uso del sistema. Para la evaluación de esta hipótesis se utilizaron los datos de un mini-cuestionario que se aplicó aleatoriamente tanto cuando la interrupción se generaba por el sistema IM como cuando se generaba con CWS-IM. En este caso tomamos el valor arrojado para el nivel percibido de perjuicio de la interacción para hacer una comparación de medias con la prueba t de Student. Las diferencias de las medias de percepción entre los sistemas IM (media =2.04, desv. est.= 1.115) y

para CWS-IM (media =1.32, desv. est.=0.47), resultaron con una diferencia estadísticamente significativa ($t_{.01}=5.888$). Por lo que podemos afirmar que las interacciones realizadas con IM se percibieron más perjudiciales que las reportadas con CWS-IM.

Los resultados obtenidos fueron analizados y se reflexionó sobre posibles razones para explicarlos. A continuación se presentan "posibles explicaciones" para cada resultado.

En cuanto a *utilidad y facilidad de uso*, se considera que los participantes percibieron a la herramienta CWS-IM más útil que a la de IM, porque la primera herramienta permite a los trabajadores conocer de manera rápida y con poco esfuerzo qué actividad realiza una persona, el rol que está desempeñando en un momento determinado y el nombre del documento que se está modificando. Además, CWS-IM proporciona elementos para organizar las actividades asignadas de manera automática, tomando la información desde el sistema gestor de proyectos. También, con CWS-IM las actividades que son asignadas en el gestor de proyectos son reconocidas en el momento que son capturadas. Esto aunado a que CWS-IM es un medio de comunicación que permite interactuar de manera remota con los contactos del proyecto. Esto es realizado a través de grupos de usuarios que se presentan de acuerdo al proyecto y/o actividad en la que están trabajando. Por lo que claramente la utilidad de CWS-IM fue apreciada en buena manera por los participantes. En cambio, en cuanto a facilidad de uso resultó que tanto el sistema IM como CWS-IM fueron percibidos como fáciles de usar. Puesto que CWS-IM es considerada más útil que IM y además CWS-IM fue percibida como fácil de usar al igual que IM, se tienen elementos suficientes para afirmar que CWS-IM es una herramienta que sus usuarios potenciales perciben como que puede ser adoptada de manera favorable en DSD.

En lo que respecta a *apoyo al inicio de interacción*, los resultados indican que la herramienta ofrece información suficiente al iniciador de la interacción, la cual le permite determinar la actividad actual de un receptor o contacto potencial, e influenciar su decisión de contactarlo sólo cuando sea considerado

apropiado (p. ej. cuando están trabajando en actividades similares). En cambio, al receptor le podría ayudar a reducir problemas relacionados con los costos ocultos causados por la interrupción, tales como las fallas de memoria prospectiva [27] y la sobrecarga mental [12].

En cuanto a *nivel de interrupción de la herramienta*, se considera que aun cuando las personas son interrumpidas cuando están trabajando en una actividad, en la cual se tiene un interés compartido, no significa que dicha interrupción no vaya a causar molestia o perturbar la actividad. Esto ha sido reportado por otros trabajos de investigación como [63], quien identificó que los trabajadores pueden tener más voluntad para atender a un colega cuando se identifican intereses comunes entre ambas partes (emisor y receptor). Sin embargo, factores como la carga de trabajo, la prioridad o nivel de urgencia de la actividad que se realiza y eventos que suceden en torno al proyecto (p. ej. hitos, fechas de entrega, uso de recursos compartidos, etc.) son factores que pueden determinar el nivel del efecto de interrupción.

Finalmente en lo referente a *interrupción según la herramienta utilizada* en el estudio, los resultados muestran que CWS-IM fue considerada menos perjudicial que IM pues ofrece información de la actividad actual de los receptores, y esta información permite al emisor interpretar la situación de trabajo del receptor. De esta forma el emisor tiene más elementos para poder decidir si el momento es adecuado para iniciar colaboración, lo cual fue soportado por los resultados de utilidad, facilidad de uso, adopción y entendimiento de la actividad actual.

Esta evaluación nos ofrece resultados diferentes a los que se encontraron en los trabajos de manejo de interrupciones, ya que el prototipo permitió medir la percepción del impacto de la interrupción, la utilidad y facilidad de uso de la herramienta con un grupo de trabajadores del DSD. Adicionalmente, este caso de estudio permitió evaluar el prototipo con trabajadores reales realizando distintos roles de trabajo durante su jornada laboral del DSD. Esto fue importante pues en los estudios analizados de DSD no fue encontrado uno similar, la mayoría de ellos presentaban casos de estudio controlados.

Por lo que este estudio de caso puede ser considerado como un conjunto inicial de resultados que pueden servir como una base a partir de la cual otros investigadores podrían comenzar a explorar más a fondo este tipo de temas.

9. Conclusión

Este trabajo se inició con un proceso de entendimiento inicial de las características de las actividades de los desarrolladores de software basado en la literatura. Con esto se procedió a establecer ideas de diseño que pudieran contribuir a las necesidades de interacción de los trabajadores de DSD durante la realización de sus actividades. De esta manera, se definió que se contaba con elementos suficientes para tomar en cuenta la propuesta de contar con soporte para interacciones informales, las cuales promovieran inicios de colaboración en momentos apropiados entre los miembros de un grupo de trabajo virtual. Considerando estas ideas de diseño, se introdujo el concepto de Esferas de Trabajo Colaborativas como un medio que proporciona al emisor (iniciador de una interacción) información detallada referente a la situación o estatus de trabajo de un receptor potencial, de tal forma que dicha información ayude al emisor a tomar una mejor decisión respecto a cuándo y cómo abordar a un colega en un momento determinado. En este momento se crea la propuesta que se denomina *Disponibilidad Selectiva*, la cual surge como parte del concepto de esferas de trabajo colaborativas a manera de idea para lograr los inicios de colaboración informados.

De lo anterior se derivó el desarrollo de un prototipo que denominamos CWS-IM, el cual es un mensajero instantáneo extendido con soporte para disponibilidad selectiva a través de esferas de trabajo colaborativas. El prototipo fue desarrollado para soportar la recolección, comparación e interpretación de información de la actividad actual de los usuarios, con lo cual las esferas de trabajo individuales se alinean para formar esferas de trabajo colaborativas entre los miembros del equipo de trabajo.

Con lo anterior y con la finalidad de presentar el uso y los beneficios del prototipo CWS-IM, se llevó a cabo un estudio de caso en un ambiente de DSD real en una fábrica de software con sedes en Sonora, México y Arizona, E.U. Éste consistió de un estudio de 3 semanas de trabajo, en el cual se obtuvo la percepción de 16 desarrolladores DSD en términos de utilidad y facilidad de uso a través de los cuestionarios SUS y TAM. Estos resultados indicaron que el prototipo fue bien recibido por los participantes, pues lo consideran más útil y adecuado que la mensajería instantánea tradicional para las actividades de DSD. Cabe mencionar que la evidencia muestra que los participantes entendieron la información proporcionada por CWS-IM respecto a la actividad de los colegas. El estudio también confirmó que aunque las personas estén trabajando en una misma actividad o una similar pueden estar más dispuestas para aceptar una interacción, pero no necesariamente significa que ésta será percibida de manera positiva. También, la evidencia muestra que los participantes (actuando como receptores de la interacción) percibieron las interrupciones como menos negativas cuando se utilizó CWS-IM que cuando se utilizó el mensajero instantáneo tradicional (MI).

Las principales contribuciones de esta investigación se centraron en 1) la conceptualización y caracterización del concepto de *esferas de trabajo colaborativas* para el apoyo de inicios de colaboración de manera informada, 2) la propuesta y diseño del concepto *disponibilidad selectiva*, y 3) el desarrollo y evaluación de una herramienta aporta evidencia respecto a la factibilidad de instrumentar este tipo de soporte.

Referencias

1. **Aguiar, A. & David, G. (2005).** WikiWiki weaving heterogeneous software artifacts. *Proc. of the 2005 international symposium on Wikis*, San Diego, California.
2. **Brush, A.B., Meyers, B.R., Tan, D.S., & Czerwinski, M. (2007).** Understanding memory triggers for task tracking. *Proc. of the SIGCHI*

Conference on Human Factors in Computing Systems, San Jose, California.

3. **Fernández, A., Garzaldeen, B., Grützner, I., & Münch, J. (2004).** Guided support for collaborative modeling, enactment and simulation of software development processes. *Software Process: Improvement and Practice*, Vol. 9, No. 2, pp. 95–106.
4. **Mockus, A. & Herbsleb, J.D. (2002).** Expertise browser: a quantitative approach to identifying expertise. *Proc. of the 24th International Conference on Software Engineering*, Orlando, Florida.
5. **Sarma, A., Maccherone, L., Wagstrom, P., & Herbsleb, J. (2009).** Tesseract: Interactive visual exploration of socio-technical relationships in software development. *Proc. of the 2009 IEEE 31st International Conference on Software Engineering*.
6. **Stinchcombe, A. & Heimer, C. (1986).** *Organization Theory and Project Management: Administering Uncertainty in Norwegian Offshore Oil*. A Scandinavian University Press Publication.
7. **Morán, A.L., Favela, J., Martínez, A., & Decouchant, D. (2004).** On the Design of Potential Collaboration Spaces. *The International Journal of Computer Applications and Technology (IJCAT)*, Vol. 19, No. 3/4, pp. 261–271.
8. **Morán, A.L., Favela, J., Romero, R., Natsu, H., Perez, C., Robles, O., & Martínez, A. (2008).** Potential and actual collaboration support for distributed Pair-Programming. *Computación y sistemas*, Vol. 11, No. 3, pp. 211–229.
9. **Al-Ani, B. & Redmiles, D. (2009).** Trust in Distributed Teams: Support through Continuous Coordination. *Software, IEEE*, Vol. 26, No. 6, pp. 35–40, doi: 10.1109/MS.2009.192.
10. **Bruegge, B., Dutoit, A.H., & Wolf, T. (2006).** Sysiphus: enabling informal collaboration in global software development. *Proc. of the IEEE International Conference on Global Software Engineering (ICGSE '06)*, Florianopolis, Brazil.
11. **Bruegge, B., Lucia, A., Fasano, F., & Tortora, G. (2006).** Supporting Distributed Software Development with fine-grained Artefact Management. *Proc. of the IEEE international conference on Global Software Engineering*.
12. **Bailey, B.P. & Iqbal, S.T. (2008).** Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Trans. Comput.-Hum. Interact.*, Vol. 14, No. 4, pp. 1–28, doi: 10.1145/1314683.1314689.
13. **Bellur, U. (2006).** An academic perspective on globalization in the software industry. *Proc. of The 30th annual international computer software and applications conference (COMPSAC'06)*.
14. **Brooke, J. (1996).** SUS: A quick and dirty usability scale. *Proc. of the Usability evaluation in industry*.
15. **Speier, C., Vessey, I., & Valacich, J.S. (2003).** The effects of interruptions, task complexity, and information. *Decision Sciences*, Vol. 34, No. 4, pp. 771–797.
16. **Communicator. (2014).** *What is communicator for Mac?* Retrieved May 21, 2014.
17. **Damian, D., Lanubile, F., & Mallardo, T. (2006).** The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations. *Proc. of the 28th international conference on Software engineering*, Shanghai, China.
18. **Damian, D., Chisan, J., Allen, P., & Corrie, B. (2003).** Awareness meets requirements management: Awareness needs in global software development. *Proc. of the International Workshop on Global Software Development 2003*, Portland, Oregon.
19. **Damian, D. & Moitra, D. (2006).** Guest editors' Introduction: Global Software Development: How far have we come? *IEEE Software*, Vol. 23, No. 5, pp. 17–19.
20. **Davis, F. (1989).** Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, Vol. 13, pp. 319–340.
21. **Dullemond, K. (2010).** ASPIC: Awareness-Based Support Project for Interpersonal Collaboration in Software Engineering. *Proc. of the Global Software Engineering (ICGSE), 5th IEEE International Conference on*, Princeton, NJ, USA.
22. **Cutrell, E.B., Czerwinski, M., & Horvitz, E. (2000).** Effects of instant messaging interruptions on computing tasks. *Proc. of the CHI '00 Extended Abstracts on Human Factors in Computing Systems*, The Hague, The Netherlands.
23. **Gilbert, E. & Karahalios, K. (2007).** CodeSaw: A Social Visualization of Distributed Software Development. *Proc. of the INTERACT 2007*.
24. **Isaacs, E., Walendowski, A., & Ranganthan, D. (2002).** Hubbub: a sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. *Proc. of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, Minneapolis, Minnesota, USA.

25. **Mark, G., Gudith, D., & Klocke, U. (2008).** The cost of interrupted work: more speed and stress. *Proc. of The Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, Florence, Italy.
26. **Spanjers, H., ter Huurne, M., Graaf, B., Lormans, M., Bendas, D., & van Solingen, R. (2006).** Tool support for distributed software engineering. *Proc. of the IEEE International Conference on Global Software Engineering (ICGSE '06)*, Florianopolis, Brazil.
27. **Ellis, J. & Kvavilashvili, L. (2000).** Prospective memory in 2000: Past, presente and future directions. *Applied Cognitive Psychology*, Vol 14, No. 7, pp. 1–9.
28. **Biehl, J.T., Czerwinski, M., Smith, G., & Robertson, G.G. (2007).** FASTDash: a visual dashboard for fostering awareness in software teams. *Proc. of the SIGCHI conference on Human factors in computing systems*, San Jose, California, USA.
29. **Herbsleb, J.D., & Mockus, A. (2003).** An Empirical Study of Speed and Communication in Globally Distributed Software Development. *Software Engineering*, Vol. 29, No. 6, pp. 481–494.
30. **Gido, J. & Clemens, J. (2003).** *Administración exitosa de proyectos (2da ed.)*. Thomson, Mexico.
31. **Schneider, K.A., Gutwin, C., Penner, R., & Paquette, D. (2004).** Mining a software developer's local interaction history. *Proc. of the 1st International Workshop on Mining Software Repositories (MSR'04)*.
32. **Elliot, K., Neustaedter, C., & Greenberg, S. (2006).** Sticky Spots: A Location-Based Messaging System for the Home Video. *Proc. of ACM Conference on Computer Supported Cooperative Work - ACM CSCW'06*.
33. **Kwan, I., Schroter, A., & Damian, D. (2011).** Does Socio-Technical Congruence Have An Effect on Software Build Success? A Study of Coordination in a Software Project. *IEEE Transaction on Software Engineering*, Vol. 37, No. 3, pp. 307–324.
34. **Layman, L., Williams, L., Damian, D., & Bures, H. (2006).** Essential communication practices for Extreme Programming in a global software development team. *Information and software technology*, Vol. 48, No. 9, pp. 781–794.
35. **Cataldo, M. & Herbsleb, J.D. (2008).** Communication patterns in geographically distributed software development and engineers' contributions to the development effort. *Proc. of the 2008 international workshop on Cooperative and human aspects of software engineering*, Leipzig, Germany.
36. **Czerwinski, M., Horvitz, E., & Wilhite, S. (2004).** A diary study of task switching and interruptions. *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*.
37. **McChesney, I. (1997).** Effective coordination in the software process - historical perspectives and future directions. *Software Quality Control*, Vol. 6, No. 3, pp 235–246.
38. **Tobis, M. & Tobis, I. (2002).** *Managing Multiple Projects*. New York: McGraw-Hill.
39. **Min, Q., Liu, Zh., & Ji, Sh. (2010).** Communication Effectiveness in Global Virtual Teams: A Case Study of Software Outsourcing Industry in China. *Proc. of the 43rd Hawaii International Conference on System Sciences*, Koloa, Kauai, Hawaii, USA.
40. **Mandviwalla, M., & Olfman, L. (1994).** What do groups need? A proposed set of generic groupware requirements. *ACM Trans. Comput.-Hum. Interact.*, Vol. 1, No. 3, pp. 245–268, doi: 10.1145/196699.196715.
41. **Boulila, N., Dutoit, A.H., & Brügge, B. (2003).** D-Meeting: An Object-Oriented Framework for supporting distributing modeling of software. *Proc. of the ICSE International Workshop on Global Software Development*, Portland, Oregon.
42. **Romero, N., McEwan, G., & Greenberg, S. (2007).** A field study of community bar: (mis)-matches between theory and practice. *Proc. of the international ACM conference on Supporting group work*, Sanibel Island, Florida, USA.
43. **Adamczyk, P.D., Iqbal, S.T., & Bailey, B.P. (2005).** A method, system, and tools for intelligent interruption management. *Proc. of the 4th international workshop on Task models and diagrams*, Gdansk, Poland.
44. **Dourish, P., & Bellotti, V. (1992).** Awareness and Coordination in Shared Workspaces. *Proc. of the ACM Conference on Computer-Supported Cooperative Work CSCW'92*, Toronto, Ontario.
45. **Ovaska, P., Rossi, M., & Marttiin, P. (2003).** Architecture as a coordination tool in multi-site software development. *Software Process: Improvement and Practice*, Vol. 8, No. 4, pp. 233–247.
46. **Pressman, R.S. (2005).** *Software Engineering: A Practitioner's Approach (6th ed.)*. New York: McGraw Hill.
47. **Kraut, R.E. & Streeter, L.A. (1995).** Coordination in software development. *Commun. ACM*, Vol. 38, No. 3, pp. 69–81.

48. **Holmes, R. & Walker, R.J. (2008).** Promoting developer-specific awareness. *Proc. of the 2008 international workshop on Cooperative and human aspects of software engineering*, Leipzig, Germany.
49. **Kraut, R., Fish, R., Root, B., & Chalfonte, B. (1990).** Informal communication in organizations: Form, function and technology. *Proc. of the People's reactions to technology in factories, offices and aerospace, The Claremont Symposium on Applied Social Psychology*, Sage Publications.
50. **Prikladnicki, R., Audy, J.L.N., & Evaristo, J.R. (2003).** Distributed Software Development: Toward an Understanding of the Relationship between Project Team, Users and Customers. *Proc. of the 5th Int'l Conf. Enterprise Information Systems (ICEIS 03)*, Angers, France.
51. **Fussell, S., Kiesler, S., Setlock, L.D., & Scupelli, P. (2004).** Effects of instant messaging on the management of multiple project trajectories. *Paper Proc. of the SIGCHI Conference on Human Factors in Computing Systems*.
52. **Sarkar, S., Sindhgatta, R., & Pooloth, K. (2008).** A collaborative platform for application knowledge management in software maintenance projects. *Proc. of the 1st Bangalore annual Compute conference*, Bangalore, India.
53. **Salinger, S., Oezbek, C., Beecher, K., & Schenk, J. (2010).** Saros: an eclipse plug-in for distributed party programming. *Proc. of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, Cape Town, South Africa.
54. **Sarma, A. (2003).** Palantir: Raising Awareness among Configuration Management Workspaces. *Proc. of the Software Engineering, 2003.. 25th International Conference on*, Portland, OR, USA, doi: 10.1109/ICSE.2003.1201222.
55. **Skype. (2014).** What is Skype? Retrieved May 21, 2014.
56. **Strauss, A. (1985).** Work and the division of labor. *The Sociological Quarterly*, Vol. 26, No. 1, pp. 1–19.
57. **Treude, C., & Storey, M.-A. (2010).** Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds. *Proc. of the 32nd ACM/IEEE International Conference on Software Engineering*, Vol. 1, Cape Town, South Africa.
58. **Bellotti, V., Ducheneaut, N., Howard, M., & Smith, I. (2002).** Taskmaster: recasting email as task management. (Position paper) *Proc. of the CSCW'02 workshop on Redesigning Email for the 21st Century*.
59. **González, V., Galicia, L., & Favela, J. (2007).** Supporting the Planning and Organization of Multiple Activities in the Workplace. *Proc. of the INTERACT 2007*.
60. **Gonzalez, V. & Mark, G. (2004).** Constant, Constant, Multi-tasking Craziness: Managing Multiple Working Spheres. *Proc. of The SIGCHI Conference on Human Factors in Computing Systems*.
61. **Xiao, W., Chi, C., & Yang, M. (2007).** On-line collaborative software development via wiki. *Proc. of the 2007 international symposium on Wikis*, Montreal, Quebec, Canada.
62. **Wong, S. (2010).** An Architecture-Centric Approach to Coordination. *Proc. of the Global Software Engineering (ICGSE), 5th IEEE International Conference*, Princeton, NJ, USA.
63. **Ye, Y. (2006).** Supporting software development as knowledge-intensive and collaborative activity. *Proc. of the 2006 international workshop on interdisciplinary software engineering research*, Shanghai, China.
- Ramón René Palacio Cinco** es profesor de Ingeniería de Software en el Instituto Tecnológico de Sonora (ITSON) en Navojoa, Sonora, México, donde es miembro del cuerpo académico Redes y Sistemas Inteligentes. Sus intereses de investigación incluyen Redes de Computadoras, Interacción Humano-Computadora (HCI), Cómputo móvil y ubicuo, e Ingeniería de Software. Con su grupo de investigación realiza estudios de campo para obtener un mejor entendimiento de las prácticas actuales de trabajo y basado en ese entendimiento, diseña, desarrollar y evalúa tecnologías para diversos ambientes de trabajo. Es licenciado en Sistemas Computacionales Administrativos por el Centro de Estudios Superiores del Estado de Sonora (CESUES), México; es maestro en Administración de Tecnologías de Información, por el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), México; y doctor en ciencias en Ciencias de la Computación por la Universidad Autónoma de Baja California (UABC), México.
- Alberto L. Morán y Solares** es profesor de Ciencias Computacionales en la Universidad Autónoma de Baja California (UABC) en Ensenada, B.C., México, donde es líder del

cuerpo académico Tecnologías para Ambientes Inteligentes. Sus intereses de investigación incluyen Interacción Humano-Computadora (HCI), Cómputo móvil y ubicuo, CSCW e Ingeniería de Software. Con su grupo de investigación realiza estudios de campo para obtener un mejor entendimiento de las prácticas actuales de trabajo y basado en ese entendimiento, diseña y evalúa tecnologías de ambientes inteligentes para diversos ambientes de trabajo, incluyendo hospitales, casas de cuidado y compañías de desarrollo de software. Es licenciado en Ciencias Computacionales por la Universidad Autónoma de Baja California (UABC), México; es maestro en ciencias en Ciencias de la Computación, por el Centro de Investigación y de Educación Superior de Ensenada (CICESE), México; y doctor en ciencias en Ciencias de la Computación por el Institut National Polytechnique de Grenoble (INPG), Francia.

Víctor Manuel González y González es profesor en Interacción Humano-Computadora en el Departamento Académico de Computación del

Instituto Tecnológico Autónomo de México (ITAM). Se especializa en la investigación de la Interacción Humano-Computadora y cuenta con amplia experiencia en la investigación del uso y adopción de Tecnologías de Información y Comunicación (TIC), con un énfasis en los sistemas interactivos. Sus áreas de interés son la Visualización de Información, las Analíticas Visuales, las Interfaces Cerebro-Computadora, el Modelado Cognitivo de las prácticas de la programación de software en el contexto de la fragmentación del trabajo. Dr. González recibió grados de Doctor en Filosofía (Ph.D.) y Maestro en Ciencias en las áreas de Ciencias de la Información y Computación en la Universidad de California en Irvine (EEUU), el grado de Maestro en Ciencias con especialidad en Telecomunicaciones y Sistemas de Información por la Universidad de Essex (Reino Unido) y el grado de Ingeniero en Electrónica y Comunicaciones por el Tecnológico de Monterrey (México).

Article received on 15/03/2013, accepted on 14/10/2014.