

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA ENSENADA



Soporte a la entrada en colaboración en el desarrollo distribuido de software mediante esferas de trabajo colaborativas

T E S I S

que presenta para obtener el grado de DOCTOR EN CIENCIAS

RAMÓN RENÉ PALACIO CINCO

**DIRECTOR DE TESIS:
DR. ALBERTO LEOPOLDO MORÁN Y SOLARES**

**CO-DIRECTOR DE TESIS:
DR. VÍCTOR MANUEL GONZÁLEZ Y GONZÁLEZ**

ENSENADA, B. C.

AGOSTO DE 2010

RESUMEN de la Tesis de **Ramón René Palacio Cinco**, presentada como requisito parcial para la obtención del grado de DOCTOR EN CIENCIAS. Ensenada, Baja California. Agosto de 2010.

**SOPORTE A LA ENTRADA EN COLABORACIÓN EN EL DESARROLLO
DISTRIBUIDO DE SOFTWARE MEDIANTE ESFERAS DE TRABAJO
COLABORATIVAS**

Resumen aprobado por:



Dr. Alberto Leopoldo Morán y Solares
Director de tesis



Dr. Víctor Manuel González y González
Co-Director de tesis

Las organizaciones de desarrollo de software se están enfrentando a un cambio de paradigma hacia el Desarrollo Distribuido Software (DSD). Este cambio presenta situaciones de las cuales las organizaciones pueden beneficiarse, tales como recursos humanos altamente calificados y grupos de desarrollo más cercanos a la ubicación de los clientes, entre otros. Sin embargo, también presenta desafíos a los que las organizaciones tienen que adaptarse, como dificultades de coordinación, gestión del conocimiento y comunicación inadecuadas y la escasez de relaciones de confianza entre el equipo virtual.

En esta tesis, en particular se estudian la falta de oportunidades adecuadas para interacciones informales en ambientes de DSD. Las interacciones informales han sido identificadas como base para el logro de coordinación, comunicación y las limitaciones de las relaciones de confianza. Dada la naturaleza de los trabajadores en ambientes de DSD, las interacciones informales principalmente están soportadas por tecnologías de comunicación. Sin embargo, estas tecnologías (p. ej. el teléfono y la mensajería instantánea) no consideran el contexto de receptor remoto al momento de iniciar una interacción, de tal manera que los intentos de interacción son altamente intrusivos para el receptor, y el beneficio de esto es principalmente para el emisor de una interacción.

Este trabajo de tesis intenta enfrentar este problema mediante la obtención de información de las actividades personales de los colegas remotos, y el suministro de esta información al emisor de una interacción, de tal forma que el emisor podría contar con información suficiente sobre la actividad actual del receptor remoto, y basado en ésta base podrá decidir si es el momento adecuado para iniciar una interacción.

Para lograr esto, se introduce y define el concepto de Esferas de Trabajo Colaborativas (CWS, por sus siglas en inglés), mediante la cual los desarrolladores

obtienen información relacionada a las actividades personales de sus colegas distribuidos. CWS permite identificar oportunidades para colaborar en momentos adecuados tanto para el que hace el contacto y el que está siendo contactado. Se argumenta que con CWS, los desarrolladores de software serán capaces de tener conciencia acerca del estado y progreso de un colega en cuanto a lo que han logrado en alguna actividad, y utilizar esta información para informar el inicio de una interacción para la colaborar.

Se ilustra este concepto con el diseño y desarrollo de CWS-IM, una herramienta de mensajería instantánea que soporta Esferas de Trabajo Colaborativas para Desarrolladores Distribuidos de Software. El uso actual de CWS-IM fue evaluado mediante su introducción dentro de un equipo de DSD en una fábrica de software, donde se llevó a cabo un estudio de tres semanas. La percepción de utilidad y facilidad de uso fue evaluada con 16 desarrolladores participantes mediante las escalas SUS y TAM. La evidencia de los resultados de la evaluación indican que CWS-IM fue bien recibido por los desarrolladores DSD participantes. Ellos percibieron a CWS-IM como más útil que las aplicaciones de mensajería instantánea tradicional, y fácil de usar para coordinación y comunicación durante la realización de sus actividades de DSD.

Palabras clave: *Sistemas Colaborativos, Desarrollo Distribuido de Software, Manejo de Interrupciones, Ingeniería de Software.*

ABSTRACT of the thesis, presented by **Ramón René Palacio Cinco**, in order to obtain the DOCTOR of SCIENCE.. Ensenada, Baja California, México. August 2010.

SUPPORT FOR STARTING COLLABORATION IN DISTRIBUTED SOFTWARE DEVELOPMENT THROUGH COLLABORATIVE WORKING SPHERES

Approved by:



Dr. Alberto Leopoldo Morán y Solares
Thesis Advisor



Dr. Víctor Manuel González y González
Thesis Co-Advisor

Abstract body

Software development organizations are facing a paradigm shift towards Distributed Software Development (DSD). This shift introduces situations from which organizations may benefit, including highly skilled human resources, and development groups closer to the location of clients, among others. However, it also introduces challenges to which organizations have to adapt, such as coordination difficulties, inadequate knowledge management and communication, and lack of inter-virtual-team trust relationships.

In this thesis, we particularly study the lack of timely adequate opportunities for informal interaction in DSD settings. Informal interactions have been identified as an underpinning foundation to overcome coordination, communication and trust limitations. Given the distributed nature of the workers in DSD settings, informal interactions among them are mainly supported by communication technologies. However, these technologies (e.g. the telephone and Instant Messenger systems) do not consider the context of the remote receiver at the moment of starting an interaction, thus resulting in interaction attempts that are highly intrusive for the receiver, and which chiefly benefit the issuer of an interaction.

We attempted to confront this problem through obtaining information from the personal activities of remote colleagues and providing it to the issuer of an interaction, so that the issuer could have enough information about the current activity of the remote receiver, and based on this be able to decide whether the time is right to start an interaction.

To achieve this, we introduce and define the concept of Collaborative Working Spheres (CWS), through which developers obtain information related to the personal

activities of their distributed colleagues. CWS allow identifying opportunities for collaboration in suitable moments both for the one making contact and the one being contacted. We argue that with CWS, software developers will be able to become aware about the status and progress their partners have achieved in some activity, and use this information to inform their starting an interaction for collaboration.

We illustrate this concept with the design and development of CWS-IM, an IM-based tool that supports Collaborative Working Spheres for Distributed Software Developers. Actual use of CWS-IM was evaluated by introducing it into a DSD team in a software factory, where a three-week study was conducted. Perception of usefulness and ease of use was evaluated with 16 developer participants by means of SUS and TAM scales. Evidence from the evaluation results indicates that CWS-IM was well received by the participating DSD developers. They perceived CWS-IM as being more useful than traditional IM applications, and as easy to use, for coordination and communication during the execution of their DSD activities.

Keywords: *Collaborative Systems, Distributed Software Development, Interruptions Management, Software Engineering.*

Dedicatoria

*A mi esposa, hija, padres y hermanos.
Sin su apoyo no lo hubiera logrado.*

Agradecimientos

Primeramente quiero dar gracias a Dios por darme la fuerza y sabiduría necesaria para concluir este trabajo.

Quiero dar un agradecimiento muy especial a mis dos directores de tesis: Dr. Alberto Leopoldo Morán y Solares y el Dr. Víctor Manuel González y González. Por haberme guiado en esta difícil tarea. Gracias de todo corazón, espero que durante este nuevo camino que inicia continuemos fomentando esta amistad y colaboración que ha nacido entre nosotros.

A los miembros de mi comité de tesis Dra. Marcela Deyanira Rodríguez Urrea, Dr. José Ángel González Fraga, y al Dr. Juan Iván Nieto Hipólito, cuyos consejos y comentarios acertados contribuyeron en gran medida al desarrollo de esta tesis y a los resultados derivados de ella.

Al British Council RxP Program Award por el apoyo para mi estancia de investigación en la Universidad de Manchester, en Manchester, Inglaterra.

Al grupo ALARCOS de la Escuela Superior de Informática de la Universidad de Castilla-La Mancha, en Ciudad Real, España, por haberme recibido y apoyado durante mi estancia académica. En especial a la Dra. Aurora Vizcaíno B. quien contribuyó de gran forma para el logro de esta tesis y en gran parte de las publicaciones reportadas.

A mi esposa e hija, Karina E. y Mya Karina quiénes tuvieron la voluntad de ser pacientes, y me dieron la fuerza con su amor y cariño para sacar adelante este proyecto tan importante en mi carrera.

A mis padres, Ramón René y María Luisa, que supieron inculcarme el gusto por el estudio y por prepararme día con día, no sólo en lo profesional, sino en la vida; y a mis hermanos, José Luis y Denisse Araceli.

A todos los amigos que hice en Ensenada, especialmente a Vicky, Sandra, Paco y Raúl, con quienes me tocó convivir y compartir experiencias durante mi estancia en la Universidad.

A la familia González Parra (Luis, Juanita y Angélica) por habernos dado la oportunidad de formar parte de su familia desde nuestra llegada a Ensenada.

Al Instituto Tecnológico de Sonora, que otorgó los medios necesarios para la realización de mis estudios de Doctorado.

Al Programa de Mejoramiento del Profesorado (*PROMEP*), por la beca otorgada desde Agosto de 2006 a Julio de 2010.

Y a todos aquellos que de una u otra forma intervinieron en el logro de esta meta y en facilitar mi estancia en Ensenada durante el trayecto.

ÍNDICE

<i>Dedicatoria</i>	vi
<i>Agradecimientos</i>	vii
ÍNDICE	ix
Índice de figuras	xii
Índice de tablas	xiii
1. Introducción	1
1.1 Antecedentes	2
1.2 Planteamiento del problema	8
1.3 Objetivos de la investigación	9
1.4 Metodología de investigación	9
1.5 Contribuciones principales de esta tesis.....	11
1.6 Contenido del documento	14
2. Marco teórico	15
2.1 Desarrollo distribuido de software	16
2.1.1 Beneficios y Desafíos.....	17
2.1.2 Entorno de trabajo.....	19
2.2 Modelo de trabajo propuesto para el DSD	22
2.2.1 Conceptualización de instancias y relaciones	23
2.2.2 Visualización general del modelo	26
2.3 Articulación de trabajo y su proceso.....	29
2.3.1 Modelado del proceso de trabajo en gráfica rica	31
2.3.2 Modelado de actividades.....	33
2.4 Resumen del capítulo	36
3. Trabajos relacionados	37
3.1 Interrupciones.....	38
3.1.1 Manejo de interrupciones.....	38
3.1.2 Diminución del impacto negativo	40
3.2 Desarrollo distribuido de software y el papel de la comunicación.	41
3.2.1 Mejorando la comunicación inadecuada en DSD.....	41

3.2.2 Distribución de información de conciencia	43
3.2.3 Coordinación.....	46
3.2.4 Costos vulnerables	51
3.2.5 Mantener relaciones de confianza.....	52
3.2.6 Entrar en colaboración.	54
3.3 Hallazgos identificados	55
3.4 Resumen del capítulo	57
4. Esferas de trabajo colaborativas.....	59
4.1 Características de las actividades del desarrollo de software.....	60
4.2 Gestión personal de actividades	63
4.2.1 Esfera de trabajo.....	63
4.3 Esferas de trabajo colaborativas.....	65
4.3.1 Requerimientos identificados.....	66
4.3.2 Implicaciones de diseño	69
4.3.3 Modelo conceptual.....	71
4.3.4 Características	73
4.3.5 Representación propuesta	75
4.3.6 Definición.....	77
4.4 Resumen del capítulo	79
5. Identificando las preferencias de los trabajadores del DSD: estudio preliminar..	80
5.1 CWS-IM: Primer versión de interfaces de usuario	81
5.2 Descripción de los escenarios de trabajo	83
5.3 Objetivos del estudio.....	88
5.4 Hipótesis.....	88
5.5 Diseño del estudio.....	90
5.6 Participantes	90
5.7 Procedimiento	91
5.8 Limitaciones	91
5.9 Resultados	92
5.10 Discusión.....	96
5.11 Resumen del capítulo	98
6. CWS-IM: un mensajero instantáneo con soporte para esferas de trabajo colaborativas	100

6.1 Interfaz gráfica	101
6.1.1 Elementos de información	101
6.1.2 Mecanismos	103
6.1.3 Componentes de la interfaz gráfica de usuario	106
6.2 Obtención de información.....	110
6.3 Implantación.....	113
6.4 Funcionalidad de CWS-IM	116
6.4.1 Interpretación del código de colores y el rol.....	117
6.5 Resumen del capítulo	120
7. Evaluación de un caso de estudio	121
7.1 Objetivos	122
7.2 Hipótesis.....	122
7.3 Diseño del estudio	124
7.4 Participantes	124
7.5 Procedimiento	125
7.6 Limitaciones.....	126
7.7 Resultados	127
7.8 Discusión.....	132
7.9 Resumen del capítulo.....	136
8. Conclusiones, aportaciones y trabajo futuro	137
8.1 Conclusiones	138
8.2 Aportaciones	140
8.3 Trabajo futuro	150
Referencias	152
APÉNDICE A. CUESTIONARIO TAM (Technology Acceptance Model).....	163
APÉNDICE B. Proceso de análisis y diseño del sistema multi-agente basado en la metodología ingenias.	164
APÉNDICE C. CUESTIONARIO SUS (System Usability Scale).....	167

Índice de figuras

Figura 1.1. Modelo de colaboración para investigación científica (tomado de [8]).	3
Figura 1.2. Matriz de espacio y tiempo (tomado de [10]).	4
Figura 1.3. Diagrama de estado: entrar en colaboración.	7
Figura 1.4. Metodología seguida durante el proceso de investigación.	10
Figura 2.1. Modelo de interrelación de instancias centrado en la unidad de trabajo para DSD.	24
Figura 2.2. Modelo de trabajo: perspectiva especificaciones.	26
Figura 2.3. Modelo de trabajo: perspectiva esfera de trabajo.	27
Figura 2.4. Modelo de trabajo: perspectiva ejecución.	27
Figura 2.5. Proceso de articulación de unidades de trabajo en DSD.	32
Figura 2.6. Diagrama de actividades del proceso de articulación de unidades de trabajo en DSD.	34
Figura 4.1. Modelo conceptual de esferas de trabajo colaborativas.	72
Figura 4.2. Representación desde la perspectiva individual.	75
Figura 4.3. Representación desde la perspectiva grupal.	76
Figura 5.1. Primera versión de interfaces de usuario para CWS-IM.	83
Figura 5.2. Interfaces para la condición con uso de la Mensajería Instantánea tradicional.	85
Figura 5.3. Interfaces para la condición con uso de CWS-IM.	86
Figura 6.1. Componentes GUI de CWS-IM: Vista de contactos.	108
Figura 6.2. Ventana para conversación en grupo.	109
Figura 6.3. Componentes GUI de CWS-IM: Vista de tareas.	110
Figura 6.4. Distribución de agentes.	112
Figura 6.5. Diagrama arquitectura.	114
Figura 6.6. Representación de un espacio de colaboración compartido.	116
Figura 6.7. Lista de contactos de Jimmy.	119

Índice de tablas

Tabla 2.1. Conceptos de articulación de trabajo y desarrollo de software.	30
Tabla 2.2. Notación del modelado.	31
Tabla 4.1. Requerimientos identificados para interacción.	67
Tabla 4.2. Implicaciones de diseño.	70
Tabla 4.3. Características de esferas de trabajo colaborativas.	74
Tabla 5.1. Elementos de información para CWS-IM	82
Tabla 5.2. Resultados TAM referentes al escenario 1.	93
Tabla 5.3. Resultados TAM referentes al escenario 2.	94
Tabla 5.4. Resultados TAM referentes al escenario 3.	96
Tabla 6.1. Elementos de información para la interfaz CWS-IM.	103
Tabla 6.2. Elementos de información para CWS-IM y mecanismos para obtenerlos.	105
Tabla 6.3. Componentes de interfaz grafica de usuario (GUI).	106

Capítulo 1

*“Investigar significa pagar la entrada por adelantado
y entrar sin saber lo que se va a ver”.*

Julius Robert Oppenheimer

*“Investigar es ver lo que todo el mundo ha visto,
y pensar lo que nadie más ha pensado”.*

Albert Szent-Györgi

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 [1]. 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 [2-4]. 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 [5, 6], las cuales causan efectos negativos en la realización de tareas [7].

Esta tesis de manera particular aborda los 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, la cual aporta ideas de diseño para el desarrollo de una herramienta prototipo. Esta 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 de DSD en una fábrica de software con la finalidad de evaluarse mediante un caso de estudio. 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 este capítulo introductorio se describen antecedentes y el planteamiento del problema, luego se presentan las preguntas de investigación que guiaron el trabajo, para enseguida definir los objetivos a alcanzar. Posteriormente se detalla la metodología de trabajo utilizada durante el desarrollo del mismo, las principales contribuciones y se finaliza con la descripción del contenido del resto del documento.

1.1 Antecedentes

Dentro de las organizaciones, el trabajo en grupo realizado por sus miembros es fundamental para el alcance de sus propósitos. Por ello, desde la creación de grupos de trabajo, el acuerdo común que rige a los integrantes es el trabajar en conjunto, dicho de otra manera: colaborar. Por lo que para la mayoría de las disciplinas o áreas como la ciencia, el arte, negocios, educación, tecnología, entre otros la generación de nuevas ideas, la realización de tareas y la preparación de informes o reportes son procesos que involucran

una extensa interacción social [8]. Dicha interacción social es parte fundamental para la realización del logro de las metas comunes del grupo (p.ej. un artículo científico).

Según Robert Kraut et al. [8], la colaboración se produce en diferentes escenarios o etapas (iniciación, ejecución y presentación pública) y en diferentes niveles (relaciones y tareas) (ver Figura 1.1). El escenario de inicio requiere, a nivel de la relación, encontrar un socio e interactuar de manera intensa en torno a una idea o suposición (compartir hipótesis a fondo), mientras que en el nivel de tarea, requiere generar ideas y planificar el trabajo.

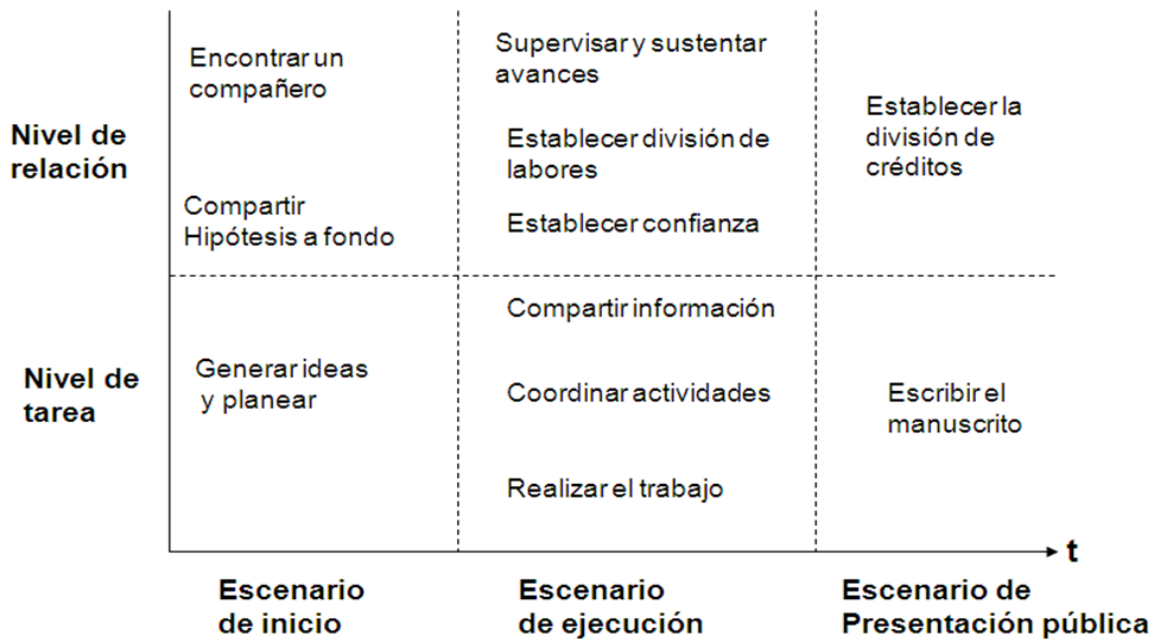


Figura 1.1. Modelo de colaboración para investigación científica (tomado de [8]).

Por otra parte, el escenario de ejecución de un proyecto, en el nivel de tarea, implica el intercambio de información (Comunicación), la coordinación de actividades (Coordinación) y hacer el trabajo (Producción). Estos corresponden a los tres principales ejes de colaboración propuesta por [9]. Cuando las tareas o actividades se realizan en grupos (p. ej. colaborativamente), cada uno de los ejes representa una instancia de colaboración a nivel de tarea. De esta manera, para propósitos de este trabajo se considera que el iniciar un esfuerzo colaborativo como *entrar en colaboración* tanto en el nivel de tarea como en el de actividad. Además, entrar en colaboración en este nivel podría incluir la

interacción con un colega en cualquiera de los tres ejes, incluyendo incluso a todos ellos en la misma interacción.

Las dimensiones de la matriz de espacio-tiempo propuestas por Ellis et al. [10] para caracterizar la interacción humana asistida por computadora, muestra que el trabajo en grupo puede realizarse de manera colocalizada (mismo lugar) o distribuida (diferentes lugares) (ver Figura 1.2).

	Mismo tiempo	Diferentes tiempos
Mismo lugar	Interacción cara a cara	Interacción asíncrona
Diferentes lugares	Interacción distribuida síncrona	Interacción distribuida asíncrona

Figura 1.2. Matriz de espacio y tiempo (tomado de [10]).

En la situación de trabajo llamada colocalizada los miembros del grupo de trabajo se encuentran a la vista unos de otros, es decir si un miembro del grupo trata de saber si un colega de trabajo está disponible para solicitarle ayuda, solo basta con mirar en ese momento hacía su lugar de trabajo para conocer que está haciendo y si se encuentra en su lugar de trabajo. Cuando el colega se encuentra en su lugar de trabajo (mismo lugar - mismo tiempo) es posible realizar interacción cara a cara con él, en caso contrario (mismo lugar - diferentes tiempos) se le podría dejar un mensaje para cuando regrese pueda enterarse del motivo de la interacción.

En cambio, en la situación de trabajo distribuida los miembros del grupo de trabajo no están a la vista, lo que impide obtener información de lo que sucede en su espacio de trabajo en determinado momento (diferentes lugares – mismo tiempo), por lo que generalmente se recurre a herramientas de comunicación síncronos como el teléfono o la videoconferencia para interactuar. En caso contrario (diferentes lugares – diferentes

tiempos) es posible realizar interacción con los miembros del grupo mediante interacción asíncrona.

De manera particular, este trabajo será enfocado a los inicios de colaboración en entornos de trabajo distribuidos síncronos (diferentes lugares – mismo tiempo). Más particularmente en grupos de desarrollo de software que realizan sus actividades en este tipo de situación. Es por ello que a continuación se iniciará con la descripción de este entorno de trabajo.

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 satisface dicha necesidad [11]. El trabajo en entornos de desarrollo de software se caracteriza por la existencia de un alto nivel de comunicación y coordinación entre los participantes [2, 12, 13], lo cual es debido a:

- *La necesidad de tomar una decisión en consenso entre los miembros de un grupo:* se refiere a un proceso de decisión que no sólo busca el acuerdo de la mayoría de los participantes, sino que busca resolver o atenuar las objeciones de la minoría para alcanzar decisiones más satisfactorias. Por ejemplo, durante la aprobación de requerimientos entre el cliente y el analista; la aceptación de un cambio por el desarrollador y el ingeniero de control de cambios, etc.
- *Las decisiones deben darse a conocer de manera precisa y expedita:* consiste en mantener lo más informado posible al grupo de trabajo sobre los avances del proyecto. Esto es realizado con la finalidad de que los miembros del grupo tengan elementos para decidir qué y cuándo realizar las actividades asignadas. Por ejemplo, notificar y entregar versiones nuevas de un documento de diseño.
- *La interacción entre los miembros del equipo de trabajo debe ser regular y frecuente:* esto es debido a la demanda constante de comunicación entre los miembros del grupo durante las fases del proyecto de desarrollo de software. Por ejemplo, la solicitud y entrega de información de avance de las tareas asignadas.
- *El trabajo de desarrollo es cooperativo y colaborativo:* se refiere a que los desarrolladores trabajan de forma convenida y coordinada para producir un

artefacto. Por ejemplo, la programación en pares, la creación y revisión en grupo de un documento de especificación de diseño, etc.

Como se puede notar, lidiar con estas características es muy diferente dependiendo de la manera en cómo se realice el proceso de desarrollo. En el desarrollo colocalizado, los miembros del grupo de trabajo están a la vista o están fácilmente accesibles, por lo cual es posible ver o conocer que están haciendo, prácticamente sin mayor esfuerzo. Por ello resulta fácil saber o intuir cuando es un momento apropiado para interactuar con alguno de los miembros. Por ejemplo, en algunas ocasiones se puede abordar a un colega en algún pasillo de la empresa para pedirle una cita para discutir algo relacionado a una actividad de desarrollo, o incluso discutirlo ahí mismo, lo cual permite mantener la comunicación y la coordinación de buena manera. En el desarrollo distribuido de software (DSD, por sus siglas en inglés) [14], los participantes y los procesos de desarrollo están ubicados en sitios remotos, por lo que no se cuenta con la información de contexto que se tiene en el caso colocalizado, lo cual dificulta los procesos de comunicación, coordinación y producción. Esto es porque lo que era un equipo de desarrollo colocalizado que se reunía periódicamente para conversar y discutir cara a cara, ha pasado a ser un equipo virtual, es decir una red de equipos, cada uno posiblemente situado en un lugar geográfico diferente, que hacen uso de tecnologías de comunicación para interactuar y llevar a cabo una actividad en conjunto. En este caso cuando se requiere contactar a un colega de trabajo (entrar en colaboración), el desarrollador no cuenta con información de la actividad actual del colega remoto. Esto se debe a que las tecnologías de comunicación síncrona que son comúnmente encontradas en sus espacios de trabajo (p. ej. el teléfono y la mensajería instantánea), no proporcionan información de la actividad actual del colega remoto. Por ello, los colegas remotos al ser abordados frecuentemente son interrumpidos durante su actividad [7, 15]. Además se ha confirmado que el trabajo distribuido está asociado con el retardo debido a la interdependencia de actividades entre los sitios de trabajo [16], por lo que se sugiere que la tecnología de comunicación podría ser efectiva si permitiera encontrar a los colaboradores o colegas adecuados, poseer conciencia de su disponibilidad y tener fácil acceso a un medio de comunicación informal. Para lograr esto se ha demostrado que la coordinación de actividades requiere de comunicación constante y efectiva [1, 17].

Cabe mencionar que el DSD organiza y controla su trabajo a través de metodologías de administración de proyectos. Para realizar esto, al inicio de un proyecto se asignan las actividades a los miembros del grupo. Esto es generalmente por el rol o función que los miembros tienen en el proyecto (p. ej. programador, analista, administrador del proyecto, etc.). Al estar asignadas las actividades o tareas del proyecto, cada uno de los involucrados inician con su actividad de manera individual (ver Figura 1.3), durante la cual surge el interés por interactuar. Tal interés puede ser motivado para localizar a un colega, dejar un mensaje, notificar un avance de actividad, obtener ayuda, acordar una reunión o entregar un documento [18]. Cuando se encuentra la oportunidad o posibilidad para interactuar, se inicia el trabajo colaborativo. Al finalizar el trabajo colaborativo, cada uno de los involucrados regresan eventualmente a su estado de trabajo individual.

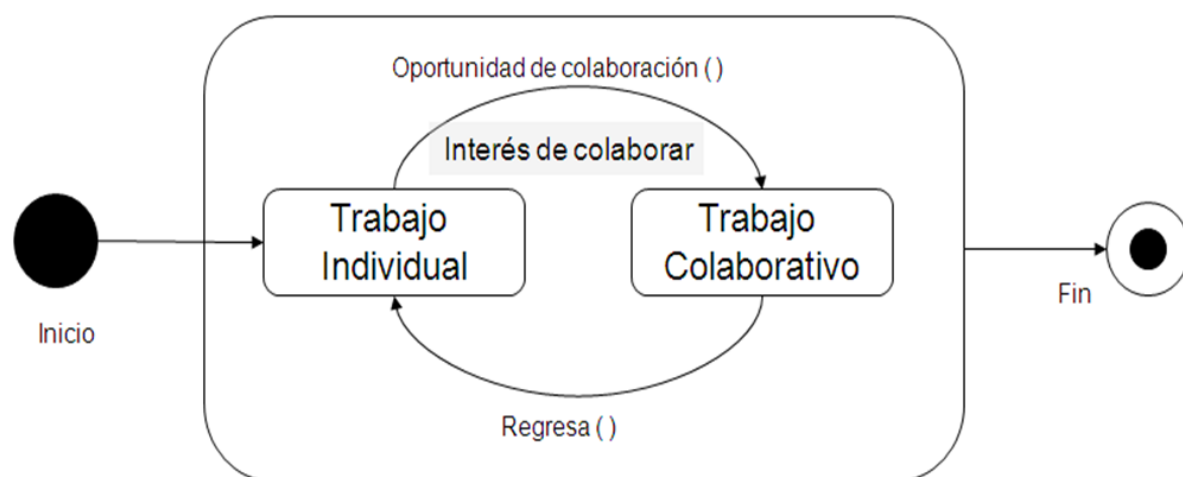


Figura 1.3. Diagrama de estado: entrar en colaboración.

Por ello, para lograr una buena comunicación y coordinación de actividades o tareas con un colega remoto en tiempo real, es necesario conocer el contexto de la actividad que es realizada en un momento determinado. Para esto es necesario que el interesado en iniciar una interacción (emisor) lo haga de manera informada, para así determinar cuál sería el momento más apropiado para iniciar el contacto [19].

1.2 Planteamiento del problema

Con base a la problemática descrita en los antecedentes mencionados previamente, se han definido las siguientes preguntas que guían este trabajo de investigación:

- En primer lugar se requiere dar respuesta a la pregunta: *¿Cuáles son los requerimientos para un sistema que soporte la entrada en colaboración en el desarrollo distribuido de software?*, lo cual nos permitirá identificar las características que debe tener el nuevo sistema. Para esto, se consideran las siguientes sub-preguntas:
 - *¿Qué es necesario para identificar y determinar el momento adecuado para entrar en colaboración?*
 - *¿Cómo y qué tipo de información se necesita obtener de la gestión personal de actividades?*
 - *¿Cómo la gestión personal de actividades puede apoyar a determinar la oportunidad de interacción?*

- Posteriormente será necesario responder a la pregunta: *¿Cómo se puede proporcionar la perspectiva de actividad grupal a los miembros del equipo de trabajo de DSD?*, con la cual se busca obtener el modelado de una propuesta tecnológica que permita guiar el desarrollo de un prototipo. Para ello, se incluyen las siguientes sub-preguntas:
 - *¿Qué elementos de información deben ser obtenidos y distribuidos en DSD en lo individual y lo grupal?*
 - *¿Cómo puede ser obtenida, representada y presentada esta información?*
 - *¿Cómo se determina el momento apropiado para entrar en colaboración?*
 - *¿Cuáles son las características de diseño que debe cumplir la tecnología que brinde apoyo para los inicios de colaboración informados?*

- Finalmente es importante conocer *¿Cuál es el impacto de los inicios de colaboración informados en DSD?*, para lo cual se hace necesario responder la siguiente sub-pregunta:

- ¿Cuál es la percepción de los desarrolladores de DSD en términos de utilidad y facilidad de uso de una herramienta que apoye los inicios de colaboración informados?

1.3 Objetivos de la investigación

Objetivo general:

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.

Objetivos específicos:

- Caracterizar la forma de trabajo de un grupo de desarrolladores distribuidos de software.
- 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.
- Crear un diseño que proporcione apoyo adecuado a los inicios de colaboración informados basado en los requerimientos.
- Desarrollar una herramienta tecnológica prototipo que ofrezca soporte a los inicios de colaboración informados basados en el diseño.
- Evaluar la percepción de utilidad, y de facilidad de uso para determinar la intención de adopción de la herramienta por un grupo de prueba

1.4 Metodología de investigación

La metodología de investigación utilizada para el desarrollo de este trabajo consta de nueve etapas (ver Figura 1.4). Enseguida se describen las actividades llevadas a cabo en las diferentes etapas de esta metodología.

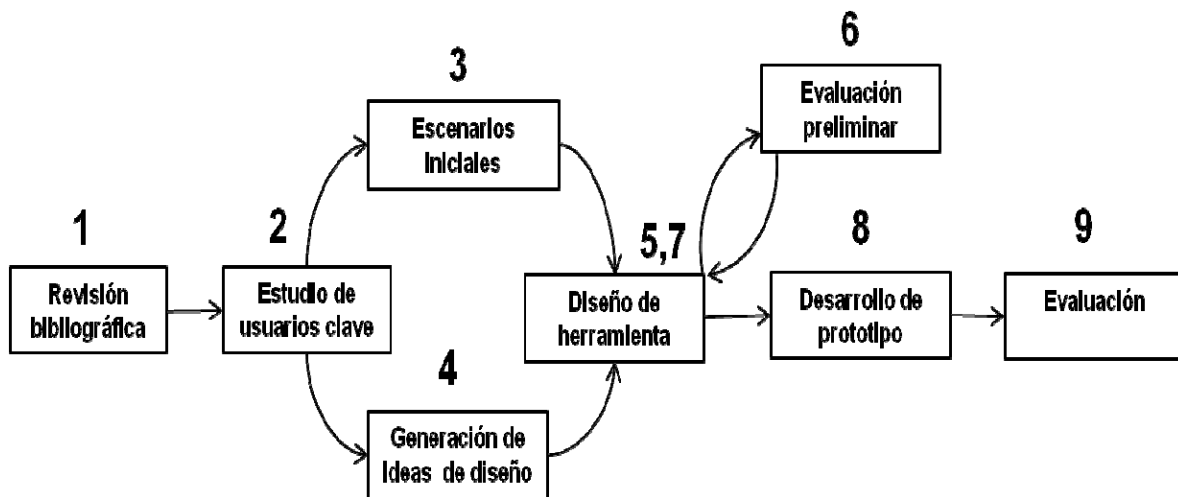


Figura 1.4. Metodología seguida durante el proceso de investigación

1. *Revisión bibliográfica.* En esta etapa se realizó un análisis bibliográfico extenso con la finalidad de identificar los trabajos previos realizados para el tema propuesto y fundamentar el marco teórico para el presente trabajo. Con esta etapa se obtuvo un modelo de trabajo para DSD y un conjunto de hallazgos que sustentan la creación de una nueva propuesta para inicios de colaboración en DSD.
2. *Estudio de usuarios clave.* Esta etapa consistió en realizar un análisis de las actividades que realizan los trabajadores del DSD. La finalidad de esta etapa fue la de comprender e identificar las necesidades de interacción entre los miembros de un grupo de trabajo virtual durante las fases de desarrollo de software. Con esta etapa se obtuvo la caracterización de actividades de desarrollo distribuido de software.
3. *Escenarios iniciales.* Se estudiaron distintos escenarios que representaban la forma en que se realizan las actividades DSD. Con esto se visualizó y contextualizó la problemática referente a las necesidades de interacción de los trabajadores del DSD.
4. *Generación de 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 de DSD. De esta manera se obtuvo la caracterización y modelado del concepto Esferas de Trabajo Colaborativas (CWS, de las siglas en inglés de Collaborative Working Spheres) basado en un conjunto de elementos de información identificados durante la ejecución de esta etapa.

5. *Diseño de herramienta (1ª iteración)*. Esta primera parte del diseño se llevó a cabo con base en las ideas de diseño de la etapa anterior. A partir de éstas se definió la funcionalidad básica inicial que tendría la herramienta y diseñó una primera versión de las interfaces de usuario (GUI, por sus siglas en inglés).
6. *Evaluación preliminar*: Teniendo una primera versión de los diseños de las interfaces de usuario, se procedió a realizar una evaluación basada en escenarios [20] 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.
7. *Diseño de herramienta (2ª iteración)*. Con base en los resultados de la evaluación preliminar se informó el rediseño de la herramienta. Por ejemplo, se corroboraron o modificaron los elementos de información y la forma de ser desplegados en la interfaz.
8. *Desarrollo de prototipo*. Esta etapa consistió en el desarrollo de la herramienta prototipo basado en el diseño refinado de la misma.
9. *Evaluación*. 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. Asimismo, se realizó un análisis detallado de los datos obtenidos durante el estudio, con los cuales se generaron las conclusiones de la investigación.

Resumidas las nueve etapas que conforman la metodología utilizada en esta investigación, se pasará a definir las partes por las que está constituida la presente tesis.

1.5 Contribuciones principales de esta tesis

Las principales contribuciones derivadas de este trabajo de tesis son presentadas a continuación:

1. *Un modelo de trabajo para desarrollo distribuido de software, que integra los distintos conceptos manejados por administración de proyectos de software y elementos de la actividad de desarrollo de software*. Este modelo está basado en un análisis de literatura, tanto del área de ingeniería de software como de administración

de proyectos. Éste fue creado porque no se encontró consistencia en la definición de conceptos y relaciones de las distintas instancias involucradas en el desarrollo de software. Por ello, con el fin de contar con la unificación de tales instancias, se creó un modelo de trabajo para DSD, que sirva de fundamento para el proceso de comprensión de las distintas instancias o entidades que involucran el desarrollo de software. Este modelo de trabajo presentado en el Capítulo 2, fue base para comprender el proceso de trabajo del DSD.

2. *Definición de un conjunto de requerimientos para el logro de inicio de interacciones informados en DSD.* Para obtener estos requerimientos fue necesario revisar otras áreas de investigación como interacción humano computadora y sistemas colaborativos. Esto permitió identificar distintas propuestas de interacción en grupos de desarrollo de software que habían sido reportados por distintos investigadores y al mismo tiempo permitió encontrar el área de oportunidad para conducir esta investigación. Este análisis de literatura presentado en el Capítulo 2, sirvió de apoyo para comprender como se presentan las necesidades de comunicación o interacción entre los trabajadores del DSD y que aspectos se deben considerar para abordarlas.
3. *Definición de un conjunto de hallazgos a considerar para lograr inicios de interacción informados en DSD.* Estos hallazgos fueron obtenidos después de realizar un análisis extenso de literatura de Manejo de Interrupciones y Desarrollo Distribuido de Software (DSD). Este análisis fue enfocado de manera particular en coordinación de actividades, escasez de conciencia y la entrada o inicio de colaboración. Los hallazgos descritos en el Capítulo 3, contribuyeron a la definición de un conjunto de ideas de diseño para la creación de mecanismos para iniciar colaboración en DSD.
4. *Caracterización de las actividades de desarrollo de software.* Para llegar a la obtención de estas características fue necesario un análisis de literatura, en la cual no se identificó una caracterización de actividades que fuera útil para los propósitos de esta tesis. Por lo que se desarrolló una a partir de algunas propuestas existentes. La caracterización resultante de este análisis está descrita en el Capítulo 4, la cual sirvió de base para comprender los elementos que componen este tipo de actividades y para crear la caracterización de esferas de trabajo colaborativas.

5. *Caracterización y modelado de Esferas de Trabajo Colaborativas.* Esta caracterización de las esferas de trabajo colaborativas se fundamentó en la caracterización de las actividades de DSD propuesto y el análisis de literatura de gestión personal de actividades y potencial de colaboración. Esta es presentada en el Capítulo 4, y es parte fundamental de este trabajo, puesto que de aquí surge la propuesta de cómo entrar en colaboración de manera informada.
6. *El resultado de la preferencia del usuario con respecto a la funcionalidad de las esferas de trabajo colaborativas.* Como parte del caso de estudio realizado para mostrar la funcionalidad y utilidad de las Esferas de Trabajo Colaborativas, se obtuvieron una serie de propuestas que contribuyeron a mejorar el diseño de los inicios de interacción informados dentro del ambiente de trabajo de DSD. Estos resultados presentados y discutidos en el Capítulo 5, contribuyeron en buena manera para el desarrollo de un prototipo con soporte para inicios de colaboración.
7. *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 esta tesis, es introducir un concepto nuevo al ambiente de trabajo del DSD. Este desarrollo fue soportado por las implicaciones de diseño y la caracterización de las esferas de trabajo colaborativas. Además se consideraron los resultados del estudio preliminar para redefinir los mecanismos y servicios que proporciona el prototipo. Este prototipo descrito a detalle en el Capítulo 6, fue base para la evaluación del caso de estudio en un ambiente de trabajo real.
8. *Evaluación del prototipo en un caso de estudio en el ambiente de DSD.* Para llevar a cabo esta evaluación se requirió tener una versión funcional del prototipo propuesto para introducirla a un ambiente de DSD. Este caso de estudio permitió evaluar el prototipo por 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. Esta evaluación es presentada en el capítulo 7.

1.6 Contenido del documento

El resto de este documento está dividido en 7 capítulos y 3 apéndices, el contenido de cada uno de ellos se describe brevemente a continuación:

En el capítulo 2 se presentan una serie de trabajos de distintas áreas de investigación, con los cuales se argumenta el marco teórico para entender y establecer en qué consiste el trabajo del DSD. De manera particular, se centran en trabajos de desarrollo distribuido de software destacando los beneficios y sus oportunidades de investigación. También se describe un modelo de trabajo creado a partir del análisis de literatura de administración de proyectos, ingeniería de software, gestión de tareas y la teoría de articulación de trabajo.

En el capítulo 3 se presenta un resumen general de trabajos de investigación que contribuyen algunos elementos que inciden de alguna manera a los inicios de colaboración informados. Estos trabajos son parte de los campos de estudio: manejo de interrupciones y desarrollo distribuido de software.

En capítulo 4 se muestra el entendimiento de las actividades de los desarrolladores distribuidos de software. También se presenta como se ha representado la fragmentación de tareas con el concepto de esferas de trabajo. Dicho concepto se ha retomado para proponer un concepto extendido a través de su integración con el concepto de colaboración potencial. Esto en conjunto se propone como esferas de trabajo colaborativas.

En el Capítulo 5 se describe una evaluación de un diseño de interfaces preliminar basada en escenarios. Dicha evaluación es presentada en cuatro partes: la delimitación de los objetivos que deben alcanzarse, la proposición de las hipótesis, la validación de las hipótesis y el análisis de los resultados. Este estudio fue realizado con el fin de percibir si un grupo de desarrolladores consideraba útil y fácil de utilizar la funcionalidad de la herramienta que denominamos CWS-IM en sus actividades de DSD.

En el capítulo 6 se presenta el re-diseño de las interfaces de usuario para iniciar colaboración informada mediante CWS-IM. Además, se describe como la información es obtenida y distribuida a través del sistema multi-agente y cómo esta información es presentada al usuario final con los componentes de interfaz gráfica de usuario (GUI, por sus siglas en inglés). Finalmente también se muestran los detalles relacionados con la implantación misma de CWS-IM y la descripción de su funcionamiento.

En el capítulo 7 se describe la evaluación de un caso de estudio, el cual consistió en introducir la herramienta CWS-IM en una fábrica de software. Dicha evaluación es presentada en cuatro partes: la delimitación de los objetivos que deben alcanzarse, la proposición de las hipótesis, la validación de las hipótesis y el análisis de los resultados. La finalidad de esta evaluación fue contrastar el uso de la mensajería instantánea tradicional con el CWS-IM en términos de utilidad, facilidad de uso, apoyo al inicio de interacción y el nivel de interrupción.

En el Capítulo 8 se presentan las conclusiones y reflexiones derivadas de este trabajo; las principales aportaciones, entre las que se encuentran las publicaciones derivadas de esta tesis; así como algunas oportunidades de trabajo futuro identificadas durante el desarrollo de la misma.

El apéndice A presenta el cuestionario TAM aplicado al grupo de participantes tanto para la evaluación preliminar como para la evaluación del caso de estudio en la fábrica de software.

El apéndice B presenta el diseño del sistema multi-agente con el uso de la metodología INGENIAS.

Finalmente el apéndice C presenta el cuestionario SUS aplicado en el caso de estudio realizado en la fábrica de software.

Capítulo 2

“Los que se enamoran de la práctica sin la teoría son como los pilotos sin timón ni brújula, que nunca podrán saber a dónde van”.

Leonardo Da Vinci

2. Marco teórico

En este capítulo se presenta un análisis de trabajos de investigación relevantes para el tema de desarrollo distribuido de software (DSD). Dicho análisis permite argumentar en qué consiste el trabajo del DSD. Con esto se logra un entendimiento inicial del entorno de trabajo y se establece la necesidad de contar con tecnologías que permitan inicios de interacción remota de manera informada.

Para abordar esto, se hace necesario el entendimiento de los distintos conceptos y tópicos relacionados con las actividades de DSD. Por ello, en este capítulo se propone y describe un modelo de trabajo, el cual es 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.

Finalmente se identifican una serie de aspectos que deben ser cubiertos para lograr los objetivos de este trabajo.

2.1 Desarrollo distribuido de software

Las organizaciones que se dedican al desarrollo de software se están enfrentando a un cambio emergente de paradigma de trabajo que se dirige hacia una distribución de los procesos y equipos de desarrollo. Dicho cambio se debe a la oportunidad de aprovechar los horarios de trabajo al máximo, obtener provecho de la distribución de los recursos, disminución de costos y estar demográficamente más cerca del consumidor final [21, 22]. Sin embargo, hay aspectos negativos que surgen por la naturaleza misma de esta nueva forma de trabajo, que traen como consecuencia problemas de comunicación entre miembros del grupo de trabajo. Por ello, la comunicación inadecuada es en particular un tema considerado un reto clave para la coordinación apropiada de actividades de desarrollo en esta modalidad de trabajo [4, 14, 17, 23].

Esta nueva modalidad para desarrollar plantea diversos retos para la ingeniería de software, tales como la necesidad de nuevas técnicas, o extensión de las existentes, para soportar la inclusión de servicios provistos por terceros; y nuevos procesos que incluyen mecanismos y herramientas que permitan lidiar con el hecho de que los equipos de desarrollo se encuentran separados (p. ej. análisis de requisitos, diseño, administración, etc.). En este nuevo escenario, lo que tradicionalmente era un equipo de desarrollo co-localizado, que tenía reuniones periódicas cara a cara para conversar y discutir, ha pasado a ser un equipo virtual, es decir una red de equipos con integrantes ubicados en diferentes ubicaciones geográficas, que hacen uso de tecnologías de comunicación para interactuar y realizar una tarea en conjunto. Esta nueva forma de desarrollo se le conoce como Desarrollo Distribuido de Software (DSD, por sus siglas en inglés) [14].

Existen una serie de características que definen escenarios para el DSD. Una de ellas es la distancia entre los distintos miembros o equipos, que pueden variar desde unos cuantos metros (cuando los equipos trabajan en edificios separados pero adyacentes) hasta decenas, cientos o miles de kilómetros (cuando están en ciudades diferentes). Un caso especial del desarrollo distribuido de software es cuando las distancias están entre ciudades de diferentes países, inclusive continentes, el cual es conocido como Desarrollo Global de Software (GSD, por sus siglas en inglés) [14, 17, 24].

Cabe mencionar que el trabajo en entornos de desarrollo de software se caracteriza porque existe un alto nivel de comunicación y coordinación entre los participantes [2, 25].

Esto se debe a que la toma de decisiones requiere un consenso de grupo, las decisiones deben darse a conocer de manera precisa y expedita, la interacción es regular y frecuente entre los miembros del equipo, y el trabajo es cooperativo y colaborativo.

Como se mencionó en el capítulo anterior, realizar el trabajo de DSD implica que los participantes están ubicados en sitios remotos, por lo que no se cuenta con esa información de contexto que se tiene en el caso co-localizado, lo cual dificulta la colaboración. Esto es porque aún cuando este tipo de organizaciones utilizan metodologías formales para administración de proyectos, dichas metodologías se ven limitadas por la falta de comunicación informal [2, 26], lo que genera bajos niveles de confianza entre colegas y un conocimiento pobre acerca del trabajo y del progreso en los sitios remotos [2, 17].

A pesar de los inconvenientes de la comunicación, existe un conjunto de razones que motivan a la industria del software a desarrollar bajo esta nueva modalidad de trabajo. Los principales beneficios y desafíos de trabajar bajo esta modalidad son presentados en la siguiente sub-sección.

2.1.1 Beneficios y Desafíos

Diversos trabajos relacionados en el área de DSD destacan los siguientes beneficios para esta modalidad:

1. Mano de obra calificada: las compañías de software requieren recursos humanos altamente calificados y buscan satisfacer esta necesidad empleando programadores en diversas ciudades y países [22, 24].
2. Cercanía con el cliente: para estar más cerca del mercado y para tener un tiempo de respuesta más corto, muchas compañías han puesto a grupos de desarrollo más cerca de la ubicación de su cliente [3, 22].
3. Creación de grupos de trabajo: los grupos virtuales de desarrollo requieren ser creados rápidamente para explotar oportunidades en este nuevo mercado [27].
4. Trabajo continuo en los proyectos: al trabajar en diversas zonas horarias los grupos de desarrollo pueden laborar continuamente en proyectos críticos [28, 29].

5. Mano de obra de costo accesible: la reducción de costos por contratación de recursos humanos provenientes de lugares donde la mano de obra es más barata [30].

Sin embargo, trabajar bajo este paradigma, enfrenta a las organizaciones con desafíos interesantes, tales como:

1. Diferencias culturales: en un contexto de este tipo las diferencias culturales pueden afectar los proyectos de diferentes maneras, incluyendo la efectividad de la comunicación y coordinación, la toma de decisiones de grupo y el desempeño de los equipos [29, 31].
2. Diferencia horaria: las diferencias de horarios entre los grupos de trabajo, afectan en las actividades donde se necesita una colaboración intensiva entre los grupos implicados, por lo que la comunicación síncrona es difícil de establecer dentro de la jornada de trabajo normal [23].
3. Comunicación inadecuada: las diferencias de lenguaje y la distancia, además de las diferencias culturales y las diferencias horarias (mencionadas previamente), entre otros, dificultan la interacción entre los trabajadores, por lo que los participantes deben de realizar esfuerzos adicionales para lograr que la comunicación se haga de manera efectiva [3, 4].
4. Administración del conocimiento: los trabajadores del DSD necesitan compartir mucha información (p. ej. modelo de negocio, requisitos, decisiones de diseño, etc.) proveniente de diferentes fuentes (p. ej. clientes, proveedores, analistas de requerimientos y de sistema, etc.) y que se encuentran en sitios distantes. Si no se cuenta con un mecanismo efectivo para compartir información y conocimiento no es posible explotar los beneficios del DSD [29].
5. Mantener relaciones de confianza: entre mayor sea la distancia física entre las personas o grupos, más difícil es mantener relaciones de confianza debido a la falta de una comunicación informal y espontánea. Esto, es debido a que este tipo de comunicación brinda diversas y frecuentes oportunidades, no sólo para hablar de cuestiones del trabajo y resolver problemas asociados al proceso de

desarrollo, sino también de los temas sociales y personales que son la base en las relaciones interpersonales [2, 32].

En la siguiente sub-sección se describe el ambiente de trabajo resaltando las necesidades de interacción entre el grupo de trabajo.

2.1.2 Entorno de trabajo

Para entender las necesidades de interacción de los trabajadores del DSD y cómo estas pueden iniciar de una manera informada, es importante entender el entorno de trabajo. Por ello es necesario entender cómo se presentan las situaciones de colaboración a las que se enfrentan los desarrolladores durante su jornada laboral; tales como, consultar a un experto, lograr acuerdos para la realización de una tarea, acordar una reunión, entre otras. Para esto, el uso de escenarios puede ser útil, puesto que brinda la posibilidad de identificar oportunidades para determinar requerimientos de diseño para una aplicación de software. Un escenario es una descripción textual acerca de las actividades, el cual requiere claridad en los temas de interés que se quiere abordar [20]. En este caso, el interés radica en el entendimiento de las interacciones en el entorno de DSD, al que nos queremos enfocar en este trabajo. Por ello, se analizaron trabajos referentes a la coordinación de actividades de DSD [2, 33, 34], el proceso de desarrollo [35] y de administración de proyectos de software [36-38]. Con esto, de manera particular, se detectaron y definieron escenarios que sirvieron de apoyo durante el proceso de entendimiento de las actividades de los trabajadores del DSD. Con la creación de estos escenarios, particularmente fue posible identificar situaciones en que los desarrolladores requieren interacción remota en tiempo real con sus colegas. Para esto los escenarios fueron clasificados en 2 partes. El primero está relacionado con la planeación del proyecto de software; por ejemplo, asignación de actividades, interacciones, etc. Con este escenario se denota la dependencia que existe cuando un administrador de proyectos asigna y notifica las actividades del proyecto a un grupo de trabajadores de DSD. El segundo está enfocado en la etapa de ejecución del proyecto, con lo cual se resalta la dependencia existente en las actividades del proyecto; por ejemplo, la disponibilidad de un compañero para hacer una consulta, conocer el progreso de una actividad, etc. Con este escenario se denota la necesidad de interacción que existe

entre los trabajadores de DSD para poder realizar sus actividades. A continuación se describe un ejemplo para cada escenario:

Asignación de actividades (Planeación): Juan es un líder de proyecto, a quien recientemente le asignaron un cliente con un proyecto muy importante para la empresa. Para esto Juan cuenta con una serie de descripciones y puntos acordados con el cliente en una propuesta de venta. Con esta información, Juan convoca a la reunión de inicio de proyecto, con el fin de plantear las actividades que se deben de realizar y establecer los compromisos que se deben cumplir. Para ello, es necesario especificar las actividades que se van a realizar, presupuestar el esfuerzo en horas de cada actividad, indicar los roles implicados para el proyecto y las fechas de entrega de los productos. Juan en su afán de asegurarse que el proyecto se realice lo mejor posible, en primer lugar requiere consultar y acordar con sus colegas las actividades que se habrán de asignar; en segundo lugar requiere confirmar que la estimación en horas de trabajo de cada actividad sea la adecuada; y en tercer lugar necesita acordar o confirmar los tiempos de entrega de cada actividad. Todo esto para llevar a buen término el desarrollo del proyecto. Puesto que sus colegas de trabajo se encuentran localizados en distintas ubicaciones geográficas, Juan tiene que solicitar la disponibilidad de horario de cada colega para acordar una reunión en grupo. Para esto, Juan utiliza el teléfono, mensajería instantánea y correo electrónico para contactar y acordar el medio de comunicación viable y la fecha para llevar a cabo la reunión. Finalmente el grupo le expresa a Juan la preferencia por utilizar videoconferencia para la reunión en una fecha determinada. Después de haber realizado la reunión, Juan les hace llegar vía correo electrónico los compromisos y acuerdos resultantes.

Buscar a un colega (Ejecución). María es la responsable de las actividades de control de calidad. Ella tiene asignada la actividad “A1” del Proyecto “P1”. Ella se encuentra en la Ciudad de México. Roberto es el arquitecto de software y se encuentra ubicado en Ensenada, BC. En este caso, durante las pruebas del módulo correspondiente a la actividad “A1”, María se da cuenta que el manual (artefacto) no está actualizado. Sin embargo, tiene conocimiento de que el responsable de realizarlo fue Roberto. Con este conocimiento María tiene interés de saber por qué no se ha actualizado dicho manual, pues significa un retraso en la planeación personal de sus actividades. Por ello, María busca a

Roberto mediante el uso del mensajero instantáneo. Roberto en ese momento se encuentra trabajando en una documentación urgente para la actividad “A2” del proyecto “P3” y por consiguiente su actividad actual es interrumpida para interactuar con María.

Como se puede notar en los escenarios, las actividades de desarrollo de software demandan interacciones constantes entre el grupo de trabajo. Aún cuando el desarrollo de software puede estar basado en metodologías preestablecidas sus resultados no son fijos o determinados. Esto se debe a que los requerimientos cambian de un proyecto a otro, incluso cuando se desarrollan para un mismo dominio de aplicación. Por esta razón, el desarrollo de software se torna complejo debido a los problemas de coordinación que se presentan durante el desarrollo de las actividades, las cuales no sólo se presentan de manera común sino de manera inevitable [2].

Por esta razón, resulta interesante conocer el contexto de la actividad individual de los compañeros de trabajo al momento de tener la intención por iniciar interacción. De tal manera que al contar con información de la actividad individual actual con respecto a lo que realizan sus compañeros, los iniciadores de la interacción podrían desistir la interrupción y abordarlo en otro momento o incluso podrían buscar a otro colaborador como alternativa para satisfacer el motivo de la interacción.

Es importante mencionar que las tecnologías de comunicación actuales como el mensajero instantáneo, permiten obtener cierta información del contexto del usuario (p. ej. el estatus de usuario), sin embargo no indican de manera implícita en que actividad del proyecto está trabajando la persona que es buscada. Otras tecnologías como el teléfono y las videoconferencias no ofrecen información del estatus del usuario buscado.

Por esto, para fines de contar con inicios de interacción informados, sería relevante contar con una tecnología que permita identificar e informar el estatus de los miembros según la actividad que realizan de manera síncrona. Con esto, podría ser posible conocer los elementos de información del espacio de trabajo de los posibles receptores, lo cual proporcionaría cierta ayuda al emisor para determinar si inicia una interacción o desiste de ella, porque se considera que el momento no es oportuno.

Para iniciar interacción o colaboración de manera informada en grupos de desarrollo virtual, es importante comprender el trabajo del DSD para entender a detalle en qué

consiste este tipo de entorno. En la siguiente sección se presenta un modelo de trabajo basado en la literatura, el cual facilita un análisis más profundo de las actividades realizadas en este tipo de entornos.

2.2 Modelo de trabajo propuesto para el DSD

Con el fin de entender cómo se lleva a cabo el trabajo de DSD, se realizó un análisis de literatura [14, 21, 33, 35-37, 39-41], en el cual no se identificó una caracterización del trabajo de este tipo de organizaciones. Pero en cambio, se identificaron tres áreas particulares (administración de proyectos, ingeniería de software y gestión de tareas) que aportan diversos elementos relevantes al trabajo de DSD, los cuales son descritos brevemente a continuación:

- Comprender como la industria de software se organiza para la realización de sus actividades (administración de proyectos).
- Analizar el proceso de desarrollo de software durante la realización de actividades (ingeniería de software).
- Identificar y unificar las distintas entidades o instancias que se involucran durante la gestión del proyecto de software (administración de proyectos, ingeniería de software y gestión de tareas).
- Entender la forma en que los desarrolladores se organizan y realizan sus actividades individuales (gestión de tareas).
- Entender como los desarrolladores logran unir sus esfuerzos para completar actividades en grupo (teoría de articulación de trabajo).

Conforme se hizo este análisis de la literatura se definieron instancias y sus relaciones. Esto ayudó a que se lograra un mejor entendimiento respecto a cómo el equipo de trabajo logra sumar los esfuerzos individuales para el cumplimiento de la unidad de trabajo principal (p. ej. proyecto o programa) de la organización.

Con base en los elementos antes descritos, en los siguientes apartados se presenta un modelo de trabajo propuesto para DSD.

2.2.1 Conceptualización de instancias y relaciones

El interés de este modelo está centrado en comprender cómo las organizaciones que se dedican a desarrollar software, desglosan sus actividades en diferentes niveles referentes a la unidad de trabajo. Por ejemplo, algunos autores de administración de proyectos consideran que la jerarquización debiera partir de un proyecto y desglosarse en actividades [37]; otros que el proyecto debiera desglosarse en tareas [36]; e incluso hay quienes argumentan que la jerarquía más alta es un Programa de proyectos [36, 37].

La Figura 2.1 nos muestra el modelo resultante de nuestro análisis. En él se pueden observar las instancias y relaciones que están implicadas en el entorno de trabajo del desarrollo de software.

Nótese que el modelo realiza una generalización con la instancia *Unidad de Trabajo*, con la cual se pretende que el modelo sea más flexible y se adapte a cualquier forma de jerarquizar el trabajo dentro de una organización de desarrollo de software (p. ej. un proyecto se descompone en actividades, las actividades en tareas, las tareas en acciones, etc.).

A continuación se presentan las definiciones para las instancias involucradas en el modelo:

- *Programa*: Según [36, 37] un programa es un conjunto de proyectos estrechamente relacionados que hacen referencia a un objetivo común y se financian y ejecutan como una iniciativa única teniendo un marco de tiempo a largo plazo.
- *Unidad de Trabajo*: Se propone que sean todas las unidades que se desglosan a partir de un programa dado, las cuales incluyen proyectos, actividades y tareas para realizar un trabajo en pro de las metas de la misma unidad de trabajo y de los objetivos del programa. Las unidades de trabajo se realizan en un marco de tiempo dado, el cual puede variar durante su ejecución.

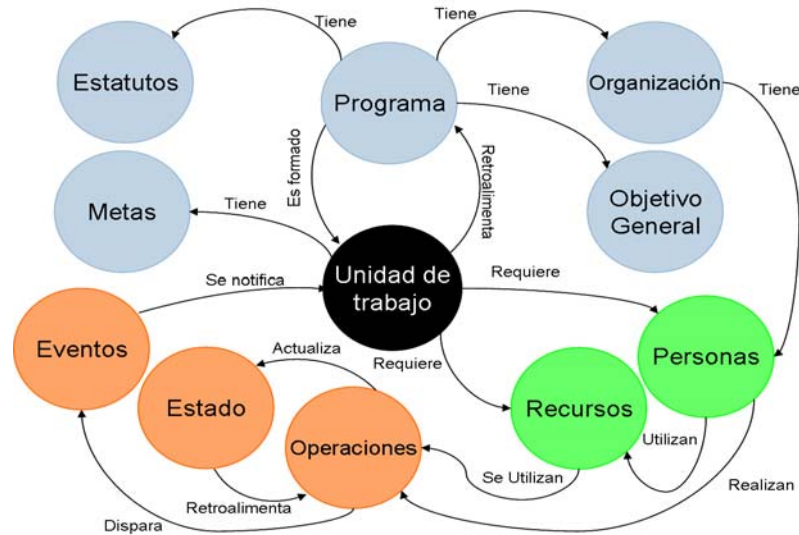


Figura 2.1. Modelo de interrelación de instancias centrado en la unidad de trabajo para DSD.

- *Recursos:* Un recurso es un conjunto de elementos disponibles para completar las unidades de trabajo [37]. Para propósitos del análisis, se considera como tipo de recursos cada cosa que es relevante para un trabajo. Por lo que el modelo considera: 1) recurso de trabajo: aquel que es utilizado como parte de la realización de un trabajo (por ejemplo, documento.doc) y 2) recurso para trabajo: aquel que se utiliza como herramienta necesaria para realizar un trabajo (p. ej., Microsoft Word).
- *Estatutos:* Los autores [42] los definen como las reglas que rigen la operación del programa. Estos especifican el alcance, los resultados a entregar y la forma de adquirir los recursos necesarios.
- *Objetivo General:* Se considera como las especificaciones en forma precisa y directa en relación a lo que se va a hacer. Este constituye la primera instancia de congruencia entre el programa y la(s) unidad (es) de trabajo.
- *Organización:* Es referido como la estructura organizacional, la cual indica la forma de dividir, organizar y coordinar (subestructuras) las unidades de trabajo de la organización. [42, 43]. Los colaboradores están agrupados de manera jerárquica por especialidad o departamentos (producción, administración, recursos humanos etc.)[43].

- *Metas*: Es el fin o la razón por la que se dirigen las acciones sobre una unidad de trabajo [44]. Es el punto final alcanzado por la(s) unidad(es) de trabajo.
- *Personas*: En este documento se considera a los trabajadores que están implicados en el programa, los cuales colaborarán en su desarrollo mediante diferentes roles basados en sus atributos, tales como son experiencia, habilidad, conocimiento, entrenamiento, ocupación, entre otros [43].
- *Operaciones*: Las unidades de trabajo pueden ser descompuestas en operaciones de usuario y operaciones de sistema. Sin embargo estas no pueden ser entendidas sin un marco de referencia creado por la unidad de trabajo correspondiente. Esta definición fue tomada del concepto de acción de [44].
- *Eventos*: Es una condición generada por la ejecución o realización de una unidad de trabajo, aun cuando la realización de esta unidad de trabajo pudiera ser causada por algo externo al dominio de la misma [44].
- *Estado*: Se refiere a la información que es mantenida en un sesión que cambia dinámicamente mientras las personas trabajan en los documentos y la forma en que estas actividades cambian su estado [45].

Las relaciones del modelo presentado en la Figura 1.1 están definidas de la siguiente forma:

- *Tiene*: indica que se contiene dentro de sí a una instancia.
- *Es Formado*: se integra de varios elementos de otra instancia.
- *Retroalimenta*: envía información relevante para una instancia.
- *Requiere*: es necesario para la realización de una unidad de trabajo.
- *Se Notifica*: comunica formalmente a la unidad de trabajo un cambio en su estado.
- *Se Utilizan*: Se requiere para realizar una operación.
- *Realizan*: efectúan o llevan a cabo la operación.
- *Dispara*: Especifica el inicio de algún evento.

2.2.2 Visualización general del modelo

Una vez que fueron definidas las instancias y relaciones del modelo de trabajo, se dedujo que dicho modelo se puede visualizar desde tres perspectivas importantes que facilitan el entendimiento y explicación del trabajo de DSD. Tales perspectivas son descritas a continuación:

1. *Especificaciones*: La primera perspectiva se muestra en la Figura 2.2. Hay un momento en que estas instancias son definidas por personas que tienen la experiencia y el conocimiento. Esta es la parte estática del modelo dado que son resultados de una planeación. En este caso, la *Unidad de Trabajo* se define en alguna expresión de trabajo la cual puede ser en proyectos, actividades, tareas u otra forma. Asimismo, la Unidad de Trabajo es realimentada por la información contenida por el Programa (objetivo general, organización y estatutos). De tal manera que el programa cuenta con un conjunto de reglas o estatutos para regir las unidades de trabajo y el objetivo general que se busca alcanzar a nivel organizacional. Además la unidad de trabajo se asigna a los miembros de la estructura organizacional según las metas para la que fue creada.



Figura 2.2. Modelo de trabajo: perspectiva especificaciones.

2. *Esfera de Trabajo*: La segunda perspectiva, considerando la definición propuesta por [46], tiene que ver directamente con las unidades de trabajo que son asignadas. Esta parte se presenta cuando cada uno de los miembros se

organiza de manera individual para identificar con quién, cuándo y cómo realizar la unidad de trabajo asignada. Esta parte se compone de las instancias que se muestran en la Figura 2.3. La unidad de trabajo incluye a quién o a quienes se ha asignado dicha unidad de trabajo y la fecha límite para entregar.

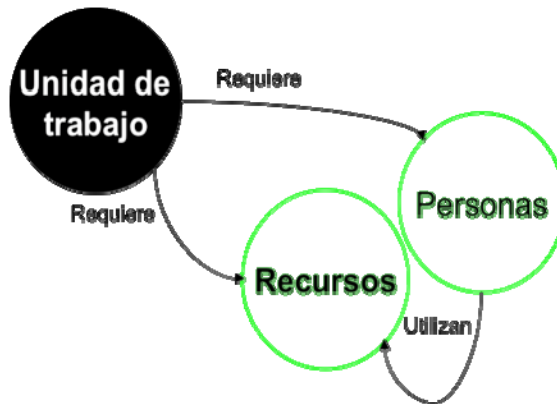


Figura 2.3. Modelo de trabajo: perspectiva esfera de trabajo.

3. *Ejecución:* Es la parte dinámica del modelo, dado que se presentan en el momento y el lugar de trabajo donde se realizan las unidades de trabajo. Esta perspectiva del modelo se compone de las instancias de color naranja y la de color negro (ver Figura 2.4). La unidad de trabajo recibe información referente a su estado de avance, la cual es generada durante la operación de las unidades de trabajo. Esta información es resultado del progreso en una unidad de trabajo.

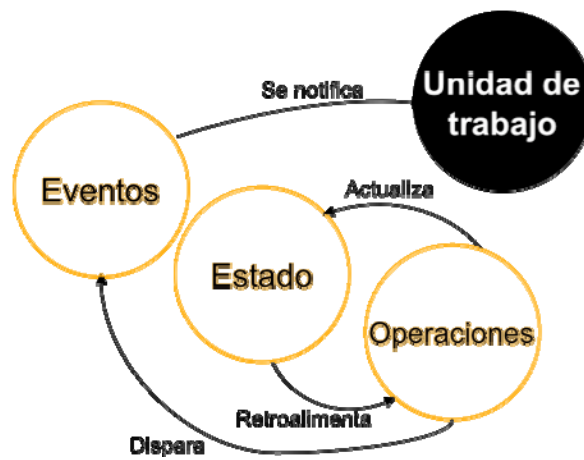


Figura 2.4. Modelo de trabajo: perspectiva ejecución.

El modelo de trabajo para DSD (ver Figura 2.1) aportan elementos importantes para el entendimiento, definición y relaciones que existen entre instancias que están involucradas en el trabajo del DSD. Pero es importante que este modelo sea complementado con la perspectiva de la teoría de articulación de trabajo propuesta por [43], 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, los cuales son:

- *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 [36, 37] 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.
- *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. En caso de contingencias, las personas improvisan para cumplir con sus tareas asignadas [7, 46, 47].
- *Nivel 3.* Se considera el agrupamiento y secuencia de tareas. Este nivel lo relacionamos con la parte de *Esfera de trabajo*, dado que las personas administran su trabajo de manera individual [48] para lograr cumplir con el objetivo del programa.

Con el modelo de trabajo descrito en esta sección, se puede tener una visión teórica de los elementos y sus relaciones que fueron reportados como parte de la organización y

división de las unidades de trabajo. En la siguiente sección se muestra la parte operativa del modelo mediante la teoría de articulación de trabajo.

2.3 Articulación de trabajo y su proceso

Para lograr un mejor entendimiento de la parte teórica previamente presentada, en esta sección introducimos algunos conceptos de teoría de articulación de trabajo y su proceso organizacional [43, 49], con los cuales se pretende entender, de mejor manera, lo que implica conjuntar los esfuerzos de cada uno de los miembros del equipo de trabajo.

La teoría de articulación de trabajo [43] involucra la organización y la relación que existe entre las tareas con las personas que las realizan. Esto implica:

- La integración de las múltiples tareas, agrupado de tareas y segmentos del total de las tareas (arco de trabajo).
- La integración de esfuerzos de varias unidades de trabajadores (individuos, departamentos, etc.).
- La integración de actores con sus varios tipos de trabajo y tareas implicadas.

En la teoría de articulación de trabajo se define una trayectoria como la multitud de tareas secuenciales y simultáneas que deben ser ejecutadas (proponer un modelo nuevo de computadora, construir un edificio, atender la enfermedad de un paciente, etc.). La trayectoria es lo que se conoce como proyecto. Algunos conceptos que son relacionados con la trayectoria y sus similitudes con los conceptos de desarrollo de software se describen en la Tabla 2.1.

Como se puede notar, los conceptos manejados en la teoría de articulación de trabajo y desarrollo de software son similares, lo cual marca la pauta para afirmar que durante el proceso de desarrollo de software (ejecución u operación de las unidades de trabajo), cada miembro tiene la responsabilidad (formal o informal) para acoplar algún aspecto del trabajo. Asimismo el acoplamiento de las unidades de trabajo (articulación) se puede realizar en cualquier lugar o sitio donde algún aspecto de trabajo se esté realizando.

Tabla 2.1. Conceptos de articulación de trabajo y desarrollo de software.

Concepto	Articulación de trabajo	Desarrollo de software
<i>Arco o línea de trabajo</i>	Consiste de la totalidad de tareas ordenadas de manera secuencial y simultánea a lo largo del curso de la trayectoria.	Se interpreta como la totalidad de unidades de trabajo que se deben realizar para satisfacer las expectativas del cliente.
<i>Actor</i>	Un actor puede ser una unidad de cualquier tamaño (una persona, equipo, departamento, subdivisión, división, organización, coalición de organizaciones). Los actores pueden variar en el número de atributos (experiencia, habilidad, conocimiento, entrenamiento, ocupación entre otros). Los trabajadores o unidades de trabajadores se hacen responsables para completar aquellas tareas acordadas mediante la respuesta a ciertos criterios contestando las siguientes preguntas: Cuándo, cómo, qué tan pronto y qué nivel de calidad.	Son los mismos desarrolladores, los cuales, al realizar su actividad trabajan en un rol determinado (programador, arquitecto de software, etc.).
<i>Asignación de tareas</i>	Las tareas pueden ser impuestas; pueden ser solicitadas; y además pueden ser asumidas sin haber sido solicitadas o impuestas.	La asignación de unidades de trabajo generalmente se realiza en función al rol que desempeña cada trabajador en la organización.
<i>Derechos</i>	Los actores pueden demandar, imponer, asumir, manipular, argumentar y negociar sobre varios tipos de trabajo y porciones de los arcos.	En el desarrollo de software la responsabilidad de los trabajadores va adherida a un rol de trabajo. Dicho rol representa el perfil y el tipo de funciones que la persona ejecuta dentro de la organización.
<i>Trayectoria</i>	Una trayectoria es un grupo de actividades o tareas de trabajo relacionadas, organizadas bajo la dirección de un responsable y tendientes a alcanzar ciertos objetivos o metas cuando se llevan a cabo.	La trayectoria es el equivalente a un proyecto, ya que se lleva a cabo en una serie de unidades de trabajo interdependientes, esto es, unidades de trabajo no repetitivas que han de cumplirse en determinada secuencia a fin de alcanzar el objetivo del proyecto.

Durante el proceso de trabajo es necesario considerar los siguientes aspectos:




- *Garantizar el flujo de los recursos.* Durante la realización de las unidades de trabajo es necesario contar con los materiales, equipos, personas y elementos que sean necesarios para realizar el trabajo.
- *Tener arreglado la división de las labores.* Es importante que los trabajadores conozcan anticipadamente lo que tienen que realizar.
- *Seleccionar motivadores para los trabajadores.* Los líderes de la organización deben considerar algunos tipos de compensaciones para que los trabajadores realicen su trabajo con la mejor actitud posible.
- *Supervisar responsabilidades asignadas o delegadas para la realización de la unidad de trabajo.* En este caso es necesario que los líderes o administradores de proyectos cuenten con mecanismos de control que les permitan tener certeza de los avances de las unidades de trabajo que realizan sus colaboradores.

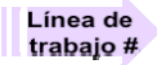



2.3.1 Modelado del proceso de trabajo en gráfica rica

Para representar el modelo de trabajo propuesto (ver Figura 2.1) con la teoría de articulación y el proceso de trabajo descrito previamente, se utilizó la gráfica rica ya que entre otras cosas nos permite observar un proceso de manera general e identificar cada una de las partes que intervienen [50]. Con este gráfico, se puede ilustrar la complejidad que conlleva sumar los esfuerzos de cada uno de los miembros del grupo de trabajo para lograr completar las unidades de trabajo que les fueron asignadas.

Para entender de manera correcta la gráfica rica, en la Tabla 2.2 se definen cada uno de los elementos que se utilizan en el modelado.

Tabla 2.2. Notación del modelado.

Icono	Significado
	Se refieren a los miembros del equipo de trabajo.
	Se refiere a las Tareas Articulas Individuales (TAI), que son organizadas de manera personal.
	Se refiere a las Unidades de Trabajo asignadas de un proyecto o trayectoria.

	Se refiere al conjunto de esfuerzos realizados hacia un proyecto en particular.
	Se refiere al proyecto o trayectoria que contiene las unidades de trabajo asignadas a los miembros o colaboradores.
	Indica el esfuerzo realizado de una Tarea Individual. Las intersecciones entre las líneas, representan esfuerzos de trabajo colaborativas.
	Resalta el conjunto de esfuerzos colaborativos realizados de la línea de trabajo. Podemos entenderlo también como Trabajo Articulado en Grupo.

Como se muestra en la Figura 2.5, el proceso de articulación de trabajo en ambientes de DSD implica de inicio, que cada uno de los colaboradores administre de manera individual el orden en que realizarán sus actividades según el criterio más conveniente (por ejemplo, fecha de entrega, nivel de urgencia, etc.). A este proceso individual se le denominó Trabajo Articulado Individual (TAI).

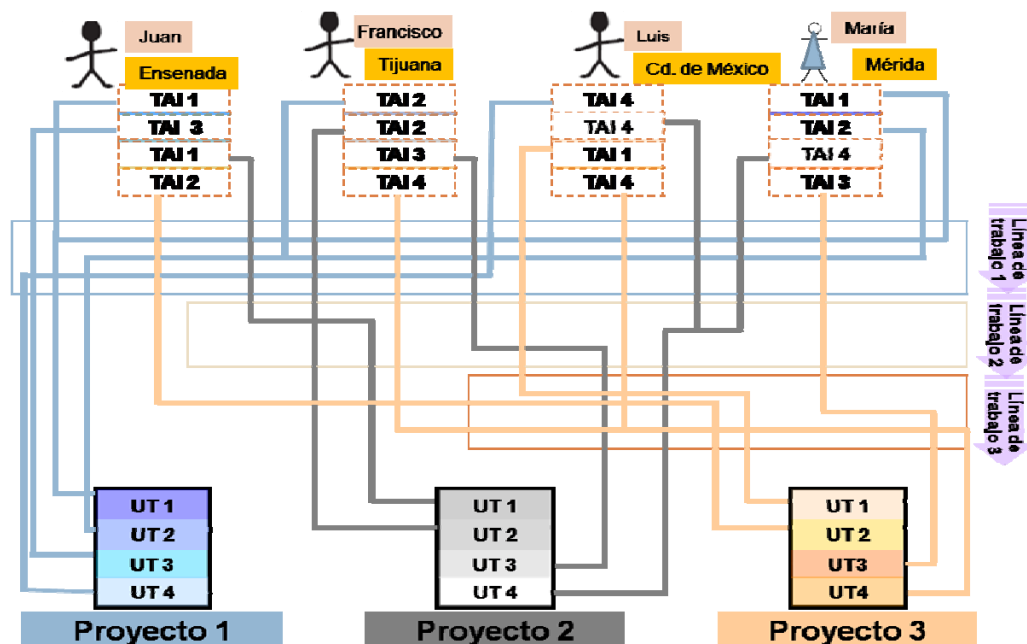


Figura 2.5. Proceso de articulación de unidades de trabajo en DSD.

Durante la jornada laboral se van enfocando los esfuerzos para ir avanzando en la ejecución de las Unidades de Trabajo (UT) que les fueron asignadas. En algunos casos las TAI requerirán acoplarlas con alguna TAI de otro compañero de trabajo. Esta suma de esfuerzos entre los miembros del grupo de trabajo se ha denominado Trabajo Articulado en

Grupo. Además se puede notar como los esfuerzos de los miembros del equipo de trabajo están enfocados a un fin común (Proyecto), los cuales son representados como líneas de trabajo.

Cabe mencionar, que la Figura 2.5 resulta útil para entender cómo los trabajadores del DSD pueden estar involucrados en unidades de trabajo de proyectos o trayectorias distintas. Dichas unidades de trabajo, eventualmente, para poder ser completadas necesitarán de la cooperación o el aporte de algún compañero, donde necesariamente requerirán iniciar colaboración. Por ello, entre más personas y unidades de trabajo estén involucradas en el desarrollo de software el número de entradas en colaboración entre ellos será mayor. Aunado a esto, la distancia física entre los miembros del equipo de trabajo conducirá a que el ambiente de desarrollo se vuelva más complejo.

2.3.2 Modelado de actividades

Después de haber entendido el proceso de articulación de trabajo mediante la gráfica rica (ver Figura 2.5), en este apartado se presenta un diagrama de actividades del mismo. Esto se realiza con la finalidad de tener una visualización con más detalle de las actividades que intervienen en dicho proceso.

Como se puede ver en la Figura 2.6, el diagrama de actividades destaca 2 tipos de actividad: Actividades en *colaboración e individuales*.

El diagrama de actividades parte de un conjunto de actividades colaborativas, que tienen que ver con la definición de las UT que son representados en conjunto por una trayectoria o proyecto. Dichas UTs son ordenadas (arco de trabajo) y divididas entre los trabajadores asignados.

Posteriormente, cada uno de los colaboradores de manera individual organiza sus actividades asignadas acorde a un criterio de prioridad (fecha de entrega, nivel de urgencia, etc.). Asimismo identifican cuales unidades de trabajo necesitan ser realizadas con la cooperación o el aporte de otros colaboradores, por lo que se toman acuerdos en conjunto para organizar de manera grupal como unirán cada uno de los aportes. Cuando se logran este tipo de actividades se define la línea de trabajo a seguir.

A partir de que ya hay una línea de trabajo definida, inicia el proceso de ejecución de las actividades (representado con las cajas resaltadas en azul: línea de trabajo, ejecución

de línea de trabajo por tarea, realizar trabajo individual, articular trabajo colaborativo, y actualiza estatus de tarea en arco de trabajo). Generalmente se inicia realizando esfuerzos de manera individual y cuando la actividad requiere cooperación de otros colaboradores, se realiza la interacción para articular el trabajo de grupo. Independientemente si las unidades de trabajo son individuales o colaborativas, se debe actualizar el estatus de avance hasta que son concluidas.

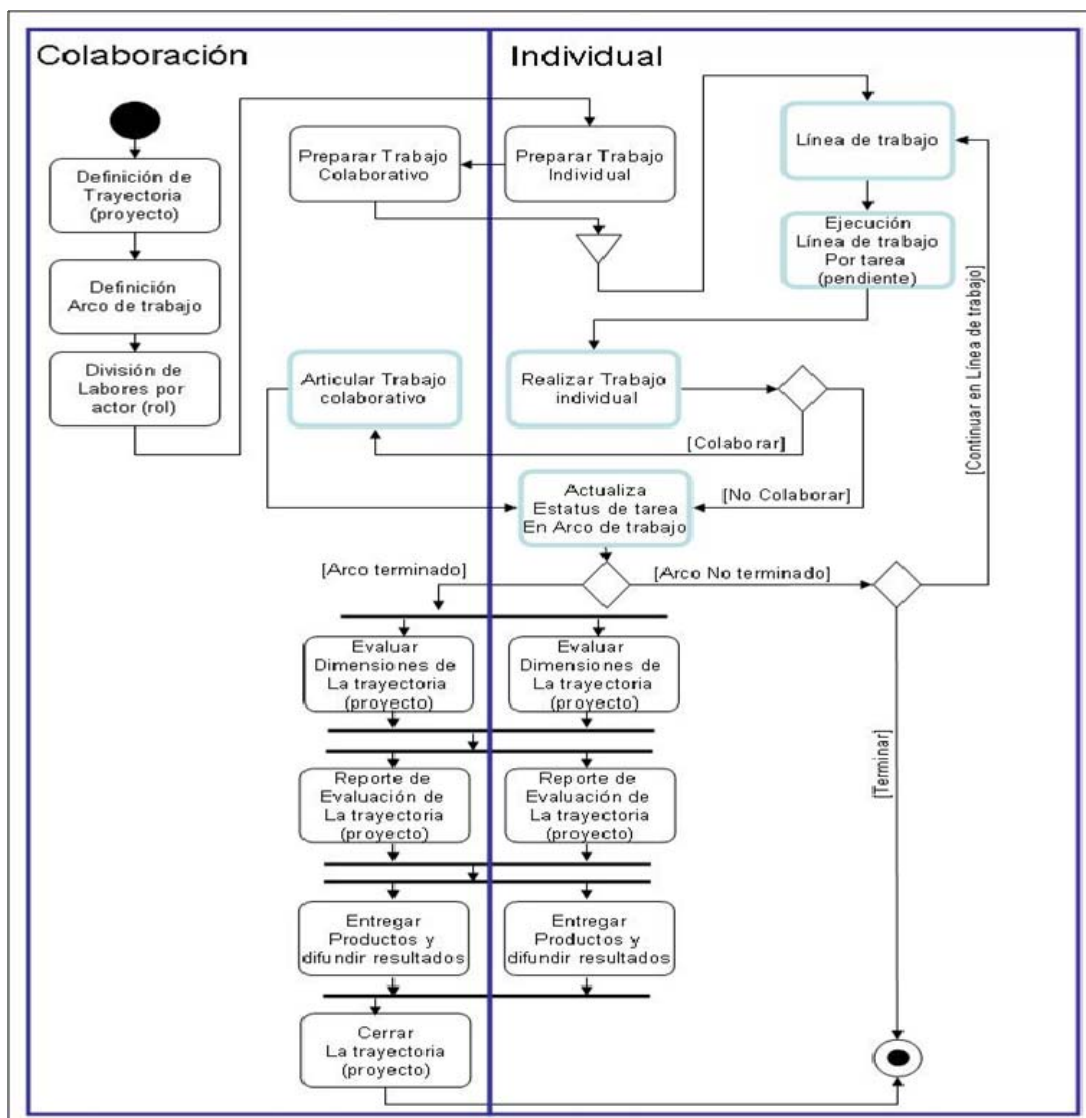


Figura 2.6. Diagrama de actividades del proceso de articulación de unidades de trabajo en DSD.

Finalmente, cuando el arco de trabajo es concluido, se realizan las actividades de cierre del proyecto. Este proceso tiene que ver con la evaluación en las distintas dimensiones del proyecto (tiempo, costo, etc.). Dicha evaluación tiene que ser documentada a través de reportes para posteriormente entregar los productos al cliente y difundir los resultados. Con estas actividades ya logradas se realiza el cierre formal del proyecto.

El diagrama de actividades del proceso de trabajo, ayuda a modelar las operaciones que son reportadas en la literatura. Esta representación fue de utilidad para identificar los flujos de trabajo paso a paso del proceso de articulación en el ambiente de DSD. Con este análisis, resulta evidente que el DSD es una actividad intelectual compleja, que a pesar de que el personal está distribuido en distintas ubicaciones geográficas, el proceso de desarrollo demanda interacciones constantes entre ellos para la coordinación de sus actividades.

Por ello, la constante necesidad por interactuar o colaborar en DSD hace necesario que los inicios de la colaboración se lleven a cabo en momentos adecuados para los involucrados. Esto es porque se supone que las personas al estar trabajando en actividades o tareas similares, tienen más voluntad para interactuar o colaborar [19, 41].

Con esto podemos establecer que para llevar a cabo una buena coordinación en DSD se requiere de una buena comunicación entre los miembros del grupo de trabajo [25]. Asimismo una buena comunicación requiere ser iniciada en un momento apropiado tanto para el emisor como para el receptor.

En este sentido, para lograr inicios de colaboración informados o apropiados en un grupo de desarrolladores distribuidos de software, se requiere la realización de un análisis enfocado a lo que se ha aportado en la literatura en cuanto a inicios de colaboración informados y a las características de las actividades de desarrollo de software. Este análisis conducirá a modelar los procesos de trabajo e identificar requerimientos para entradas en colaboración. Además, el análisis proporcionará las bases para desarrollar un prototipo tecnológico que sea útil y fácil de usar en este ambiente de trabajo.

2.4 Resumen del capítulo

En este capítulo se ha dado una descripción de los principales conceptos teóricos involucrados con el desarrollo distribuido de software. Como parte de los resultados del análisis de la literatura se obtuvo lo siguiente:

- Se describió el desarrollo distribuido de software y sus implicaciones.
- Se describieron los distintos conceptos que intervienen en el proceso de articulación de trabajo y se relacionaron con el proceso de desarrollo de software.
- Se definió un modelo de trabajo para el desarrollo distribuido de software a través de una gráfica rica.
- Se definió el proceso de articulación de trabajo mediante un diagrama de actividades.

En el siguiente capítulo se presenta un análisis basado en un conjunto de trabajos de investigación de las áreas de manejo de interrupciones y desarrollo distribuido de software. Con éstos se identifican ideas de diseño relevantes que sirven como complemento al marco teórico (descrito en este capítulo) y a su vez para soportar el diseño de una herramienta para el inicio de colaboración informado en DSD.

Capítulo 3

*“Es preciso conocer el fin hacia el que debemos
dirigir nuestras acciones”.*

Confucio

3. Trabajos relacionados

En el capítulo anterior se describió un modelo de trabajo para desarrollo distribuido de software (DSD por sus siglas en inglés), el cual fue resultado de un análisis de literatura. Dicho análisis fue soporte para establecer el marco teórico y entender en qué consiste el trabajo del DSD. En este capítulo se presenta un análisis basado en una serie de trabajos que aportan elementos que sirven como complemento para establecer ideas de diseño para el inicio de colaboración informado en DSD.

Este capítulo está compuesto en dos secciones, con las cuales se muestra cómo los trabajos citados inciden en la presente investigación. Estas secciones son *manejo de interrupciones y desarrollo distribuido de software*.

En cuanto al manejo de interrupciones, diversos investigadores han reportado cómo los trabajadores son interrumpidos en sus labores [7, 51, 52] y cómo la complejidad de la tarea (duración y tipo) y la cantidad de interrupciones impactan de manera negativa al tratar de reanudar la tarea interrumpida (p. ej. falla de memoria prospectiva) [5, 51, 53, 54]. Por lo tanto, es necesario analizar más a fondo lo referente a interrupciones. Por ello hemos

revisado estos trabajos bajo dos perspectivas: costo de la interrupción y disminución del impacto negativo.

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.) [21, 55, 56], ya que a partir de estas perspectivas se contribuye a 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 en base a los requerimientos de: coordinación, distribución de información de conciencia y entrada en colaboración.

3.1 Interrupciones

Una interrupción puede ser considerada como un evento que requiere a un usuario reorientar su atención a otra tarea, abandonando o suspendiendo su actividad primaria o actual. Las interrupciones frecuentemente pueden afectar de manera adversa el trabajo del usuario. Por lo que es importante identificar la información contenida en una interrupción que justifique que el usuario deje de hacer lo que esté haciendo y la atienda en el momento [15, 52, 57].

3.1.1 Manejo de interrupciones

En un intento por mejorar la experiencia del usuario frente a las interrupciones, se han hecho esfuerzos por tener una forma de medir el costo de que el usuario sea interrumpido [5, 7, 51]. Los logros de estos trabajos, generalmente inciden en ideas o guías de diseño que los desarrolladores de sistemas pueden considerar para disminuir el impacto de la interrupción. Con esto se busca evitar que las interrupciones afecten de manera negativa la actividad. En entornos o ambientes controlados se ha demostrado que las interrupciones impactan de manera negativa en el tiempo para completar tareas [5, 7], en la tasa de error o impacto [52, 58], en la toma de decisiones [15, 53] y en el estado afectivo de las personas [58].

El problema ocasionado por la interrupción, que es continuamente abordado, es recordar lo que se tiene que trabajar en una tarea en particular (falla de memoria prospectiva), la cual se ve afectada de manera importante cuando aumentan las interrupciones y el número de tareas por hacer [7, 54]. Como parte de los resultados de Czerwinski et al. [7] se afirma que durante el intercambio de tareas provocado por una interrupción existen algunas variables que se derivan y deben tomarse en cuenta para entender los efectos de la interrupción. Dichas variables son la complejidad de la misma tarea, su duración, duración de la ausencia del trabajador, el número de las interrupciones y el tipo de tarea. Específicamente el tratar de retomar una tarea es considerado difícil por parte de los trabajadores [7], puesto que ellos tienen que realizar acciones explícitas para recordar lo último que hicieron en una tarea en particular y cuanto tiempo le invirtieron como lo demuestran Brush et al. en su trabajo reportado en [47]. Asimismo, Cutrell et al. [5] observaron que los trabajadores durante la ejecución de una tarea tardan mucho tiempo en atender una solicitud de interacción porque por lo general no la atienden hasta que ellos hayan finalizado su actividad primaria. También un factor importante a considerar es el que Dabbish y Kraut reportan en [15], donde argumentan que el avance y la facilidad del acceso de las tecnologías de comunicación (p. ej. correo electrónico, mensajería instantánea, dispositivos móviles) han aumentado las fuentes de interrupción. Además es importante tomar en cuenta que las interrupciones son menos perjudiciales cuando tienen relevancia con la tarea que se realiza en determinado momento, como ha sido reportado en [5], en su esfuerzo por entender cuando las interrupciones pueden ser consideradas irrelevantes.

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 como fue reportado por Ye [19], ya que se argumenta que al contactar a un colega (receptor) para tratar una tarea de interés compartido, existe más voluntad por parte del receptor para que ésta se lleve a cabo. Por lo que este argumento es relevante para diseñar los inicios de colaboración informados.

3.1.2 Diminución del impacto negativo

Debido a los altos costos que pueden provocar las interrupciones, es importante entender como mitigar el impacto de una interrupción. Algunos trabajos se han reportado en este sentido.

Una de las propuestas es mediante negociaciones para realizar la interrupción. Esto ha sido abordado bajo la visión de que las personas pueden tener cuatro respuestas frente a una interrupción: aceptarla completamente, aceptarla con algún inconveniente, negarla o abandonarla. Esta técnica fue considerada la más adecuada siempre y cuando la rapidez por realizar la interrupción sea o no sea crítica [59]. Otros trabajos han utilizado la técnica de observación con grupos de desarrolladores de software y sugieren una serie de implicaciones de diseño para mitigar los efectos negativos de la interrupción [7, 52].

Un sistema interesante que utiliza modelos de inferencia son las notificaciones de atención sensible reportadas por Horvitz et al. [60], las cuales equilibran los costos de interrupción con el contenido de la notificación. Esto lo realiza mediante la clasificación de la urgencia de los mensajes, prioridades y patrones de usuario observados durante su jornada de trabajo. En particular con este sistema se reconoce que el usuario puede tener más voluntad para atender una interrupción. Para lograr esto, se tienen que entrenar los modelos de inferencia utilizados.

También se han propuesto una serie de trabajos que buscan entender el comportamiento en el momento oportuno para la interrupción [52, 61], donde se han reportado estrategias de concordancia o coincidencia (a través de señales visuales) para tener una idea del nivel de utilidad del contenido de la interrupción. En este sentido un trabajo reciente reportado por Bailey y Iqbal [6], el cual está basado en especulaciones de teóricos cognitivos, muestran un experimento controlado de cómo las interrupciones pueden ser mitigadas siempre y cuando se realicen cuando la carga mental de las personas se encuentre en un nivel bajo. Para validar los distintos momentos de la carga de trabajo se basaron en un experimento controlado de la luz mediante el tamaño de la dilatación de la pupila. Con esto demostraron que la sobrecarga mental de las personas durante la realización de una tarea es cambiante, la cual es posible diferenciar. Con esto proponen como implicación de diseño asignar costos de interrupción a los diversos momentos de la realización de una tarea.

Los trabajos referentes a la mitigación de la interrupción, ofrecen una serie de ideas de diseño que promueven la alineación de esfuerzos para iniciar la interacción, es decir, las interrupciones pueden ser consideradas relevantes o importantes para la persona que es contactada (receptor) siempre y cuando el motivo de dicha interrupción vaya en función a la tarea que realiza en ese momento. Es por ello, que los esfuerzos por proveer inicios de colaboración informados pueden estar en función de encontrar mecanismos o servicios que promuevan el suministro de información de la actividad del contacto potencial al iniciador (emisor). Esto es 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.

3.2 Desarrollo distribuido de software y el papel de la comunicación.

Uno de los aspectos críticos clave para el éxito de los proyectos en DSD, es la comunicación entre los miembros del equipo de trabajo. Esto es debido a la dificultad para entablar comunicación de manera frecuente entre los trabajadores, por lo que éstos deben realizar esfuerzos adicionales para lograr que la comunicación se haga de manera efectiva [3, 4]. Por ello en el siguiente apartado se describen trabajos de investigación que han realizado esfuerzos para que la comunicación en DSD sea más adecuada.

3.2.1 Mejorando la comunicación inadecuada en DSD.

Diversos investigadores han realizado diferentes propuestas, una de ellas es la de “contexto de proyectos compartidos” [62], donde se busca mantener un entendimiento actualizado de las modificaciones realizadas al diseño de un sistema por parte de equipos de trabajo distribuidos. Con esto se argumenta que es posible disminuir la necesidad de comunicación entre grupos distribuidos. Por lo que, en este trabajo se hace énfasis en que el arquitecto de software debe de tener una visión de la creación de artefactos arquitectónicos, entender las dependencias entre los elementos de la arquitectura y la asignación de su desarrollo a los equipos distribuidos. Esto les permite tener una visión global de todo el proyecto y pueden asignar actividades lo menos dependientes entre grupos de trabajo

distribuido. Sin embargo, destacan que para asegurar el éxito del proyecto se debe facilitar y gestionar tareas o actividades interrelacionadas.

Otro tipo de propuesta son las redes de comunicación como las que analizan Cataldo y Herbsleb [63], donde indican que es necesaria una evolución en las formas de hacer frente a las estrategias de asignación y realizar actividades bajo enfoques de modularidad. El hallazgo principal de este análisis fue que los desarrolladores clave deben desempeñar un rol importante para facilitar la comunicación entre los sitios distribuidos, pueden representar la adaptación efectiva de la comunicación o pueden significar un cuello de botella importante. Por esta razón se sugiere la definición de un rol emergente en DSD que funja como “enlace” entre los sitios que comparten o coordinan actividades o tareas comunes.

También, se han propuesto estrategias basadas en experiencias como las reportadas por Repenning et al. [64] mediante la utilización de una arquitectura de componentes para apoyar la distribución entre sitios. Esto hace posible que cada sitio cuente con sus propios componentes y trabajen de manera independiente, disminuyendo la necesidad por tener comunicación y coordinación entre sitios distantes.

Por otro lado, actualmente existen tecnologías que soportan la comunicación remota asíncrona (correo electrónico, wikis, etc.) y síncrona (mensajería instantánea, videoconferencias, etc.), las cuales son muy recurridas por los grupos de trabajo distribuidos. Este tipo de herramientas son frecuentemente encontradas en los espacios de trabajo de DSD, sin embargo tienen como principal inconveniente el escaso entendimiento de lo que un colega está realizando antes o al momento de ser abordado. Esto en buena manera es debido a que las características actuales de estas herramientas favorecen al iniciador de la interacción, pero no necesariamente al receptor. Es decir no ofrecen información pertinente de la actividad actual del receptor potencial para reconocer lo que se está o estuvo realizando con las tareas compartidas, para que de esta forma el iniciador pueda inferir la conveniencia o no de interrumpirlo en un momento determinado.

Estas propuestas enfocadas a mejorar o tratar los problemas relacionados con la comunicación entre el grupo de trabajo, por un lado tratan de evitar dependencias entre desarrolladores distribuidos, de tal manera que disminuyan la necesidad para colaborar y

por otro lado se fomenta el uso de herramientas de comunicación. No obstante realizar actividades de coordinación en DSD es inevitable [2]. Por lo que desde el punto de vista de los inicios de colaboración informados el uso de herramientas de comunicación síncronas es importante. Pero se requiere que éstas posean elementos de información que notifiquen al iniciador de la colaboración que actividad está realizando en determinado momento el contacto potencial. Por ello, en el siguiente apartado se presentan contribuciones relacionadas con este tipo de elementos que promueven el conocimiento de la actividad de los miembros del grupo de trabajo.

3.2.2 Distribución de información de conciencia

La conciencia según Dourish y Bellotti [65] es definida como la comprensión de las actividades de otros, lo que proporciona un contexto para su propia actividad. Este contexto se utiliza para asegurar que las contribuciones individuales son relevantes para la actividad del grupo de trabajo, y evaluar las acciones individuales en relación con los objetivos y el progreso del grupo. En este sentido se han propuesto un conjunto de características para soportar conciencia [66], las cuales demandan en general información acerca del estatus del proyecto y de la administración de cambios. Esta información típicamente es propagada a través de canales de comunicación informales. En este trabajo también se argumenta que muchos errores de los sistemas son aún atribuibles a malentendidos, falta de comunicación y la mala gestión de requerimientos para el sistema. Esto es debido en gran parte a la escasa comunicación que existe en el ambiente GSD, aún cuando se ha reportado que más de la mitad de las actividades son de naturaleza colaborativa [56].

También se ha propuesto la utilización de grupos de conciencia para crear oportunidades de comunicación informal. Pero resaltan que primero se deben atender retos que incluyen la identificación de incentivos para que los desarrolladores acepten la monitorización de las acciones del grupo y lo concerniente a privacidad [67].

Algunas herramientas que han sido propuestas para distribuir información de conciencia entre los miembros de un grupo para reportar alguna actividad son:

Palantír [68] está diseñado para proporcionar información de conciencia de espacio de trabajo como complemento de los sistemas de gestión de configuración existentes. De

manera particular, esta herramienta informa al desarrollador cual de sus colegas cambia ciertos artefactos, calcula una medida de severidad simple relacionado a los cambios y despliega gráficamente la información de una manera configurable y no-invasiva.

ProjectWatcher [69] es un sistema prototipo que recopila información acerca de los artefactos del proyecto y las acciones del desarrollador sobre los artefactos, y además visualiza esta información de conciencia en el ambiente de desarrollo Eclipse.

StickySpots [70] es un sistema de mensajería basado en localización que mantiene informado a los miembros de un grupo mediante mensajes digitales cortos a varias ubicaciones del edificio.

MRAC [71] es una herramienta basada en video que introduce el concepto de “virtually mobile Mixed Reality Architectural Cells” enlazando permanentemente un espacio físico con uno virtual. Este soporta manejo de conciencia, interacción social y privacidad para equipos de trabajo pequeños y medianos.

Community Bar [72] es un groupware diseñado para soportar interacciones casuales dentro de un grupo pequeño distribuido. A través del uso de video (con webcam) se obtiene información del contacto, así como con pequeñas notas y funciones para recrear información del contexto que está disponible cuando la gente trabaja en situaciones colocalizadas (p. ej. visualizar mediante el video el código que un colega está modificando).

FASTDash [73], es una herramienta amigable que utiliza una representación del código fuente compartido, el cual resalta las actividades actuales de los miembros del equipo. Esta herramienta permite a un desarrollador determinar de manera rápida que miembros del equipo han liberado archivos fuentes, cuales están siendo vistos y que métodos y clases se están modificando. Con esto se obtiene conocimiento inmediato de posibles situaciones que pueden ser conflictivas (p. ej. editar el mismo código fuente).

IMPROMPTU (IMPROving MDE’s Potential to support Tasks that are genUine) es un marco de trabajo reportado en [74], que permite a los usuarios compartir información de las tareas mediante despliegues de ambiente. Con esto, se permite la interacción conjunta con información centrada en el problema a resolver, la cual es compartida en los despliegues para discusión y reflexión. A través de la interfaz de usuario se ofrece una

representación visual de los miembros del grupo de trabajo, disponibilidad de despliegues y aplicaciones que han sido puestas a disposición del grupo.

YooHoo [75], es un sistema para mantener informados a los desarrolladores con respecto a los cambios en los códigos fuentes. Esto es realizado mediante notificaciones. Las notificaciones no están instantáneamente disponibles al desarrollador, sino que es el ambiente de desarrollo el que valora los cambios relevantes en intervalos específicos de tiempo. Este intervalo puede configurarse por proyecto, lo cual permite la personalización de notificaciones a los usuarios.

TOTA (Tuples On The Air), es un middleware con soporte para actividades conscientes de contexto adaptativo [76]. La idea principal de TOTA es utilizar conjuntos de datos o tuplas distribuidas espacialmente, las cuales son propagadas a través de la red con base en reglas específicas de la aplicación, tanto para la representación de la información contextual como para el soporte del desacoplado de interacciones entre los componentes de la aplicación. TOTA promueve un modelo de programación simple y puede facilitar el acceso a información distribuida, la navegación en redes complejas y el cumplimiento de tareas de coordinación compleja.

Como podemos notar, la distribución de información de conciencia entre grupos de trabajo, da a conocer el mismo estatus al resto del grupo de manera similar a como lo realizan los mensajeros instantáneos tradicionales. Por ejemplo: si el estado del usuario es mostrado como “ocupado”, éste estado se distribuye a todos los miembros del grupo por igual. Por el contrario se han reportado algunos riesgos al utilizar mecanismos tradicionales para coordinar actividades de manera remota (p. ej. reuniones vía audio-conferencia, videoconferencias, chat, entre otros), donde la escasez de diversos tipos de conocimiento del trabajo de equipo pueden influir en la eficacia de estos mecanismos y en la medida en que realmente necesitan ser utilizados. En este sentido Espinosa et al. [77] han propuesto dos tipos de conciencia que son importantes para el GSD: “Conocimiento de la tarea” y “Conocimiento de la presencia”. Es decir, se requiere tener información actualizada de que se está haciendo sobre una tarea, que miembros del equipo están participando, dónde se encuentran y cuando lo estuvieron o están haciendo.

Como se ha notado, los trabajos previamente presentados están centrados a aspectos de coordinación de actividades, ya que la información de conciencia es requerido de manera frecuente para coordinar actividades en grupo [65]. Por ello en el siguiente apartado se presentan trabajos más orientados a esta temática.

3.2.3 Coordinación

La modularidad de tareas es un tema que ha sido analizado para abordar problemas enfocados a la coordinación de actividades o tareas. Esto es porque la modularidad reduce la necesidad de comunicar y coordinar actividades en organizaciones de desarrollo, puesto que reduce cada una de las partes del sistema de tal forma que logren ser independientes. Por lo que su aplicación ha sido propuesta para promover una comunicación mínima entre equipos distribuidos para resolver actividades interdependientes. Por esta razón, este tipo de propuestas han sido reportados con altas probabilidades de fallar durante la integración de las actividades o tareas [78]. Por ello es importante poner atención especial en las interdependencias existentes entre actividades o tareas como discuten Ovaska et al. en [79], donde establecen que cada componente puede ser desarrollado de manera separada, ya que de esta manera no sería necesario tener en cuenta el desarrollo de otros componentes o las diferencias existentes entre los sitios como la distancia, la cultura y el lenguaje. Por lo que adoptar una arquitectura como herramienta de coordinación tiende a guiar a los desarrolladores hacia una toma de decisiones compatible, asumiendo que hay comunicación efectiva y un entendimiento común de lo que representa la arquitectura.

También, es importante planear los posibles problemas de coordinación para poder resolverlos a tiempo. Algunas formas de enfrentarlas han sido por ejemplo, contar con personal dedicado a resolver problemas, utilizar anuncios públicos y contar con listas de correo electrónico o buzones de correo dedicados como lo reportan Paasivaara y Lassenius, [80]. También ellos sugieren que se pongan en práctica las entregas frecuentes para disminuir la incertidumbre.

Asimismo, se han hecho esfuerzos por tratar de explicar y comprender la forma en que se presentan las problemáticas de coordinación. Esto lo han abordado con un enfoque teórico referente a organización. Por ejemplo, Cataldo et al. [25] presentan cuatro estudios de caso donde identifican problemas de coordinación en el GSD, bajo el enfoque de

modularidad de tareas. Otra forma de abordarlo es mediante la creación de un marco de trabajo referente al análisis de coordinación en el GSD como el presentado por Wiredu [13]. Este marco de trabajo lleva al entendimiento de que en el GSD la administración del proceso de interdependencias, conflictos inter-unidades e inter-personales, información incierta y ambigüedades, y representaciones de tecnología serán significativamente condicionadas por la distancia, la tecnología y los aspectos socioculturales. Otro análisis en este sentido, fue realizado por Tiwana [81], en el cual se conceptualizan seis clases de herramientas para coordinación del desarrollo, las cuales son: administrador de requerimientos, modelador de arquitectura, herramienta para automatización de pruebas, herramienta para desarrollo de pruebas “case”, administrador de la configuración y rastreador de solicitudes de cambios y defectos. La idea de diseño más importante de este análisis, es que las organizaciones de software primero deben examinar cuidadosamente las características de cada proyecto y luego deben seleccionar el conjunto de herramientas de coordinación que tiendan a tener el mayor impacto en su proceso de desarrollo. Por otro lado, también se han realizado estudios cualitativos [12], en los cuales se pone especial atención a la distancia, la solicitud de modificación y los comportamientos de poca disposición por colaborar entre equipos distribuidos.

Existen muchas herramientas que han sido reportadas para auxiliar la coordinación. Para este trabajo hemos recopilado las propuestas que nos aportan elementos para los inicios de colaboración en DSD. Para esto se han separado en cinco partes basados en las siguientes actividades: localización de expertos, codificación, documentación, modelado y pruebas.

Parte 1. Referente a *localización de expertos*, se propuso a Expertise Browser [82], el cual es una herramienta que busca resolver el problema de búsqueda de expertos. Para realizarlo utiliza los datos del sistema de administración de cambios para localizar a la gente con el nivel de expertez deseado. Para lograrlo la herramienta cuantifica la experiencia y presenta evidencia para validar esta cuantificación como una medida del grado de expertez. De manera fácil es posible encontrar personas que trabajaron en determinadas partes de un proyecto, y se puede visualizar cómo su experiencia se distribuye sobre el producto (módulo o subsistema). Otra herramienta que ha sido propuesta es

Step_IN (socio-technical platform for in-situ networking), la cual soporta colaboración de conocimiento situado (situated knowledge collaboration) [19]. Este sistema auxilia a un desarrollador de software a encontrar a colaboradores que primero tengan el nivel de conocimientos (experto) sobre la información de interés y segundo que tengan cierta voluntad de ofrecer ayuda de manera oportuna.

Parte 2. En cuanto a *codificación*, la herramienta conocida como CHIME (Columbia Hypermedia IMmersion Environment) reportada en [83], es una aplicación basada en internet e intranet, la cual permite a los usuarios ser localizados a través de mundos virtuales en tercera dimensión (3D) representados por un sistema de software. Los usuarios interactúan con los artefactos del proyecto mediante la exploración (“walking around”) del mundo virtual, donde los usuarios potencialmente pueden encontrar y colaborar con otros usuarios representados por sus respectivos avatares. Este trabajo fue orientado hacia el control de código fuente. Por otro lado, Augur [84] es un sistema de visualización, en el cual las propiedades del sistema de software son asignadas a un color y otras características para el despliegue gráfico del código fuente mismo. Con este sistema los desarrolladores o administradores visualizan cuales áreas del sistema están activas, de tal manera que ofrece una forma de ubicar los módulos y desarrolladores con un marco de referencia que describe su relación. CodeSaw [85] presenta una visualización de tipo social de manera fácil y comprensible para las contribuciones de códigos fuentes (líneas de código agregadas). CollabDev [86] es una herramienta que soporta un ambiente de mantenimiento distribuido. Esta herramienta extrae datos de los códigos fuentes y cambia los datos del sistema de administración de versiones. El metadato es indexado y almacenado en el repositorio. CollabDev cuenta con un motor de búsqueda, análisis de modularidad y diagnóstico de componentes, visualización de códigos y uso de componentes. La herramienta integra varios de sus componentes mediante protocolos de internet asíncronos basados en Wiki. Recientemente fue reportado Tesseract [87], el cual es un ambiente exploratorio interactivo. En este ambiente los desarrolladores pueden investigar las relaciones del código fuente, los errores o fallas, el correo electrónico y las discusiones en torno al error. Además es posible conocer las contribuciones de cada desarrollador.

Parte 3. Referente a *documentación*, una de las propuestas es XSDoc Wiki [88], la cual es una propuesta de comunicación asíncrona basada en wiki. Mediante la utilización de SSDoc Wiki se puede obtener de manera rápida el conjunto de artefactos de software heterogéneo de múltiples fuentes, en concreto, de los archivos de código fuente, los modelos, y los documentos. Esta propuesta permite asegurar que los contenidos están siempre a la última fecha. XSDoc Wiki ayuda a estar al pendiente de la dirección del desarrollo, el cual es informado por la misma documentación. Galaxy Wiki [89] es un ambiente de desarrollo basado en Wiki y Java. En este ambiente el usuario puede convenientemente iniciar o importar un proyecto java, crear o modificar una clase y ejecutar o depurar un programa. Galaxy Wiki está basado en el concepto wiki, con lo cual permite la existencia de un sistema de autoría colaborativa para propósitos de documentación y coordinación, de tal manera que permite a los desarrolladores compilar, ejecutar y depurar programas en las mismas páginas del Wiki.

Parte 4. En lo concerniente al *modelado*, el sistema D-UML (Distributed UML) es una implementación JAVA de una instancia del marco de trabajo (framework) D-Meeting [90]. D-UML es consciente de la colaboración, es decir varios usuarios pueden estar interactuando simultáneamente con la aplicación. Cada usuario que se une a la sesión del modelado puede realizar acciones, por lo que los participantes de la sesión conocen qué y quién está realizando alguna acción sobre el modelado. Un sistema similar es XCHIPS (Cooperative Hypermedia Integrated with Process Support) descrito en [91], el cual soporta modelado cooperativo, permite la representación y simula la funcionalidad. La característica principal de este sistema es que todos los cambios realizados a un modelo del proceso pueden ser realizados de manera colaborativa. Es decir, los usuarios pueden en cualquier momento acceder al modelo del proceso de manera concurrente. SYSIPHUS [26] es un ambiente distribuido que ofrece un marco de trabajo uniforme para los modelos de sistemas, artefactos de colaboración y los modelos de organización, con servicios para explorar, buscar, filtrar y analizar los modelos.

Parte 5. Finalmente, lo que respecta a la parte de *pruebas*, el sistema ADAMS (ADvanced Artefact Management System) reportado en [92] soporta 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. Otro trabajo reportado en este sentido es la infraestructura SoftFab [93], la cual habilita proyectos para automatizar la construcción y el proceso de pruebas . Esta infraestructura permite controlar los proyectos de manera remota. Las características de SoftFab que permiten la mejora del DSD son:

- Despliegue de actividades actuales y otras ubicaciones
- Acceso a los mismos reportes y bitácoras
- Acceso a los productos y recursos
- Centro de control para coordinar tareas con recursos distribuidos
- Una descripción general de los procedimientos de construcción y pruebas.
- Conocimiento a detalle de una tarea.

Sin duda la coordinación de actividades en DSD ha sido un tema muy recurrente por los investigadores. Por lo que se argumenta que coordinar actividades en situaciones de trabajo distribuidas es inevitable el uso de herramientas de comunicación. Por lo que la forma en que se inicia dicha comunicación es relevante para el tema de inicios de colaboración informados, independientemente del motivo de la interacción (p. ej. localizar a un experto). Por lo que se hace necesario contar con características de despliegue gráficos (p. ej. mediante el uso de colores) que faciliten interpretar la actividad de manera rápida como lo utiliza CodeSaw para sus propósitos de codificación. Ya que con este tipo de despliegues se podría informar el nivel de voluntad para ofrecer ayuda de los contactos en tiempo real. También es importante conocer la identidad de los contactos y las actividades que tienen asignadas de manera compartida con el iniciador (autoría colaborativa). De esta manera el iniciador podrá contar con elementos de información para inferir la actividad que realiza su colega o contacto potencial. Con esto conocerá quién es el colega con la que se comparte la actividad y qué acción u operación está realizando en determinado momento. Finalmente se considera que informar la ubicación geográfica actual de los colegas es relevante para el DSD, puesto que se pueden evitar problemas derivados de la diferencia horaria, con los cuáles pudieran causar molestias por iniciar el trabajo colaborativo e intenso en horarios no adecuados para los colegas (p. ej. iniciar la colaboración en horario de comida para un colega).

Como se puede notar, los problemas de coordinación son abordados frecuentemente para disminuir la incertidumbre en el costo del proyecto. Por lo que es importante tener en cuenta tales costos con el fin de evitarlos lo menos posible en el diseño de ideas para iniciar colaboración. En la siguiente sub-sección se ofrece más detalle de esto.

3.2.4 Costos vulnerables

Por supuesto que hay varias razones que motivan el DSD. Espinosa y Carmel [94] argumentan que los productos de software al ser digitales, sus costos de transportación son muy bajos y la entrega a tiempo es relativamente cero. Además los costos de producción en muchas ubicaciones distantes son bajos. Carmel y Agarwal [28] establecen que la dispersión geográfica habilita a las compañías al acceso de talento de software especializado y recursos técnicos.

Por el contrario, aunque hay una tendencia de negocios para el desarrollo de software “offshore” en los países de menores costos laborales, algunos de estos ahorros se ven contrarrestados por costos incrementados por una mala planeación o coordinación como es discutido por Keil et al. [95]. Por ello se sugiere que durante la recopilación de información para la propuesta de un proyecto de DSD, deben considerarse factores de producto (adecuación de arquitectura), personales (culturales, nivel de habilidad, entendimiento compartido) y de proyecto (modelo de colaboración, distancia física, herramientas e infraestructura).

Las compañías de desarrollo de software necesitan tener en cuenta que existen perdidas inevitables en cuanto a calidad y productividad cuando se decide mover o extender la producción de software a otra ubicación geográfica con el fin de reducir el costo de mano de obra. Pero se argumenta que las empresas que practican alta calidad en sus procesos de software tienden a superar más rápido los efectos negativos de la distribución que las empresas que no lo hacen [96].

Los problemas de costos atribuibles por la distribución [94] son los siguientes:

- Producción: costos derivados por la coordinación de actividades o tareas.
- Comunicación: costos por mantener enlaces de comunicación, envíos y recepción de mensajes.

- Retardo: costos incurridos por un actor que está esperando que otro complete su actividad o tarea.
- Aclaración: costo adicional de comunicación y retardo debido a los malentendidos por una mala comunicación.
- Re-trabajo: costo adicional de producción debido a problemas de comunicación.

Independientemente del método de desarrollo de software empleado (p. ej. cascada, incremental, ágil), la coordinación es fundamental para la gestión del proceso del software. Por lo que se ha afirmado que las cuestiones del tiempo no tienen que ver con el costo del proyecto [94]. Con esto se establece que para propósitos de coordinación adecuada es fundamental que exista buena comunicación. Esto se puede notar en el énfasis especial que los investigadores han puesto en el tema de comunicación, ya que lo consideran como uno de los principales retos de investigación en el DSD [1, 17, 55, 56]. Las interacciones informales en particular es un tema que puede aportar de manera importante a la creación de relaciones de confianza entre los miembros de un grupo de trabajo remoto [2, 97]. La importancia de este tema como parte de la comunicación en DSD se trata en el siguiente apartado.

3.2.5 Mantener relaciones de confianza

Crear relaciones de confianza entre los miembros de trabajo de DSD se considera difícil, dada la distancia que existe entre los colaboradores. Se ha reportado que la comunicación cara a cara es fundamental para establecer lazos de confianza entre los colegas de trabajo. Esto es porque las personas se encuentran frecuentemente entre los pasillos (comunicación informal) o en salas de reuniones (comunicación formal) para consultar o discutir algo en particular. Por ello este tipo de interacciones causan que las personas interactúen con más familiaridad y confianza [18, 55, 97]. Dicho de otra manera, cuando hay escasez de confianza hay poca voluntad para comunicar. Por otro lado, estudios de desarrollo distribuido en entornos de software libre, discuten que los equipos de trabajo se basan más en mecanismos de control social que en los de confianza [41]. En cambio, Herbsleb et al. [98] argumentan que establecer relaciones de confianza con el transcurso del

tiempo es importante para el trabajo de DSD porque cuando existen interacciones frecuentes, es posible juzgar de manera rápida las capacidades y habilidades de un colaborador.

También Shami et al. [99] han identificado la necesidad de construir relaciones de confianza con los colaboradores que se encuentran distantes de la misma manera como se hacen en situaciones colocalizadas. Esto con el fin de evitar problemas que devalúen el trabajo que es realizado por colaboradores de una ubicación diferente como lo reporta Kiel [100]. Para esto, se argumenta que para lograr el éxito de la colaboración entre equipos globales de desarrollo, se debe invertir en el desarrollo de la socialización entre los miembros del grupo, a pesar de las restricciones impuestas por la distribución, ya que la socialización podría ser soportada con el tiempo y en distintos niveles de la organización. Por esto Oshri et al. [40] sugieren que la organización debe evaluar como están los grupos dispersos para priorizar la introducción de actividades, herramientas de comunicación y los procedimientos que conduzcan a la creación y renovación de la socialización entre miembros del grupo.

Con la existencia de confianza entre los miembros de un grupo de trabajo es posible valorar la contribución de otros (p. ej., el código aportado por diferentes miembros); es decir, el valor total del software está sujeto a las contribuciones de los desarrolladores involucrados y las decisiones tomadas con respecto a cómo las contribuciones son o fueron fusionadas [101].

Para el tema de los inicios de colaboración informados es importante considerar el tema de las relaciones de confianza en DSD. Como se ha mencionado previamente, realizar trabajo en una situación colocalizada beneficia en gran medida las relaciones de confianza. En cambio para el entorno de DSD se han hecho esfuerzos para enfrentar los retos referentes a la poca voluntad para interactuar (p.ej. [41]) y para promover comunicación frecuente en el DSD (p. ej. [98]). Para abordar estos retos se han detectado en la literatura distintas propuestas en cuanto a mecanismos y sistemas prototipos para iniciar o entrar en colaboración que tienen particularmente ciertos elementos que ayudan a informar actividad a un grupo de trabajo, con lo cual se fomenta que la herramienta de comunicación integre información útil para el trabajo en grupo. Éstas se describen en el siguiente apartado.

3.2.6 Entrar en colaboración.

Para este trabajo se considera que contar con soporte para entrar en colaboración debe ser visto desde la perspectiva que reportan Morán et al. [102]. Por lo que los requerimientos para iniciar colaboración entre miembros de un equipo distribuido deben considerar lo siguiente:

- 1) *Proximidad artificial*. Para establecer comunicación de manera remota es necesario contar con los siguientes elementos:
 - Grupo de colaboradores de trabajo adecuados
 - Co-presencia de los posibles colaboradores
 - Esfuerzo mínimo para entrar en colaboración
 - Mecanismo para determinar la presencia e identidad de los colaboradores.
- 2) *Establecer contacto con los colaboradores correctos*. Esto debe realizarse mediante un canal de comunicación apropiado.

Para esto, 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 [103]. Por lo que también se ha afirmado que es importante mantener interacciones cortas de comunicación asíncrona que pueden servir como un sustituto suficiente para la comunicación síncrona [17].

En este sentido, se han propuesto herramientas de comunicación para entrar en colaboración, las cuales intentan integrarse a las actividades de un grupo de trabajo. Por ejemplo:

TaskMaster [104] es propuesto como un rediseño de la interfaz del correo electrónico, ofreciendo una opción fácil mediante la cual se puedan controlar y visualizar las tareas que los usuarios deben atender.

Rear View Mirror (RVM) [105] es un mensajero instantáneo basado en presencia con características que ayudan a soportar equipos de trabajo. RVM ofrece la creación de

grupos. Los grupos son similares a los listados de contactos con la diferencia de que estas listas son definidas de manera independiente por cada usuario.

Hubbub [106] es un mensajero instantáneo que ofrece a los usuarios conciencia y acceso a conversaciones potenciales. Sin embargo la característica que lo distingue de otros, es que los usuarios tienen un *Sound ID* (tonos cortos) que sus contactos escuchan cada vez que se activan.

ProjectView-IM [107] sugiere modificaciones en el diseño a la interfaz de mensajería instantánea, considerando tres características: conciencia (en línea y disponible), información de usuario y recordatorios. Este diseño facilita la división de la atención entre múltiples proyectos sin perjudicar una tarea primaria.

Las herramientas aquí descritas para entrar en 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 ofrece la lista de actividades asignadas e historial de actividad; RVM ofrece presencia de usuarios y agrupados personalizados de contactos; Hubbub incluye presencia e identidad; y ProjectView-IM distribuye información de presencia, lista de actividades asignadas, identidad y notificaciones en base a recordatorios de tareas pendientes.

3.3 Hallazgos identificados

Después de haber analizado la literatura se han encontrado una serie de elementos que pueden ser útiles para un desarrollo que soporte inicio de interacciones informadas.

Para ello, podemos afirmar que los trabajos citados aportan las siguientes ideas para las interacciones informadas:

- Los temas que motivan las interacciones deben ser convenientes tanto para el emisor como el receptor en el momento dado. Estos es por la importancia para definir cierto nivel de voluntad para interactuar [19, 41] como es argumentado en las secciones 3.1.1; 3.1.2; 3.2.3; y 3.2.5.

- Según sea el tipo de información que se notifica debe ser el medio y la forma para iniciar la interacción. Esto es tratado en las secciones 3.2.1 y 3.2.6
- La herramienta para iniciar colaboración debe contar con acceso rápido a la información de la actividad de los posibles colaboradores. Para esto es importante considerar los aspectos de distribución de información que mantenga informado de manera oportuna al grupo de trabajo y que no obstruya de manera importante la actividad del usuario. Esto surge de las secciones 3.2.2; 3.2.3 y 3.2.4.
- La herramienta debe informar la actividad de manera transparente para el grupo de trabajo. Esto es para promover la facilidad de uso, ya que es importante que el usuario realice poco esfuerzo para informarse y perturbe lo menos posible su actividad. Esto es argumentado en las secciones 3.2.4 y 3.2.5.
- Para poder distribuir la actividad del grupo, es necesario determinar la actividad individual de cada desarrollador. Esto se deriva de las secciones 3.2.3 y 3.2.6.

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

- Información de la actividad actual del receptor. Esto se refiere a conocer lo que un colega esta trabajando en un tiempo determinado. Esta información puede integrarse al conocimiento del emisor con respecto al propósito de la interacción e inferir de manera informada si el momento es oportuno para iniciar colaboración.
- Estado de disponibilidad dependiendo de la actividad actual del emisor y receptor. Esto se refiere a que los estatus de usuarios en las herramientas de comunicación son rígidos en el sentido de que se despliega el mismo estatus a todos los contactos. Por lo que no es posible a través de las herramientas de comunicación saber si un colega esta realizando una actividad idéntica, similar o diferente a la propia.

Como podemos notar, las herramientas descritas en este capítulo, permiten entre otras cosas coordinar actividades a través de notificaciones de estatus de los artefactos, localización de expertos, distribución de presencia de usuarios y recordatorios de actividades pendientes. Dichas herramientas 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 para inicios de colaboración informados 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.

Hasta este punto se han abordado las siguientes preguntas de investigación: ¿Qué es necesario para determinar la oportunidad de colaboración? y ¿Qué es necesario para identificar y determinar el momento adecuado para entrar en colaboración?.

Con base al entendimiento adquirido del trabajo de DSD (descrito en el Capítulo 2) y en los hallazgos planteados en este Capítulo, se cuenta con elementos que pueden guiar el planteamiento de una propuesta para entrar en colaboración de manera informada. Por ello, es necesario abordar en el siguiente capítulo las preguntas: *¿Cómo y qué tipo de información se necesita obtener de la gestión personal de actividades?* y *¿Cómo la gestión personal de actividades puede apoyar a determinar la oportunidad de interacción?*

3.4 Resumen del capítulo

En este capítulo se han descrito una serie de trabajos relacionados con las áreas de manejo de interrupciones y el área de desarrollo distribuido de software, de los cuales se han obtenido hallazgos importantes para guiar una propuesta para iniciar colaboración en DSD. Como parte de los resultados de este análisis se obtuvo lo siguiente:

- Se describió el manejo de interrupciones y las estrategias que se han propuesto para mitigar el efecto negativo de la interrupción.

- Se describió el reto de comunicación inadecuada del DSD y como ha sido abordado por los investigadores.
- Se enlistaron una serie de hallazgos e ideas para los inicios de interacción informados.

En el siguiente capítulo se presenta un modelo para inicios de interacción informados, el cual fue definido como resultado del modelo de trabajo para DSD presentado en el capítulo anterior y los hallazgos obtenidos de este capítulo.

Capítulo 4

“Quién sabe qué hacer, también sabe cuándo”.

Arquímedes

4. Esferas de trabajo colaborativas

En los capítulos anteriores se estableció en qué consiste el trabajo del desarrollo distribuido de software (DSD, por sus siglas en inglés) y qué trabajos de investigación contribuyen de alguna manera a los inicios de colaboración informados de manera síncrona.

Como resultado de ello, se establecieron un conjunto de hallazgos que motivan la necesidad planteada de realizar entradas en colaboración de manera informada (ver sección 3.3). Por ello, en este capítulo se presenta una caracterización de las actividades de los desarrolladores distribuidos de software basado fundamentalmente en literatura de ingeniería de software, y además se argumenta como los desarrolladores manejan sus actividades personales desde su espacio de trabajo y cómo esto fue conceptualizado como esferas de trabajo. Tomando como fundamento lo anterior se propone extender el concepto de esfera de trabajo a esferas de trabajo colaborativas a través de la integración del concepto de colaboración potencial.

4.1 Características de las actividades del desarrollo de software

De acuerdo con Pressman [35], 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 cuales son las características de las actividades del desarrollo de software, en la literatura revisada, no se encontró una caracterización del trabajo de este tipo de organizaciones. En cambio existen diversas propuestas que se centran en:

- las características del software mismo (producto),
- las características de las actividades que se realizan como parte del desarrollo (proceso) o
- las características de la organización.

En este trabajo se consideran los siguientes elementos para caracterizar el desarrollo de software, inspirados por las propuestas de [2, 34], 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 [108].
 - Distribución geográfica: los miembros del equipo pueden estar ubicados de manera co-localizada, localmente distribuida y remotamente distribuida o global [14],
 - 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); por ejemplo, si el desarrollo es mediano y se considera que lo realice una sola

persona, es muy probable que la duración sea de meses o incluso años.

- *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 [2, 38]. La incertidumbre es sensible a la percepción de los participantes, ya que es muy común que lo que representa un grado de avance en una tarea de desarrollo para uno, no necesariamente es percibido de la misma manera por el otro. Esto conduce a la necesidad de interactuar o compartir información para lograr establecer de común acuerdo el grado de avance para dicha tarea, con lo cual la incertidumbre disminuye.

- *Interdependencia*: esta característica se refiere a las dependencias que se presentan entre las diferentes actividades que realizan los desarrolladores. Según [34] estas dependencias pueden ser por :
 - recursos compartidos, cuando las actividades de un trabajador sean dependientes de la liberación de algún recurso para continuar con su tarea;
 - asignación de tareas, cuando el trabajador requiere que el líder del proyecto le especifique cuáles serán sus actividades para iniciar con ellas;
 - relación productor-consumidor, cuando un trabajador tiene que terminar su tarea y su producto sirve como insumo para realizar la tarea pendiente de otro trabajador;

- restricciones con requisito previo, cuando el trabajador tiene asignada una tarea que pertenece a una secuencia de tareas que se tienen que realizar con anterioridad;
 - transferencia, se refiere al medio a través del cual el trabajador debe hacer llegar el producto a otro trabajador que lo necesita para realizar su actividades;
 - utilidad, cuando el trabajador consumidor recibe un producto, tiene la dependencia del beneficio o utilidad que el producto le proporcione para realizar su tarea;
 - restricciones simultáneas, cuando se realizan dos tareas en paralelo por trabajadores distintos para lograr una meta en común, dependen uno del otro para poder cumplirla; y
 - relación tareas/sub-tareas, cuando una tarea se divide en sub-tareas y éstas se asignan a distintos trabajadores.
- *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 [97].
 - 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). También, la comunicación informal es útil para buscar compañeros relacionados con la tarea que se realiza, tomar o dejar mensajes a los compañeros, acordar reuniones entre colegas para revisar contingencias, liberar documentos que avalan lo que se entrega como producto, ofrecer u obtener ayuda en relación a situaciones técnicas o de la metodología, dar noticias o reportar el avance de un reporte de tareas [18].
 - 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). La comunicación formal soporta los procesos de entrega de productos para avalar que se terminó cierta tarea, reportar contingencias (formatos), avisar cambios en la estructura organizacional (circulares), documentar productos (manuales y/o diagramas), entre otros [18].

Con esta caracterización es posible conocer los distintos elementos o aspectos que conforman las actividades de DSD. Con esto, en la siguiente sección se procede a caracterizar una unidad de trabajo práctica que se apegue a entender el manejo de las actividades individuales de los desarrolladores en su entorno de trabajo.

4.2 Gestión personal de actividades

La administración personal de actividades (PAM, por sus siglas en inglés), ofrece una perspectiva individual, documentada e informada del trabajo que los trabajadores tienen que realizar, de una manera organizada [48]. PAM está inspirado en el análisis de procesos y estrategias que están involucradas con la forma en que los trabajadores enfrentan la planeación y gestión de sus actividades. Este concepto está sustentado en la unidad de trabajo denominada “Esfera de Trabajo”, la cual se describe en el siguiente apartado.

4.2.1 Esfera de trabajo

Esfera de trabajo es una definición propuesta en [46], la cual ofrece una manera de conocer la frecuencia con que la gente se cambia de una actividad a otra. Este trabajo está centrado en entender las diferentes formas en que un individuo define, delimita y diferencia sus actividades.

Por lo tanto las esferas de trabajo fueron definidas como:

“un conjunto de eventos interrelacionados que comparten un motivo u objetivo común, involucran la comunicación o interacción con una constelación particular de personas, utiliza recursos únicos y tiene su propio marco de tiempo individual”.

Las características de la esfera de trabajo se mencionan a continuación:

- Permite a las personas definir sus actividades al nivel de detalle que la persona misma decida, por lo que obviamente el detalle de cada esfera de trabajo de las personas será diferente.
- Es lo suficientemente flexible para representar las actividades como la persona lo decida, además el concepto permite asociar recursos necesarios para la realización de la misma actividad.
- Pueden ser compartidas con otros, lo cual significa que la misma esfera puede tener diferente importancia para diferente gente.
- Tiene dos niveles de compromiso: Central y periférico. Es central cuando tiene importancia primaria para un individuo; de otra manera es periférico.
- Las esferas de trabajo también son manejadas de dos formas básicas: Casos urgentes, que necesitan ser resueltos al instante; o casos por defecto que implica el trabajo esperado con un tiempo calendarizado. La esfera de trabajo puede variar a veces entre ser urgente (cuando algo va mal) o involucra su estado esperado de eventos (cuando es parte del proceso normal de la actividad).
- Le da importancia a las tareas pendientes y permite definir y asociar tareas pendientes a las esferas de trabajo de manera individual.

Con la motivación de las características que abarca la esfera de trabajo, se han realizado algunos trabajos para manejar múltiples esferas [109, 110], los cuales permiten la manipulación de esferas de trabajo y sus recursos de información mediante la representación de sus actividades, enlaces a diversos recursos de información, información de contactos relevantes a la actividad e información contextual relacionada a dichas

actividades. Sin embargo, el concepto de esfera de trabajo está limitado a un enfoque individual por lo que en el siguiente apartado se propone una extensión a este concepto.

4.3 Esferas de trabajo colaborativas

Después de realizar el análisis del concepto de esferas de trabajo y entender la forma en que se fragmenta el trabajo al estarse intercambiando de actividad, se determinó que dicho concepto está enfocado a actividades personales como se describió en la sección anterior. No obstante el DSD demanda el trabajo en grupo, el cual por la naturaleza de la distribución de los colaboradores hace que el trabajo se vuelva más complejo. Esto se manifiesta por ejemplo: cuando no se cuenta con información relacionada de un compañero al que se le quiere consultar respecto a alguna unidad de trabajo que realizó en determinado momento; cuando se quiere saber quién realizó cierta unidad de trabajo que sirve de recurso para realizar las actividades propias; e inclusive cuando se desea saber en qué condición se encuentra un compañero para interrumpirlo o abordarlo. Para ello se incluyó al concepto de espacios de colaboración potencial [102], el cual tiene la finalidad de permitir a los colaboradores contar con una perspectiva personal y parcial del entorno completo del espacio de colaboración. Con esto los colaboradores pueden en primer lugar descubrir, identificar o crear oportunidades de colaboración; en segundo lugar identificar un momento adecuado para iniciar interacción; y en tercer lugar iniciar una interacción inicial que pueda ser consecuencia para interacciones futuras que completen o reanuden el trabajo colaborativo.

Algunas de las motivaciones para crear el concepto de esferas de trabajo colaborativas provienen de la integración del trabajo individual de los desarrolladores con el concepto de espacios de colaboración potencial [102], 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). Dicha información debe ser accedida mediante un monitor (herramienta tecnológica) que muestre información del contexto de los usuarios y de las unidades de trabajo mismas que previamente fueron compartidas.

Así pues, las esferas de trabajo colaborativas podrían permitir a los colaboradores a 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 [7].

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).

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 requerimientos que condujeron a las características que deben ser cubiertas por una herramienta para contar con este tipo de soporte.

4.3.1 Requerimientos identificados

Basados en la caracterización de las actividades de desarrollo de software previamente descrita, la literatura de procesos de desarrollo de software [35], los conceptos de esfera de trabajo [46] y espacios de colaboración potencial [102], se identificaron un conjunto de requerimientos que deben ser cubiertos para iniciar colaboración de una manera más informada en DSD (ver Tabla 4.1). En esta tabla se puede notar que los requerimientos están centrados en la necesidad por interactuar pero de una manera informada. Es decir, que la interacción sea oportuna tanto para el que inicia la interacción (emisor) como para el que está siendo buscado (receptor).

La Tabla 4.1 está dividida en dos columnas: la primera presenta las características de las actividades a manera de categorización. Y en la segunda se describen las necesidades

o requerimientos de interacción que se identificaron en la literatura. Cada clasificación se describe a continuación:

- *Escala*: se abordó bajo la perspectiva de proximidad artificial e interacción informal. Debido a que dependiendo del tamaño del proyecto de desarrollo de software es la cantidad de colaboradores para realizarlo, se definió la necesidad de interacción entre el grupo de trabajo. En este caso se requiere interacción entre los colaboradores de un grupo que tienen compromiso por completar una unidad de trabajo en común (R1), y además se requiere interacción con colaboradores que no están involucrados pero que eventualmente pueden aportar a la unidad de trabajo (R2).

Tabla 4.1. Requerimientos identificados para interacción.

CARACTERÍSTICA DE LA ACTIVIDAD	REQUERIMIENTOS IDENTIFICADOS
Escala	R1: Interacción entre miembros de la unidad de trabajo. R2: Interacción con colaboradores externos a la unidad de trabajo.
Incertidumbre	R3: Conocimiento del grado de avance por unidad de trabajo R4: Conocimiento de las unidades de trabajo asignadas por persona. R5: Conocimiento del objetivo general del programa. R6: Conocimiento de las metas donde inciden las unidades de trabajo. R7: Conocimiento de los estatutos (program charter) que rigen el programa. R8: Controlar las especificaciones de los proyectos.
Inter-dependencia	R9: Conciencia de los estados de los recursos. R10: Conciencia de los estados de los miembros por unidad de trabajo. R11: Coordinación en unidades de trabajo comunes o dependientes entre miembros de la unidad de trabajo.
Comunicación	R12: Conciencia del estado de las personas que colaboran en el programa. R13: Acceso de externos a recursos internos del programa. R14: Acordar cambios imprevistos relacionados con las unidades de trabajo. R15: Entrar en contacto con la persona adecuada en el momento adecuado. R16: Medio de comunicación aceptable.

- *Incertidumbre*: esta perspectiva está dada por la gestión de proyectos. Para iniciar una interacción de manera informada, existe el requerimiento de conocer información relacionada con las unidades de trabajo planteadas en los proyectos. Por lo que se requiere contar con el conocimiento del avance que tienen las unidades de trabajo (R3) para que los colaboradores cuenten con una expectativa de lo que resta por hacer; conocimiento de las unidades de trabajo asignadas (R4), lo cual le permita a los colaboradores organizarse de manera individual para llevarlas a buen término; conocer el objetivo general del programa (R5) y las metas (R6) a las que están adheridas las unidades de trabajo, ofrecerá a los colaboradores una perspectiva completa hacia donde se están enfocando los esfuerzos al realizar cada unidad de trabajo; el conocimiento de los estatus que rigen el programa (R7) permitirá a los colaboradores entender cómo deben llevarse la suma de esfuerzos para completar unidades de trabajo en grupo y bajo qué condiciones deben de realizarse; y finalmente controlar las especificaciones de los proyectos (R8) permitirá a los colaboradores tener una vista general del proyecto con respecto a su avance en general y a detalle por unidad de trabajo asignada.
- *Interdependencia*: la perspectiva fue basada en el concepto de esferas de trabajo y gestión de proyectos. Esto fue debido al requerimiento de tener conocimiento de la unidad de trabajo que realiza cada desarrollador en un momento determinado. Para esto se requiere contar con distribución de información de conciencia referente a los estados de los recursos involucrados en las unidades de trabajo (R9), y de los estados de actividad individual de los colaboradores filtrados por la unidad de trabajo (R10) que realizan en un determinado momento. Esto les permitirá a los colaboradores contar con información de manera rápida de la actividad actual del grupo de trabajo. También se requiere la coordinación entre unidades de trabajo comunes o dependientes entre miembros del grupo de trabajo (R11), que permita a los colaboradores interactuar y aportar elementos que justifiquen el grado de avance de las unidades de trabajo y del programa.
- *Comunicación*: esta perspectiva se basa en el concepto de espacios de colaboración potencial. Esto surge por la necesidad de intercambiar información de desarrollador a desarrollador en momentos adecuados. Para iniciar interacción de

manera informada se requiere que se distribuya información de conciencia del estado de las personas que colaboran en el programa u organización (R12), esto para que los colaboradores cuenten con elementos de información de la actividad de los miembros del grupo de trabajo. También se requiere que la comunicación o interacción se pueda dar tanto con personas que están y no en el grupo de trabajo actual del colaborador (R13). Además se requiere lograr acuerdos referentes a cambios imprevistos en las unidades de trabajo (R14), esto para que logre un mejor entendimiento entre los colaboradores que están implicados en unidades de trabajo dependientes. Todo esto deberá ser soportado con elementos suficientes que proporcionen soporte para la entrada en contacto con la persona adecuada en el momento apropiado (R15) a través de un medio de comunicación aceptable (R16), el cual permita el intercambio de información adecuado entre las partes involucradas.

Es importante mencionar que los requerimientos descritos en esta sección son utilizados para definir implicaciones de diseño que debe cumplir un desarrollo de software con soporte para inicios de colaboración informados con el enfoque de las esferas de trabajo colaborativas.

4.3.2 Implicaciones de diseño

Con base en los requerimientos previamente descritos, se propone un conjunto de implicaciones de diseño para proporcionar el apoyo adecuado para el inicio de colaboración entre los trabajadores de DSD (ver Tabla 4.2). Dichas implicaciones se describen brevemente a continuación:

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, aún cuando los involucrados en el proyecto se incrementen 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.

Tabla 4.2. Implicaciones de diseño.

CARACTERÍSTICA DE LA ACTIVIDAD	IMPLICACIONES DE DISEÑO
ESCALA	I1: Servicios que permitan la comunicación entre personas.
INCERTIDUMBRE	I2: Mecanismo para compartir y filtrar información del proyecto entre colegas relacionados con las unidades de trabajo.
INTER-DEPENDENCIA	I3: Mecanismo que permita conocer el grado de avance de las tareas que están realizando individualmente cada uno de los miembros del proyecto. I4: Mecanismo que permita localizar e interactuar a los miembros relacionados en tareas comunes o dependientes.
COMUNICACIÓN	I5: Mecanismo que permita conocer el estado de los miembros del proyecto, así como también la tarea que están realizando. I6: Mecanismo que identifique cuando un usuario puede interactuar con otro, basado en un perfil de necesidades, estado y/o tarea que está realizando. I7: Servicio de comunicación síncrona y asíncrona.

En el caso de *Interdependencia*, se necesita en primer lugar, (I3) un mecanismo que permita conocer el grado de avance de cada uno de los miembros del proyecto en sus actividades, 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 las implicaciones de diseño identificadas, procedemos a describir el concepto de *esferas de trabajo colaborativas* en el siguiente apartado mediante un modelo conceptual, con el cual se pretende ilustrar los diferentes conceptos que están implicados en los inicios de colaboración informados.

4.3.3 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.

Como se puede notar, la Figura 4.1 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 mediante un círculo, el cual contiene los eventos, personas, actividades, objetivos, acciones y recursos que definen la forma en la cual la gente realiza una actividad en particular.

La Figura 4.1 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. entrar en colaboración, si es el momento adecuado.

Estos tareas o pasos representan el proceso de colaboración potencial [102, 111]. 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.

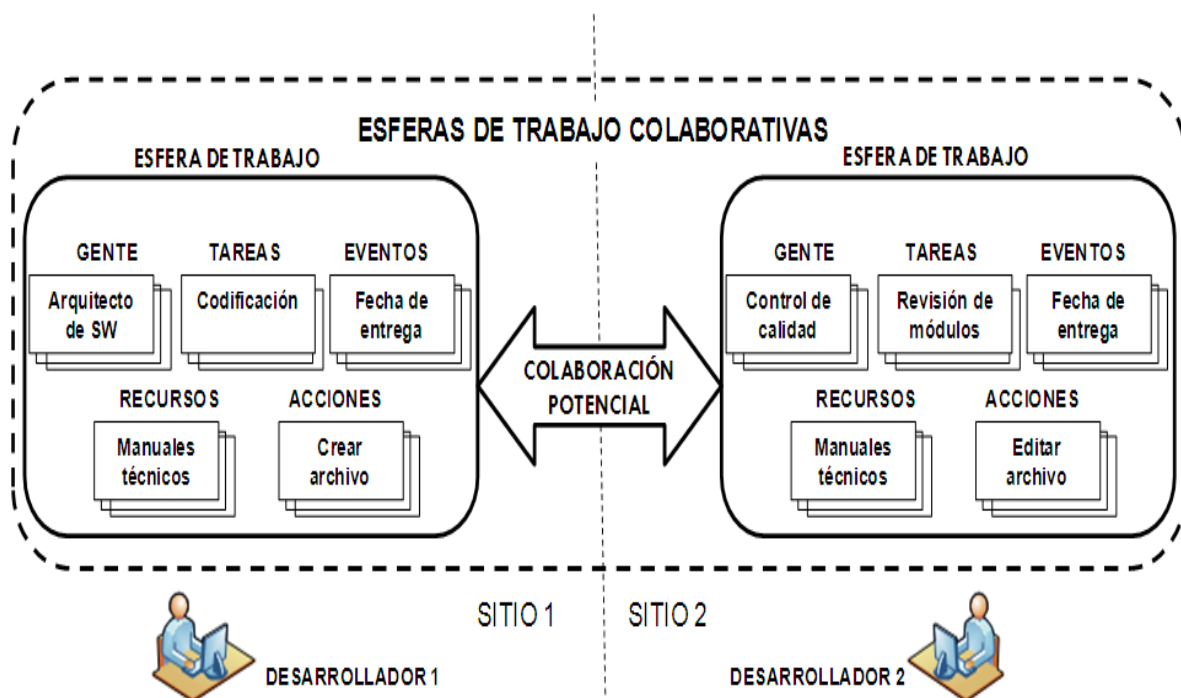


Figura 4.1. Modelo conceptual de esferas de trabajo colaborativas.

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. Con esta integración podría ser posible determinar si el momento actual es adecuado para iniciar la colaboración.

Con las implicaciones de diseño identificadas en el apartado anterior y el modelo conceptual previamente descrito, es posible caracterizar con un enfoque tecnológico a las esferas de trabajo colaborativas. Esta caracterización se presenta en la siguiente subsección.

4.3.4 Características

Una vez que se logró un entendimiento del concepto de esferas de trabajo colaborativas mediante un modelo conceptual, se decidió estudiar los requerimientos tecnológicos que son necesarios para soportar el modelo de *esferas de trabajo colaborativas*. La Tabla 4.3 presenta las características que debe cumplir la tecnología; también indica las implicaciones para el diseño (ver Tabla 4.2) que se abordan. Una descripción de cada característica se presenta a continuación:

Interacciones iniciales remotas (C1): es necesario incluir un mecanismo que permita la interacción entre los colegas remotos que están enfocados en unidades de trabajo previamente asignadas. Este mecanismo permitirá determinar la necesidad o interés en iniciar una interacción, en función de si es (o cuando es) el momento más adecuado para comenzar una interacción. Esta característica está alineada con las implicaciones de diseño I1 e I6 descritas en la Tabla 4.2.

Conciencia de colaboración potencial (C2): es necesario incluir una serie de elementos relacionados con colaboración potencial, que incluya información sobre las siguientes categorías: Quién (presencia, identidad y otros colaboradores); Qué (acciones, intenciones, recursos y habilidades); Dónde (lugar, la visión y alcance) Por qué (motivación); Cuándo (historial de eventos, dependencias y expectativas) y Cómo (historial de acciones, recursos y actividades); además sería conveniente incluir un conjunto de mecanismos mediante los cuales sea posible compartir información resultante de la misma unidad de trabajo. Esta característica está alineada con I2 e I5.

Tabla 4.3. Características de esferas de trabajo colaborativas.

Características (Implicación de diseño)	
C1	Soporte para interacciones iniciales remotas (I1 e I6).
C2	Soporte para conciencia de colaboración potencial (I2 e I5).
C3	Soporte para gestión personal de actividades (I3 e I4).
C4	Soporte para comunicación remota (I7).

Gestión personal de actividades (C3): Esto requiere la inclusión de mecanismos para vincular el trabajo realizado individualmente a las unidades de trabajo propuestas por la organización. Esto debe llevarse a cabo a través del uso de artefactos digitales o de los recursos resultantes del trabajo individual. Por otra parte, se debe contar con un conjunto de servicios que ofrezcan información referente al trabajo actual de los colaboradores, sus unidades de trabajo y los recursos. Otro requerimiento, es que los mecanismos se presenten de manera implícita y la información se adapte con respecto al contexto de trabajo de los colaboradores potenciales. Además, los mecanismos con los que se representan los diferentes niveles de conciencia, deben ser acordes a la actividad actual del colaborador potencial. Finalmente, se debe incluir un mecanismo que permita al colaborador crear grupos de contactos de manera transparente acorde a la unidad de trabajo que se realiza. Este mecanismo debe permitir la comunicación entre los miembros del grupo de trabajo. Esta característica está alineada con I3 y I4.

Comunicación remota (C4): es necesario incluir servicios de comunicación remota síncrona y asíncrona que permitan el intercambio de información entre los colaboradores de manera adecuada y sencilla, así como un conjunto de herramientas tecnológicas que permitan la comunicación privada entre los colaboradores que están involucrados en una unidad de trabajo común. Esta característica está alineada con I7.

Una vez definida la caracterización de las esferas de trabajo colaborativas, en el siguiente apartado se ilustra una esfera desde dos perspectivas: la individual y la de grupo.

4.3.5 Representación propuesta

Las esferas de trabajo colaborativas, en esencia, consideran soporte tanto a nivel individual como grupal. En la Figura 4.2 se ilustra la perspectiva individual del usuario que responde a la pregunta ¿cómo veo al grupo de trabajo?. En esta figura se denotan 3 partes importantes del modelo de la esfera de trabajo colaborativa: (A) la manipulación o realización de actividades, (B) la interacción con la tecnología y (C) el proceso cognitivo a través del cual se logra la conciencia de colaboración potencial.

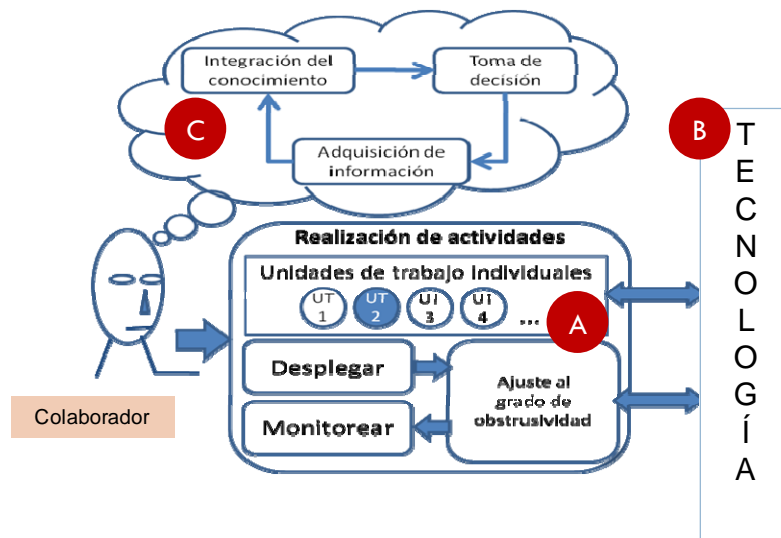


Figura 4.2. Representación desde la perspectiva individual.

Desde la perspectiva individual, durante la realización de sus actividades un colaborador del proyecto puede monitorizar información de sus compañeros de trabajo referente a la unidad de trabajo que se está trabajando o manipulando. A través de tal monitorización, el colaborador obtiene elementos de información de las actividades actuales que realizan sus colegas, lo cual es posible a través de la detección de la unidad de trabajo en la que está trabajando actualmente, en este caso “UT2”, la cual es sombreada en azul (4.2A). Para poder obtener y proporcionar información entre los colegas desde el espacio de trabajo, existe interacción con tecnología (4.2B). Dicha tecnología permite al

colaborador monitorizar las actividades de los colegas y además proporcionar o desplegar información hacia el grupo de lo que él mismo está trabajando. Tanto para la monitorización y el despliegue de información, existe un nivel de ajuste en la información adquirida y enviada con la finalidad de que sea lo menos intrusivo posible al grupo de trabajo. Dicha información adquirida a partir de los colegas, hace posible que el colaborador la integre con su conocimiento interno y ésta le ayude a tomar una decisión informada (4.2C).

En la Figura 4.3 vemos la perspectiva grupal que proporcionan las esferas de trabajo colaborativas.

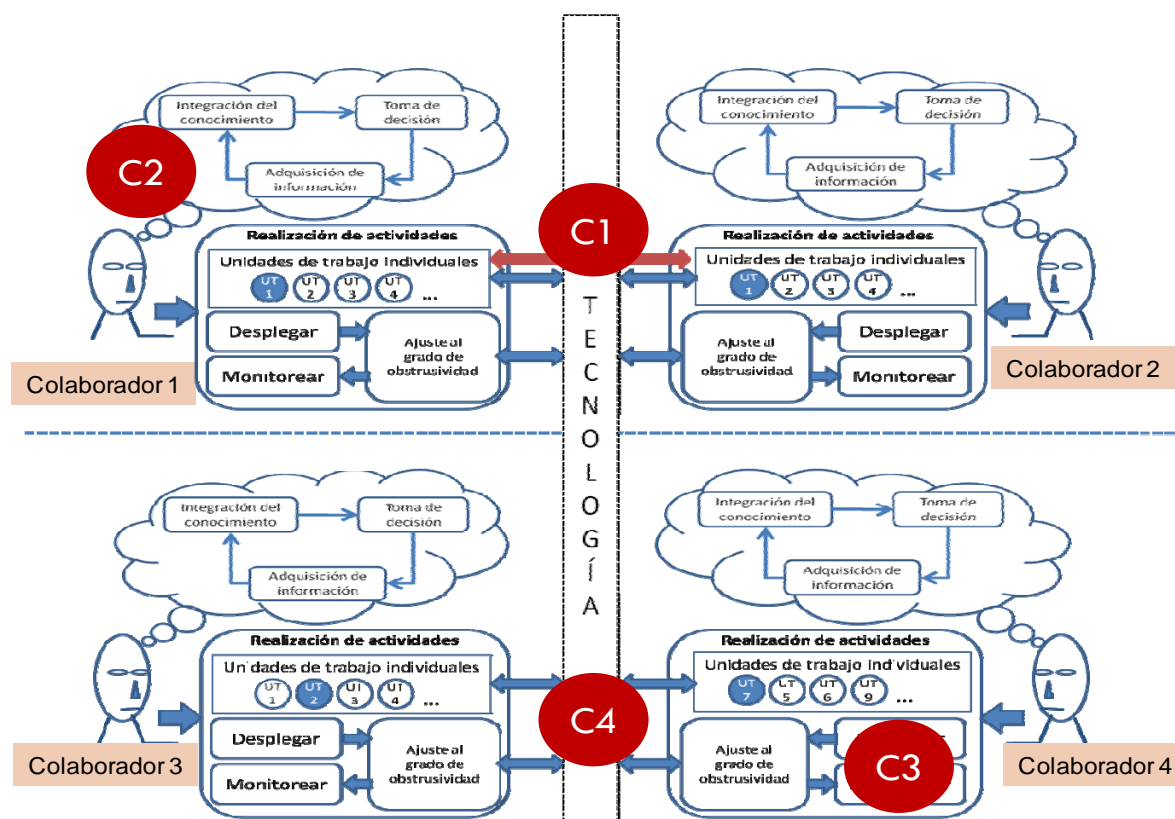


Figura 4.3. Representación desde la perspectiva grupal.

En esta figura se puede notar que los cuatro colaboradores interactúan o comparten la misma tecnología. Para explicar esta figura resaltaremos la caracterización antes descrita.

La tecnología es parte importante para las esferas de trabajo colaborativas, debido a que proporciona soporte para iniciar interacciones entre los colaboradores (C1). Esta interacción es iniciada, puesto que el colaborador 1 (emisor), con base en la información adquirida respecto al colaborador 2 (receptor), decide que tanto el tema que va a tratar con su compañero y el momento son adecuados, puesto que están trabajando en la misma unidad de trabajo (C2).

Por otro lado, la misma tecnología que indica al colaborador 1 que el momento es adecuado para iniciar colaboración con el colaborador 2, también les indica a los colaboradores 3 y 4 que los colaboradores 1 y 2 no están disponibles por estar realizando unidades de trabajo distintas (C3). Por lo que la tecnología puede hacer esto posible mediante la comparación de las unidades de trabajo que están siendo manipuladas por los miembros del grupo de trabajo. De tal manera que la tecnología tiene debidamente registrado que el colaborador 1 y 2 están trabajando en la “UT1”, que el colaborador 3 está trabajando en la “UT3” y que el colaborador 4 trabaja en la “UT7”. Con estos elementos de información distribuidos entre el grupo de trabajo se hace posible la obtención de conocimiento o conciencia de lo que está sucediendo en el entorno de trabajo del grupo.

Cabe mencionar que todos los colaboradores pueden interactuar entre sí a través de la tecnología de comunicación remota (C4) discretamente informada (unobtrusively), es decir la tecnología colecta y distribuye la información de la actividad actual del grupo de trabajo con la intervención mínima de cada uno de los colaboradores. Con esto la tecnología puede mantener informados a los involucrados. Sin embargo no impide la interacción entre colaboradores que se encuentran trabajando en unidades de trabajo diferentes. De tal forma que finalmente la decisión por aceptar o no la colaboración es responsabilidad del receptor.

4.3.6 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 interacción informados”.

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 oportunidades de colaboración [102, 111].

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

4.4 Resumen del capítulo

En este capítulo se ha proporcionado una definición inicial para esferas de trabajo colaborativas, la cual surgió del análisis de literatura y de los requerimientos identificados en el capítulo anterior. Los principales resultados de este capítulo incluyen:

- Se describieron conceptos que intervienen en el manejo de actividades personales
- Se definió un modelo conceptual para esferas de trabajo colaborativas.
- Se caracterizaron las actividades del desarrollo de software con base en la literatura y la experiencia propia.
- Se identificaron una serie de implicaciones de diseño con base en las características de las actividades del desarrollo de software y los requerimientos identificados para iniciar interacción de una manera informada.
- Se diseñó una representación de las esferas de trabajo colaborativas tanto desde la perspectiva individual como grupal.
- Se definió una caracterización que debe cumplir la tecnología que busca proporcionar soporte para esferas de trabajo colaborativas.
- Se definió el concepto de esferas de trabajo colaborativas.

En el siguiente capítulo se presenta una primer evaluación de la funcionalidad de las esferas de trabajo colaborativas.

Capítulo 5

“La ciencia avanza a pasos, no a saltos”.

Thomas Macaulay

5. Identificando las preferencias de los trabajadores del DSD: estudio preliminar

En el capítulo anterior se presentó la conceptualización de esferas de trabajo colaborativas y una propuesta de estatus de usuario llamada disponibilidad selectiva. Con base en esto, en este capítulo se ha diseñado una primera versión de interfaces de usuario para una herramienta que permita iniciar interacción de manera informada en el desarrollo distribuido de software (DSD, por sus siglas en inglés).

La propuesta consiste en un mensajero instantáneo extendido con elementos de información que cumplen con las características tecnológicas para esferas de trabajo colaborativas descritas en la sección 4.3.4. A este mensajero instantáneo propuesto se le ha denominado CWS-IM (Collaborative Working Sphere – Instant Messaging, por sus siglas en inglés).

Aquí se presenta un caso de estudio basado en escenarios de uso [20] para conocer si los trabajadores del DSD perciben útil el concepto de esferas de trabajo y disponibilidad selectiva para las actividades del DSD. Este estudio tiene como fin contrastar la primera

versión de interfaces con el uso de la mensajería instantánea (IM, por sus siglas en inglés) a través de la percepción de un grupo de desarrolladores en términos utilidad y facilidad de uso.

Para esto se crearon un conjunto de escenarios donde se introdujo a CWS-IM como un mensajero instantáneo que cuenta con soporte para iniciar interacciones informadas mediante la distribución de información de conciencia de la actividad actual de los integrantes del grupo, la cual fue definida con base en los requerimientos e ideas de diseño obtenidas de la literatura tratada en el capítulo 4.

Para recabar la información de la percepción de los participantes en cada uno de los escenarios, se utilizó el modelo de aceptación de la tecnología (TAM, por sus siglas en inglés) propuesto por [112].

El caso de estudio se dividió en cuatro pasos: la delimitación de los objetivos que deben alcanzarse, la proposición de las hipótesis, la validación de las hipótesis y el análisis de los resultados. Las siguientes secciones describen las cuatro fases realizadas para este caso de estudio.

5.1 CWS-IM: primera versión de interfaces de usuario

Para poder llevar a cabo el caso de estudio basado en escenarios fue necesario diseñar las interfaces de usuario para CWS-IM. Dichas interfaces estuvieron basadas en los elementos de información reportados en las herramientas para entrar en colaboración que fueron descritas en la sección 3.2.6.

En la Tabla 5.1 se presentan las características para las Esferas de Trabajo Colaborativas (ver sección 4.3.4), las cuales están asociadas con los elementos de información previamente identificados en la literatura con la finalidad de cumplir la característica donde fueron clasificadas.

Una descripción breve de cada una de las características y sus elementos de información se presenta a continuación:

Tabla 5.1. Elementos de información para CWS-IM

<i>Característica de esferas de trabajo colaborativas/</i>	<i>Elementos de información</i>
Soporte para interacciones iniciales remotas	<ul style="list-style-type: none">• Presencia• Identidad
Soporte para conciencia de colaboración potencial	<ul style="list-style-type: none">• Información de la actividad del contacto• Notificación de eventos• Información de actividad del contacto
Soporte para gestión de actividades personales compartidas	<ul style="list-style-type: none">• Lista de actividades
Soporte para comunicación remota	<ul style="list-style-type: none">• Equipo de trabajo• Lista de contactos

En lo que se refiere a la característica *Soporte para interacciones iniciales remotas*, se asocia con los elementos de presencia e identidad. Esto es por la necesidad de conocer quiénes están disponibles para iniciar interacción en un determinado momento y quienes no lo están.

En cuanto a la característica de *Soporte para conciencia de colaboración potencial*, se asocia con los elementos información de la actividad del contacto, notificación de eventos e información de actividad del contacto. Se requiere conocer que es lo que está o estuvo haciendo cada colega del grupo de trabajo.

Referente al *Soporte para gestión de actividades personales compartidas*, se asocia con la lista de actividades. Estos es porque se requiere conocer qué actividades han sido asignadas y quiénes tienen actividades en común.

En cuando al *Soporte para comunicación remota*, se asocia con los elementos de equipo y de trabajo y lista de contactos. Puesto que la propuesta es una herramienta basada en IM, se requiere un listado de contactos el cual este acorde a la actividad de proyecto que se está trabajando.

Como resultado de esto se obtuvo una primera versión de interfaces gráficas de usuario, la cual se ilustra en la Figura 5.1.

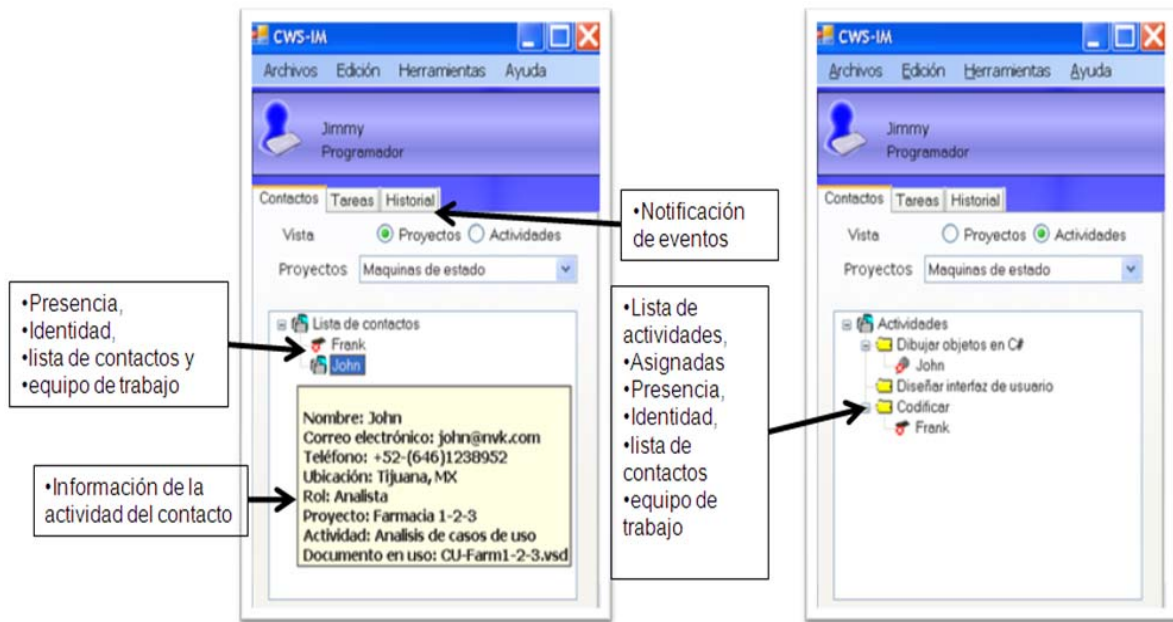


Figura 5.1. Primera versión de interfaces de usuario para CWS-IM

Con las interfaces de usuario diseñadas se procedió a crear escenarios actuales y proyectos del entorno de trabajo de DSD. Dichos escenarios son descritos en la siguiente sección.

5.2 Descripción de los escenarios de trabajo

Para llevar a cabo el estudio, los participantes tuvieron que trabajar con los escenarios de mensajería instantánea tradicional (IM) [113] y con los de CWS-IM. Para esto los participantes realizaron tres tareas específicas en cada escenario, las cuales fueron:

T1. *Buscar a un compañero de equipo y determinar su estatus de trabajo:* El escenario ubica al sujeto como un programador que está en una actividad de revisión de código de un proyecto C# .NET, para lo cual el programador edita un artefacto compartido (IU-Maq-Edo.sln). Durante la revisión, el programador (caracterizado por el sujeto del estudio) detecta un posible error en el código, por lo cual necesita consultar con el colega que lo realizó. Para esto, dicho colega ha sido previamente identificado por los comentarios realizados en el mismo código. Enseguida, el sujeto se cuestiona: ¿qué estará

haciendo? ¿Podrá atenderme?. Con esto el sujeto utiliza la herramienta y trata de determinar la actividad en la que trabaja su colega.

T2. *Identificar las actividades asignadas pendientes:* El sujeto fue contextualizado en la necesidad de conocer qué actividad debe realizar en un momento dado. Para ello, el sujeto utiliza la herramienta o las herramientas para obtener información acerca de las actividades por proyecto que le han sido asignadas. En este escenario, los sujetos tuvieron la opción de usar el sistema de administración de proyectos o consultar vía mensajero instantáneo al administrador del proyecto. Estas herramientas fueron proporcionadas para efectos del escenario.

T3. *Obtener información de la actividad actual que un compañero de trabajo está realizando en un determinado momento:* En este escenario un colega que tiene el rol de programador finalizó el módulo “Máquinas de Estado” como parte de su actividad de codificación. Por otro lado, otro programador (caracterizado por el sujeto) depende de la entrega de dicho módulo para continuar su propia actividad de codificación. De modo que para efectos del escenario, se le proporcionó al sujeto la herramienta de comunicación y el sistema de gestión de proyectos. En este escenario, el sujeto tiene por objeto encontrar una notificación acerca de la actividad dependiente, por lo que el sujeto podía decidir hacerlo por medio del sistema de gestión de proyectos o consultándolo vía mensajero al administrador del proyecto.

Cabe mencionar que las Figuras 5.2 y 5.3 contienen marcas de estas tareas (T1, T2 y T3 respectivamente), con las cuales se indica el componente que fue utilizado en los escenarios. Estos escenarios fueron presentados bajo dos condiciones de trabajo, las cuales se denominaron condición actual y condición proyectada.

Para la condición actual, los participantes utilizaron la herramienta de mensajería instantánea tradicional (IM) para realizar las tres tareas previamente mencionadas con el apoyo de un sistema de administración de proyectos (ver Figura 5.2).

En este caso para realizar la T1 (Buscar a un compañero de trabajo y determinar su estatus de trabajo), los participantes hicieron uso del listado de contactos proporcionado por

IM para identificar al contacto buscado y mediante su estatus de usuario tratar de inferir su estado de disponibilidad (ver Figura 5.2-T1).

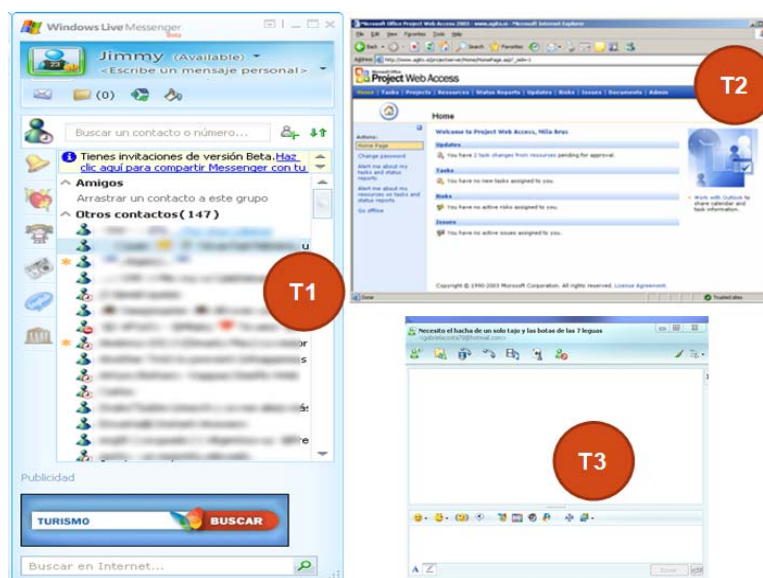


Figura 5.2. Interfaces para la condición con uso de la Mensajería Instantánea tradicional.

Para realizar la T2 (Identificar las actividades asignadas pendientes), el sujeto tenía a su disposición la herramienta IM y el sistema de administración de proyectos. De tal manera que el sujeto interactúa con el administrador del proyecto para consultar algunos detalles de las actividades (p. ej. fecha de entrega, prioridad, etc.) y el resto de la información referente a la actividad la consultaron en el sistema de administración de proyectos (ver Figura 5.2-T2).

Para realizar la T3 (Obtener información de la actividad actual que un compañero de trabajo está realizando en un determinado momento), el sujeto utiliza IM para consultar a determinados colegas mediante la ventana de conversación en que actividad se encuentran trabajando en ese momento (ver Figura 5.2-T3).

Para la condición proyectada, los participantes trabajaron con CWS-IM con las mismas tareas (ver Figura 5.3). A continuación se presenta la descripción de estos escenarios donde se involucran los componentes y la funcionalidad de CWS-IM.

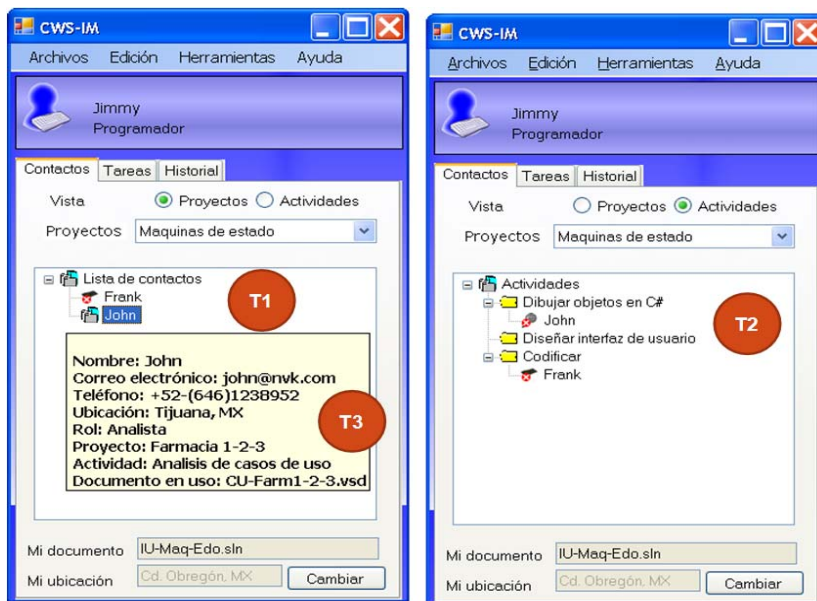


Figura 5.3. Interfaces para la condición con uso de CWS-IM.

Para realizar la T1 (Buscar a un compañero de trabajo y determinar su estatus de trabajo), se agregó a la interfaz un listado de contactos. En este caso el sujeto al estar editando el documento “IU-Maq-Edo.sln”, de manera automática CWS-IM identifica a que proyecto pertenece dicho artefacto y filtra el listado de contactos incluyendo sólo a los colegas que están involucrados en actividades de este proyecto. Además, mediante la representación del ícono del contacto puede determinar el rol en que se encuentran trabajando cada uno de los miembros del grupo. Al mismo tiempo en el ícono se les incluyó información de la actividad a través de un color representativo, el cual sigue el siguiente código de color:

- *Verde*: si el colega está trabajando en la misma actividad y proyecto.
- *Amarillo*: si el colega está trabajando en una actividad diferente pero del mismo proyecto.
- *Rojo*: Si el colega está trabajando en un proyecto diferente.

Con estos elementos agregados en la interfaz, el sujeto de un “vistazo” obtiene información relacionada a la actividad de trabajo que realiza el contacto (p. ej. proyecto en que trabaja, bajo que rol y si la actividad que realiza es similar a la del emisor).

Para realizar la T2 (Identificar las actividades asignadas pendientes), la interfaz CWS-IM incluye una vista diferente de la interfaz inicial (ver Figura 5.3-T1), la cual es accedida mediante la selección del botón de actividades (“radio button” en la Figura 5.3-T2). De tal manera que el sujeto puede visualizar sus actividades asignadas e identificar de manera rápida qué actividades son en grupo mediante los íconos del contacto y su estatus de usuario basados en el código de colores descrito anteriormente.

Como se puede notar, esta parte del diseño surge de la necesidad de consultar las actividades asignadas en herramientas organizacionales o documentos. Realizar esto implica en diferentes ocasiones iniciar sesión en una herramienta de administración de proyectos, mediante la cual se tiene acceso a toda la información referente a todos los proyectos de la organización donde se tiene que localizar la información específica que es buscada. En este sentido para lograr la T2 solo basta cambiar de vista (“radio button” en la Figura 5.3-T2) para obtener las actividades asignadas y los colegas con los que se comparte la responsabilidad de cada actividad. Esta función beneficia particularmente al iniciador de la colaboración pues se evita que se haga esta búsqueda en repositorios o documentos.

Para realizar la T3 (Obtener información de la actividad actual que un compañero de trabajo está realizando en un determinado momento), la interfaz CWS-IM puede acceder a información adicional de la actividad actual de dicho contacto mediante la selección del usuario buscado en la lista de contactos y presionando click derecho. Esta información incluye:

- Nombre completo del contacto.
- Información para establecer contacto vía correo electrónico o teléfono.
- Ubicación geográfica actual.
- Rol de trabajo en que se desempeña en ese momento.
- Nombre del proyecto en el que trabaja en determinado momento.
- Descripción breve de la actividad en la que trabaja en determinado momento.
- Nombre del documento en uso, en el cual se realiza la actividad.

Estos elementos fueron agregados atendiendo a la necesidad de proporcionar al iniciador de interacción información de la actividad actual de los receptores. De esta manera el usuario cuenta con más elementos de información para inferir la actividad que efectúa o efectúan sus posibles contactos.

5.3 Objetivos del estudio

Los objetivos a evaluar con base en los escenarios de uso son:

1. La utilidad y facilidad de uso de CWS-IM para buscar un colaborador y determinar su estatus de trabajo.
2. La utilidad y facilidad de uso de CWS-IM para encontrar las actividades asignadas pendientes.
3. La utilidad y facilidad de uso de CWS-IM para adquirir información de la actividad actual ejecutada por un colaborador (P. ej. avance de una actividad).

5.4 Hipótesis

En la actualidad, la industria de software utiliza diferentes aplicaciones para realizar sus actividades, incluidos los sistemas de gestión de proyectos, la mensajería instantánea y el correo electrónico. Con estas aplicaciones los trabajadores del DSD pueden obtener información de la presencia del usuario, las actividades asignadas y pueden determinar de alguna manera lo que un colega está haciendo. Para ello, un colaborador puede hacer uso del estatus de usuario de la mensajería instantánea para enterarse si un colega está conectado y eventualmente disponible para ser abordado mediante dicha aplicación. Además con el apoyo de otras aplicaciones (p. ej. sistema de gestión de proyectos) los trabajadores se informan de las actividades que tienen asignadas. Por lo que con este tipo de información e interacciones vía mensajería instantánea con los colegas, el colaborador puede enterarse y conocer (o al menos estimar) en qué actividad está trabajando el grupo de trabajo. Sin embargo para lograr que un colaborador se entere de esto en un determinado momento, es posible que tenga que indagar alguna información de manera individual a sus colegas, lo cual traería como consecuencia la inversión de tiempo y esfuerzo por parte del

colaborador (emisor), y la inevitable interrupción de actividades a los miembros del grupo (receptores).

En cambio, CWS-IM hace posible obtener la misma información de manera rápida y con un mínimo esfuerzo por parte de los usuarios. Por lo que se supone que podría ser más útil y fácil de usar, ya que proporciona una interfaz integrada y funcional. Por ello, con la finalidad de comparar la condición actual, donde la herramienta de comunicación es IM complementada con otras herramientas de gestión y la condición proyectada, donde se sustituye dicha herramienta con CWS-IM, se enmarca el estudio mediante el establecimiento de las siguientes hipótesis de trabajo:

H0: No existe diferencia en la utilización en cuanto a la funcionalidad de las herramientas CWS-IM y la IM tradicional.

H1: La herramienta CWS-IM facilitará la búsqueda de un colaborador y su estatus de trabajo actual (utilidad percibida).

H2: La herramienta CWS-IM será más fácil de utilizar en la búsqueda de un colaborador y su estatus de trabajo actual (facilidad de uso percibida).

H3: La herramienta CWS-IM será más útil para identificar las actividades asignadas pendientes (utilidad percibida).

H4: La herramienta CWS-IM será más fácil de utilizar en la identificación de actividades pendientes (facilidad de uso percibida).

H5: La herramienta CWS-IM será más útil para adquirir información de la actividad actual de un colaborador (utilidad percibida).

H6: La herramienta CWS-IM será más fácil de utilizar en la adquisición de información de la actividad actual de un colaborador (facilidad de uso percibida).

5.5 Diseño del estudio

El estudio fue realizado utilizando el paradigma de diseño Within Subject (todos los participantes estuvieron implicados en ambas condiciones). El diseño Within Subject es la mejor forma de asegurar que los grupos que trabajan en las diferentes condiciones de la evaluación son tan similares como sea posible, ya que en realidad son los mismos participantes. En este caso, todos los participantes estuvieron involucrados en ambas condiciones (ellos trabajaron con IM y CWS-IM). Asimismo un grupo utilizó primero el IM y posteriormente el CWS-IM. El otro grupo utilizó en primer lugar CWS-IM y después IM. Con esto se tuvo la ventaja de garantizar el control de variables referentes a la diferencia entre perfiles de participantes que podrían haber contaminado los resultados del estudio y lo referente al problema del aprendizaje [114].

En el estudio hubo una variable independiente: La condición de trabajo (Con IM o CWS-IM). Las variables afectadas por esto fueron:

- *Utilidad*: para probar si una herramienta fue más útil que otra para un conjunto de tareas.
- *Facilidad de uso*: para probar si una herramienta fue más fácil de usar que otra.

5.6 Participantes

Los participantes fueron 16 trabajadores de diferentes compañías de tres ciudades diferentes del estado de Sonora, México. Todos ellos (16) habían trabajado o estaban trabajando en proyectos de desarrollo distribuido de software (DSD).

La edad promedio fue de 30.25 años y siete (7) de ellos tenían al menos 3 años de experiencia en DSD. Todos los participantes (16) habían concluido su licenciatura en ciencias computacionales o afín. Sólo uno de ellos tenía su grado de maestría en ciencias computacionales.

Los participantes fueron divididos en dos grupos. Esta división fue realizada tomando en cuenta su experiencia en DSD. Cada grupo fue de 8 personas, 4 de ellas tenían menos de tres (3) años de experiencia en DSD y el resto tenían más de tres (3) años. Esto fue para evitar que los grupos quedaran formados con más personas con experiencia que en otro.

5.7 Procedimiento

Las actividades que realizaron los participantes se resumen a continuación:

- *Reunión inicial.* Todos los sujetos se reunieron en una sala para una plática introductoria. Se les pidió que llenaran el cuestionario para recabar información para el perfil de la muestra. (Duración 10 minutos)
- *Interacción con el primer conjunto escenarios.* En una primera sesión, que duró 15 minutos, resolvieron tres tareas en la condición actual (grupo 1) y la condición proyectada (grupo 2).
- *Interacción con el segundo conjunto escenarios.* En una segunda sesión, que duró 15 minutos, resolvieron tres tareas en la condición actual (grupo 2) y la condición proyectada (grupo 1).
- *Cuestionario de salida.* Después de terminar cada escenario (tres por sesión), los participantes fueron entrevistados y se les pidió responder el cuestionario TAM para obtener su percepción con respecto a la utilidad y facilidad de uso de la herramienta propuesta por tarea.
- *Procesamiento de datos.* Los datos obtenidos fueron procesados utilizando estadística descriptiva para cuantificar los niveles de percepción de los participantes en cuanto a utilidad y facilidad de uso.

5.8 Limitaciones

El caso de estudio descrito y los métodos utilizados para evaluar la herramienta propuesta, podría haber tenido debilidades. La influencia de estas debilidades en los resultados y la forma en que los abordamos son:

- *Los estudios within subject tienen el problema de aprendizaje.* Se intentó evitar este efecto mediante la interacción de los escenarios en orden diferente. Sin embargo, es imposible establecer si este efecto fue totalmente controlado o evitado.

- *La gente usualmente responde los cuestionarios con lo que ellos piensan que el investigador quiere oír. Se intentó evitar esto mediante el llenado de cuestionarios de manera anónima. Las personas usualmente son más sinceras cuando no necesitan dar su nombre.*
- *Este estudio está enfocado en obtener datos de utilidad y facilidad de uso de las opiniones de los usuarios. Sin embargo, no se dispone de datos cuantitativos para probar, por ejemplo, si los usuarios pasaron más tiempo realizando una tarea mediante el uso de una herramienta en particular.*

5.9 Resultados

Como se mencionó anteriormente, se obtuvo la percepción general de los participantes respecto a la utilidad y facilidad de uso de las herramientas mediante cuestionarios TAM [112]. El cuestionario constaba de dos secciones (con seis preguntas cada uno) como se presenta en las Tablas 5.1, 5.2 y 5.3. Todos los reactivos del cuestionario se midieron en una escala Likert 7, que van desde 1 (“Extremadamente en desacuerdo”) a 7 (“Extremadamente de acuerdo”).

Para el procesamiento de los datos obtenidos de los cuestionarios, se utilizó el paquete estadístico S.P.S.S. versión 15.0.1 [115]. Para llevar a cabo la comparación entre las dos condiciones de trabajo (actual y proyectada), se realizó un análisis mediante la prueba t de Student para muestras independientes. Dicho análisis fue para comparar las medias con respecto a la percepción de cada herramienta involucrada (IM o CWS-IM) en las condiciones de trabajo y determinar si existe diferencia significativa entre ellas.

Para decidir si los participantes preferían una herramienta sobre la otra para un escenario determinado, la prueba t calcula un valor p basado en el desempeño de los datos. Si el valor p es lo suficientemente pequeño (valor de $p < 0.05$), se concluye que la diferencia es significativa [116].

Por lo tanto, la Tabla 5.2 muestra las medias y desviaciones estándar para el escenario 1 (Esc. 1). Esta tabla es útil para probar las hipótesis 1 y 2 referente al escenario 1 (buscar a un compañero de trabajo y determinar su estatus de actividad).

Tabla 5.2. Resultados TAM referentes al escenario 1.

Utilidad		
Pregunta	IM	CWS-IM
	Esc. 1	Esc. 1
P1: Utilizando este sistema en mi trabajo podría llevar a cabo mis tareas más rápidamente.	4 (1.549)	5.81 (0.655)
P2: Utilizando este sistema mejoraría mi desempeño laboral.	3.44 (1.504)	5.31 (1.302)
P3: Utilizar este sistema haría que fueran más fáciles las tareas de mi trabajo.	3.81 (1.276)	5.69 (1.078)
P4: Utilizando este sistema mejoraría mi efectividad en el trabajo.	3.44 (1.459)	5.25 (1.483)
P5: Utilizando este sistema incrementaría mi productividad	3.19 (1.328)	4.81 (1.223)
P6: Encontraría útil este sistema en mi trabajo.	4.81 (1.109)	6.13 (0.806)
<i>Promedio</i>	<i>3.84</i>	<i>5.54</i>
Facilidad de uso		
Pregunta	IM	CWS-IM
	Esc. 1	Esc. 1
P7: Encontraría fácil que el sistema hiciera lo que yo quiero que haga.	4.13 (1.821)	5 (1.155)
P8: Interactuar con el sistema sería claro y entendible.	5.44 (1.365)	5.88 (1.025)
P9: Aprender a operar este sistema sería fácil para mí.	5.75 (1.291)	5.75 (0.856)
P10: Es fácil llegar a ser hábil en el uso del sistema.	6.25 (0.775)	6 (0.632)
P11: Encontraría fácil de usar el sistema.	5.19 (1.167)	6.38 (0.719)
P12: Encontraría flexible interactuar con el sistema.	5.69 (1.195)	5.88 (0.885)
<i>Promedio</i>	<i>5.34</i>	<i>5.77</i>

En este caso la media aritmética de la utilidad percibida fue 5.54 cuando se utiliza el CWS-IM, y 3.84 cuando se utiliza mensajería instantánea (IM). Esta diferencia es significativa, puesto que $t_{0.1} = 4.976$ y $p = 0.000$ (siendo $p < 0.05$). Estos valores por lo tanto nos indican que la herramienta CWS-IM fue percibida como más útil para la búsqueda de un colaborador y para determinar su estado de actividad actual que la mensajería instantánea (IM). Por otro lado, referente a la facilidad de uso la media aritmética fue 5.77

utilizando CWS-IM y 5.34 utilizando IM. En este caso no hubo diferencia significativa, puesto que los valores fueron $t_{01}=1.559$ y $p=0.130$ (siendo $p > 0.05$).

Tabla 5.3. Resultados TAM referentes al escenario 2.

Utilidad		
Pregunta	IM	CWS-IM
	Esc. 2	Esc. 2
P1: Utilizando este sistema en mi trabajo podría llevar a cabo mis tareas más rápidamente.	4.63 (1.025)	6.31 (0.793)
P2: Utilizando este sistema mejoraría mi desempeño laboral.	3.69 (1.493)	5.75 (0.856)
P3: Utilizar este sistema haría que fueran más fáciles las tareas de mi trabajo.	4.13 (1.310)	5.63 (1.088)
P4: Utilizando este sistema mejoraría mi efectividad en el trabajo.	3.56 (1.413)	5.63 (1.147)
P5: Utilizando este sistema incrementaría mi productividad	3.50 (1.461)	5.44 (1.153)
P6: Encontraría útil este sistema en mi trabajo.	5.13 (1.668)	6.38 (0.500)
<i>Promedio</i>	<i>4.16</i>	<i>5.86</i>
Facilidad de uso		
Pregunta	IM	CWS-IM
	Esc. 2	Esc. 2
P7: Encontraría fácil que el sistema hiciera lo que yo quiero que haga.	4.63 (1.360)	5.25 (0.856)
P8: Interactuar con el sistema sería claro y entendible.	5.13 (1.688)	6 (0.816)
P9: Aprender a operar este sistema sería fácil para mí.	5.69 (1.352)	6 (0.894)
P10: Es fácil llegar a ser hábil en el uso del sistema.	6.06 (0.929)	6.19 (0.750)
P11: Encontraría fácil de usar el sistema.	5.44 (1.459)	6.44 (0.727)
P12: Encontraría flexible interactuar con el sistema.	5.50 (1.211)	6.06 (0.772)
<i>Promedio</i>	<i>5.35</i>	<i>5.98</i>

Por lo tanto, no es posible afirmar que el CWS-IM fue percibido como más fácil de usar que IM para la búsqueda de un colaborador y determinar su estado de actividad actual.

La Tabla 5.3 muestra los medias y desviaciones estándar para el escenario 2 (identificar las actividades asignadas pendientes). Estos resultados ayudan a comprobar si las hipótesis 3 y 4 podrían ser aceptadas.

En este caso, para el factor Utilidad el valor de las medias aritméticas fueron 5.86 utilizando CWS-IM y 4.16 utilizando IM.

Esta diferencia es significativa puesto que $t_{.01}=5.501$ y $p=0.000$. Por lo tanto, se puede afirmar que el CWS-IM fue percibido como más útil que la IM para identificar las actividades asignadas pendientes. Con respecto a la facilidad de uso, el promedio fue de 5.98 con CWS-IM, y 5.32 en el caso de la IM. Los resultados después de aplicar el estadístico t de Student fueron $t_{.01}=2.119$ y $p=0.042$. Con esto es posible argumentar que la herramienta CWS-IM fue percibida como más fácil de usar para identificar las actividades asignadas pendientes.

La Tabla 5.4 muestra los medias y desviaciones estándar para el escenario 3 (obtener información de la actividad actual que un compañero de trabajo está realizando en un determinado momento). Estos resultados ayudan a comprobar si las hipótesis 5 y 6 fueron significativas.

Para el factor Utilidad se obtuvo una media de 6.17 utilizando CWS-IM y 4.31 utilizando IM. La diferencia entre ambas medias es significativa, puesto que los resultados del estadístico t de Student fueron $t_{.01}=4.814$ and $p=0.000$. Por consecuencia es posible afirmar que la herramienta CWS-IM fue percibida como más útil que la IM para la obtención de información sobre la actividad actual en la que un colaborador está trabajando. En cuanto al factor de facilidad de uso la media fue de 6.24 cuando los participantes utilizaron CWS-IM y 5.39 cuando utilizaron IM. Una vez más la diferencia entre las medias es significativa, ya que $t_{.01} = 3.203$ y $p = 0.003$. Eso significa que el CWS-IM fue percibido como más fácil de usar que el IM para obtener información sobre la actividad actual en la que un colaborador está trabajando.

Tabla 5.4. Resultados TAM referentes al escenario 3.

Utilidad		
Pregunta	IM	CWS-IM
	Esc. 3	Esc. 3
P1: Utilizando este sistema en mi trabajo podría llevar a cabo mis tareas más rápidamente.	4.63 (1.893)	6.44 (0.512)
P2: Utilizando este sistema mejoraría mi desempeño laboral.	4.13 (1.500)	5.81 (0.834)
P3: Utilizar este sistema haría que fueran más fáciles las tareas de mi trabajo.	4.19 (1.642)	6.38 (0.619)
P4: Utilizando este sistema mejoraría mi efectividad en el trabajo.	3.81 (1.797)	5.88 (0.885)
P5: Utilizando este sistema incrementaría mi productividad	3.81 (1.759)	5.88 (0.806)
P6: Encontraría útil este sistema en mi trabajo.	4.38 (2.094)	6.69 (0.602)
<i>Promedio</i>	<i>4.31</i>	<i>6.17</i>
Facilidad de uso		
Pregunta	IM	CWS-IM
	Esc. 3	Esc. 3
P7: Encontraría fácil que el sistema hiciera lo que yo quiero que haga.	4.44 (1.315)	5.75 (0.856)
P8: Interactuar con el sistema sería claro y entendible.	5.56 (1.263)	6.25 (0.577)
P9: Aprender a operar este sistema sería fácil para mí.	6 (1.095)	6.31 (0.704)
P10: Es fácil llegar a ser hábil en el uso del sistema.	6.31 (0.704)	6.25 (0.775)
P11: Encontraría fácil de usar el sistema.	5.31 (1.537)	6.63 (0.619)
P12: Encontraría flexible interactuar con el sistema.	5.63 (1.147)	6.25 (0.683)
<i>Promedio</i>	<i>5.39</i>	<i>6.25</i>

5.10 Discusión

Una vez que los resultados han sido obtenidos, hemos tratado de analizarlos y reflexionar sobre las razones que podrían explicarlos. Por lo tanto, se proporciona una posible explicación de cada resultado de la siguiente manera:

En el caso de H1 se considera que los participantes percibieron a la herramienta CWS-IM más útil que el IM para la búsqueda de un colaborador y descubrir su estado de

actividad, porque la primera herramienta tiene un mecanismo que permite a los trabajadores conocer en qué tarea una persona está trabajando, el rol que está desempeñando en un momento determinado y el nombre del documento que se está modificando. Por otra parte, una persona puede buscar contactos de acuerdo al proyecto en el que están trabajando.

En el caso de H2, esta hipótesis no es aceptada, por un lado porque los sujetos percibieron que tanto IM como CWS-IM eran fáciles de utilizar. Sin embargo cabe la posibilidad de que cuando una persona está muy acostumbrada a trabajar con una herramienta en particular, al cambiársela por una diferente le es frecuentemente difícil de aceptarla, ya que a primera vista la segunda herramienta por lo general parece ser más compleja. Esto podría ser la razón por la cual los participantes no creen que el CWS-IM sea más fácil de usar que el IM en la búsqueda de un colaborador y el descubrimiento de su estado referente a la actividad. Por ello, se pondrá atención especial en cómo ésta podría ser mejorada a fin de facilitar su uso.

Por otra parte, la H3 fue aceptada porque los participantes consideraron que CWS-IM era más útil que IM para identificar las actividades asignadas pendientes. Una posible razón es que el CWS-IM tiene un panel en el que las tareas asignadas de los usuarios se muestran. En este panel el usuario puede visualizarlas de manera rápida y consultar la fecha límite en que deben llevarse a cabo.

La H4 también fue aceptada porque los trabajadores consideraron que era más fácil localizar las tareas asignadas pendientes mediante el uso de CSW-IM. Esto pudo deberse a que CWS-IM obtiene automáticamente la información relacionada con las tareas asignadas desde el repositorio de proyectos. Por lo tanto, el usuario puede acceder a esta información desde la misma interfaz. En el caso de la utilización de mensajería instantánea (IM), el usuario tiene que consultar con el director del proyecto para obtener esta información o consultarlas con herramientas adicionales, lo que significa que el usuario tiene que hacer un esfuerzo adicional que es innecesario cuando la información está en la misma interfaz de la herramienta.

Por último, el hecho de que H5 y H6 fueran aceptadas indica que CWS-IM es percibido como útil y fácil de usar para obtener información sobre la actividad actual en la que un colaborador está trabajando en relación a la mensajería instantánea (IM).

Con los resultados de este caso de estudio, se considera que el desarrollo de una herramienta como CWS-IM sería útil y aceptado de forma favorable en el contexto de trabajo del DSD.

5.11 Resumen del capítulo

En este capítulo se han presentado los resultados un estudio preliminar para percibir si un grupo de desarrolladores consideraba útil y fácil de usar nuestra propuesta que denominamos CWS-IM. El estudio fue realizado mediante el uso de escenarios [20] con la finalidad de evaluar la funcionalidad de CWS-IM con base en tres escenarios en particular:

- *Buscar a un compañero de trabajo y determinar su estatus de actividad.* Se cuenta con evidencia para afirmar que CWS-IM posee elementos de información útiles para determinar de manera rápida y con poco esfuerzo quiénes tienen actividades en común y que actividad están realizando. En cambio cuando se tiene la condición de la IM, para poder determinar esta información se tiene que hacer un esfuerzo adicional para encontrar la información tanto en repositorios del proyecto como con las personas a través de interacciones.
- *Identificar las actividades asignadas pendientes.* En este caso la evidencia nos mostró que los participantes apreciaron como útil los elementos de información que están en la interfaz de CWS-IM, particularmente la referente a la lista de contactos con la vista de actividades. Asimismo consideraron que la forma de obtener estos elementos de información es fácil de aprender y no necesitan de conocimientos que requieran gran esfuerzo para entenderlos.
- *Obtener información de la actividad actual que un compañero de trabajo está realizando en un determinado momento.* La evidencia nos indicó la utilidad percibida por los participantes en relación a la forma de obtener información e integrarlo con su conocimiento respecto a la actividad de un colega. A su vez los mecanismos o componentes propuestos para mostrar la información fueron considerados fáciles de usar.

Los resultados de este estudio preliminar, dan indicios que motivan la implantación de las Esferas de Trabajo Colaborativas en un prototipo tecnológico. Es por ello, que en el siguiente capítulo se describen los detalles de diseño e implantación de esta herramienta propuesta.

Capítulo 6

“La ciencia es la expresión de una necesidad inherente al ser humano y, en todo caso, está ligada a la función superior de su naturaleza inteligente: la capacidad de crear”.

René Gerónimo Favaloro

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

En el capítulo anterior se presentó una primera versión de interfaces de usuarios para CWS-IM (por las siglas en inglés de *Collaborative Working Spheres-Instant Messaging*) y los resultados de un caso de estudio, el cual estuvo basado en escenarios de uso [20]. Dichos resultados fueron favorables a la propuesta en términos de utilidad y facilidad de uso, lo que motiva a desarrollar un prototipo tecnológico con soporte para Esferas de Trabajo Colaborativas.

La implantación de las esferas de trabajo colaborativas en una herramienta tecnológica está basada en la mensajería instantánea tradicional. Dicha herramienta fue llamada CWS-IM. La razón de esta herramienta radica en que la mensajería instantánea tradicional es una herramienta de comunicación síncrona utilizada frecuentemente por los trabajadores del desarrollo distribuido de software (DSD, por sus siglas en inglés). Sin embargo estas herramientas 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 más informados.

Por lo anterior, en este capítulo se describe de manera detallada el desarrollo de interfaces de usuario para entrar colaboración de manera informada mediante CWS-IM. Además, se describe como la información es obtenida y distribuida a través de un sistema multi-agente y cómo esta información es presentada al usuario final por medio de un conjunto de componentes de interfaz gráfica de usuario (GUI, por sus siglas en inglés). Finalmente también se muestran los detalles relacionados con la implantación misma de CWS-IM y la descripción de su funcionamiento.

6.1 Interfaz gráfica

Para que el sistema CWS-IM permita ofrecer soporte adecuado para inicios de interacción informados, debe proporcionar elementos de información que permitan al usuario integrarlos a su conocimiento para tomar la decisión que más convenga con respecto al inicio de una colaboración.

6.1.1 Elementos de información

En este apartado, se identifican y definen los elementos de información clave para la herramienta, los cuales dan respuesta a las preguntas ¿Quién?, ¿Dónde?, ¿Qué?, ¿Por qué?, ¿Cuándo? y ¿Cómo? Por lo que se considera que:

“Cuando los desarrolladores distribuidos de software están soportados con información de las esferas de trabajo colaborativas, 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, POR QUÉ 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”.

CWS-IM pretende por un lado, permitir el manejo de colaboración potencial para situaciones síncronas como son: contar con conocimiento de disponibilidad de los desarrolladores que están conectados así como el rol que están asumiendo en ese momento; tener conocimiento de las unidades de trabajo que le fueron asignadas para trabajar de manera individual y grupal; notificar a los compañeros de equipo por unidad de trabajo cuando se finaliza una unidad de trabajo; y permitir la recepción de notificaciones relevantes de manera implícita referentes a las unidades de trabajo en las que se está involucrado. Por otro lado, para las situaciones pasadas o futuras permite el manejo del potencial de colaboración para situaciones asíncronas como son llegar a tener conocimiento del estado de una unidad de trabajo que se trabajó anteriormente y; obtener información de las unidades de trabajo probables a realizar.

La Tabla 6.1 muestra un conjunto de elementos básicos y preguntas para las situaciones actuales, pasadas y futuras que pueden ser utilizadas para obtener información desde CWS-IM. Cada categoría de la tabla está compuesta por elementos de conciencia del potencial de colaboración, los cuales pueden ser solicitados para ayudar a identificar datos específicos de información relacionada con cada elemento en particular.

Por ejemplo, la categoría Quién (primer renglón) se refiere a conocer si hay colaboradores disponibles, saber quién(es) es(son), quién aporta, aportó o tiene que aportar para una unidad de trabajo en específico, y con quién(es) tengo que realizar una unidad de trabajo en particular. La categoría Dónde (segundo renglón) se refiere a tener el conocimiento de la ubicación geográfica donde se encuentran los colaboradores. La categoría Qué (tercer renglón) se refiere a qué están realizando actualmente los colaboradores, qué fin tiene lo que están realizando, qué recursos están ocupando, bajo qué rol se están desempeñando, qué unidades de trabajo se tienen que realizar, qué documentos relacionados a la unidad de trabajo se han generado y cuáles están disponibles para los colaboradores. La categoría *Por qué* (cuarto renglón) se refiere a conocer la motivación por la que los colaboradores realizan cierta unidad de trabajo y contar con conocimiento de los eventos que suceden respecto a las mismas unidades de trabajo. La categoría *Cuándo* (quinto renglón) se refiere a conocer cuando suceden o van a suceder ciertas acciones u

operaciones en un tiempo determinado, así como el momento en que los colaboradores esperan que sucedan. Y finalmente la categoría *Cómo* (sexto renglón) se refiere a conocer como una acción u operación para una unidad de trabajo en particular va a ser realizada, y que nivel de avance tiene en determinado momento.

Tabla 6.1. Elementos de información para la interfaz CWS-IM.

<i>Categoría</i>	<i>Elemento</i>	<i>Pregunta</i>
Quién	Presencia	¿Quién(es) [está / están] presentes o disponibles?
	Identidad	¿Quién(es) [es / son]?
	Autoría	¿Quién(es) [está(n) / estuvo / estuvieron / estará(n)] trabajando en la unidad de trabajo?
	Agrupado	¿Con quién(es) [tengo / tendré] que realizar esta unidad de trabajo?
Dónde	Localización	¿Dónde [está / están] trabajando?
Qué	Operación	¿Qué [está(n) / estuvo / estuvieron / estará(n)] haciendo?
	Propósito	¿En qué meta [está(n) / estuvo / estuvieron / estará(n)] trabajando?
	Rol	¿Qué rol [está(n) / estuvo / estuvieron / estará(n)] desempeñando?
	Actividades	¿Qué unidades de trabajo tengo pendientes por hacer?
	Recurso	¿Qué documentos [he / hemos] generado para esta unidad de trabajo?
	Privacidad	¿Qué documentos [se permite / permitiré] que estén disponibles para los miembros del grupo?
Por qué	Motivación	¿Por qué [está(n) / estuvo / estuvieron / estará(n)] trabajando en esa unidad de trabajo?
	Notificación	¿Por qué sucedió este evento?
Cuándo	Expectativa	¿Cuándo espera(n) que termine la unidad de trabajo?
	Historial	¿Cuándo sucedió este evento de la unidad de trabajo?
Cómo	Articulación	¿Cómo vamos a articular la unidad de trabajo grupal?
	Avance	¿Cómo va(n) con el avance de la unidad de trabajo?

Con los elementos de información identificados es posible definir qué tipo de mecanismos se requieren para obtener la información para cada elemento.

6.1.2 Mecanismos

Los tipos de mecanismos que se incluyen en CWS-IM son sociales y tecnológicos [117], los cuales son explicados a continuación:

- *Mecanismos sociales*: se refiere a los que se proveen mediante la comunicación consecucional, la realimentada y la intencional. Éstas pueden ser utilizadas por los colaboradores para obtener y proveer información cuando la presencia y/o las acciones son detectadas a través de algún mecanismo tecnológico (consecucional). Además, los colaboradores pueden obtener información de conciencia de las interacciones con la actividad individual (realimentada). Finalmente, los colaboradores pueden establecer conversación relevante al grupo de trabajo involucrado en una unidad de trabajo en específico a través de un servicio de comunicación (intencional).
- *Mecanismos tecnológicos*: se refiere a la recolección y representación de la información de las interacciones realizadas con las actividades durante el trabajo individual.
 - La recolección se refiere a obtener los elementos de información, lo cual puede ser tanto por mecanismos explícitos como implícitos. Los explícitos emplean información suministrada intencionalmente por el usuario respecto a las acciones realizadas por el mismo o sus colaboradores. Por ejemplo, capturar las horas de esfuerzo realizadas por los colaboradores para una unidad de trabajo). Los mecanismos implícitos emplean información colectada automáticamente por el sistema mientras los colaboradores realizan sus actividades (p. ej. ser notificado que un contacto ya capturó las horas de esfuerzo para una unidad de trabajo).
 - La representación se refiere a proporcionar una representación interna en el sistema, lo cual puede ser establecido por mecanismos directos o abstractos. Los directos se refieren a que la información de conciencia es mostrada como es obtenida (p. ej. el texto en un chat, un flujo de video). Los abstractos se refieren a la información de conciencia pre-procesada. Ésta permite concentrar e incrementar cantidades de información en un mismo elemento de despliegue (p. ej. el listado de contactos en un mensajero instantáneo y la representación de su actividad). Para lograr esto, se requieren mecanismos de adaptación que personalicen la información recibida a un formato que es adecuado para su representación actual (p. ej. preferencias para recibir notificaciones).

Estos tipos de mecanismos permitirán al colaborador obtener información suficiente para tener conocimiento de sus colaboradores, iniciar una interacción e identificar un momento adecuado para hacerlo.

Los mecanismos que se establecieron para que CWS-IM proporcione los elementos de información al usuario se muestran en la Tabla 6.2. La tabla muestra la asociación de cada elemento de información con el tipo (implícito o explícito) y el mecanismo que será utilizado para obtener la información. Por ejemplo, el elemento *Presencia*, es un mecanismo implícito pues mediante la obtención de información transparente al usuario referente a la actividad actual de los contactos, en el listado de usuarios es posible conocer si un colaborador está disponible o no para interactuar en base a lo que está trabajando en cierto momento. Por otro lado, el elemento *identidad* es un mecanismo explícito puesto que la información debe ser proporcionada por el usuario al momento de su registro/ingreso en el sistema.

Tabla 6.2. Elementos de información para CWS-IM y mecanismos para obtenerlos.

<i>Elemento de información</i>	<i>Tipo de mecanismo</i>	<i>Mecanismo para obtener la información</i>
Presencia	Implícito	Listado de usuarios (colaboradores) por proyecto o actividad.
Identidad	Explícito	Información proporcionada por el registro.
Autoría	Implícito	Información proporcionada desde los recursos individuales del colaborador vinculados a sus actividades personales.
Agrupado	Implícito	Listado de contactos generado con base en el proyecto o actividad que se realiza en determinado momento.
Localización	Implícito	Proporcionado por la dirección IP de la conexión al entrar al sistema.
Operación	Implícito	Generado por la presencia de documentos personales involucrados en proyectos comunes.
Propósito	Explícito	Basado en las metas dadas para los proyectos y/o actividades.
Rol	Implícito	Proporcionado por la actividad que se está realizando en cierto momento.
Actividades	Implícito	Proporcionado por el listado de actividades asignadas por proyecto, el cual es generado a partir de la identidad del colaborador.
Recurso	Explícito	Vinculación de documentos generados a las actividades asignadas de manera individual.

Privacidad	Explícito	Compartir documentos generados de manera individual a uno o varios colaboradores de manera no invasiva.
Motivación	Explícito	Especificada en los proyectos mediante la información proporcionada por el listado de actividades.
Notificación	Implícito	Información generada con base en eventos durante la operación de actividades del proyecto.
Expectativa	Explícito	Información especificada en los proyectos.
Historial	Implícito	Proporcionada por las operaciones realizadas por los colaboradores.
Articulación	Explícito	Proporcionada por las interacciones en grupo mediante chat y documentos compartidos.

Con los mecanismos definidos para cada elemento de información, en la siguiente sub-sección se especifican los elementos de la interfaz gráfica de usuario (GUI).

6.1.3 Componentes de la interfaz gráfica de usuario

Para definir los componentes GUI para CWS-IM, en la Tabla 6.3 se asocia cada componente con los elementos de información que serán cubiertos y una descripción breve de cómo el usuario puede interactuar con ellos.

Tabla 6.3. Componentes de interfaz grafica de usuario (GUI).

<i>Elemento GUI</i>	<i>Elementos de Información</i>	<i>Interacción</i>
Lista de contactos	Presencia, rol y Articulación	En relación a presencia y rol es solamente informativo. Mediante una acción de doble click sobre el usuario deseado proporciona las interacciones en grupo mediante chat.
Panel de Identificación	Identidad	Sólo informativo
Panel de recursos	Autoría, Recursos, Privacidad y Articulación	Proporciona una serie de botones de comando para presentar archivos que los colaboradores han compartido, permite compartir archivos personales al grupo o algún colaborador en específico. Además permite enlazar o quitar documentos personales como recursos a las actividades asignadas.
Vista de contactos	Agrupado y Actividades	Radio Botón para filtrar el listado contactos ya sea que se desee a nivel proyecto o actividad.
Etiqueta de	Localización	Botón para cambiar la etiqueta en caso de querer

ubicación		más detalle referente a la ubicación.
ToolTip de información del contacto	Rol, recurso y operación	Mediante un click derecho sobre el contacto deseado se ofrece información adicional del mismo. Este componente es sólo informativo.
Historial de notificaciones	Operación, Autoría, Notificación e Historial	Mediante un par de componentes datePicker se filtra la historia que se va a mostrar por fecha de vencimiento. Además mediante componentes Tab Controls se muestran los diferentes tipos de notificaciones.
Panel de tareas	Propósito, Motivación y Expectativa	Mediante Tab Controls se muestra información relacionada de la tarea a diferentes niveles.

Para ilustrar la función de cada componente GUI a continuación se presenta una descripción (ver Figura 6.1):

Lista de contactos (6.1A): ésta es utilizada para mostrar los contactos que están relacionados con la actividad que se está realizando. 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. 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 (ver Figura 6.2) y compartir la conversación con todo el equipo de trabajo a pesar de que no estén conectados en ese momento.

Panel de Identificación (6.1B): 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.

Vista de contactos (6.1C): 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 6.1A) o más específicamente a nivel actividad (ver Figura 6.1F). 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 (6.1D): 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, etc.).

Etiqueta de ubicación (6.1E): este componente proporciona información de la ubicación geográfica. 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.

Historial de notificaciones (6.3A): presenta 3 diferentes categorías de eventos: archivos (p. ej. cuando se comparte un documento), conversaciones (p. ej. cuando se realiza una conversación en grupo y no estaba disponible) y de actividades (p. ej. cuando hay un cambio en la fecha de entrega de un actividad o proyecto). Las notificaciones pueden ser consultadas por fechas.

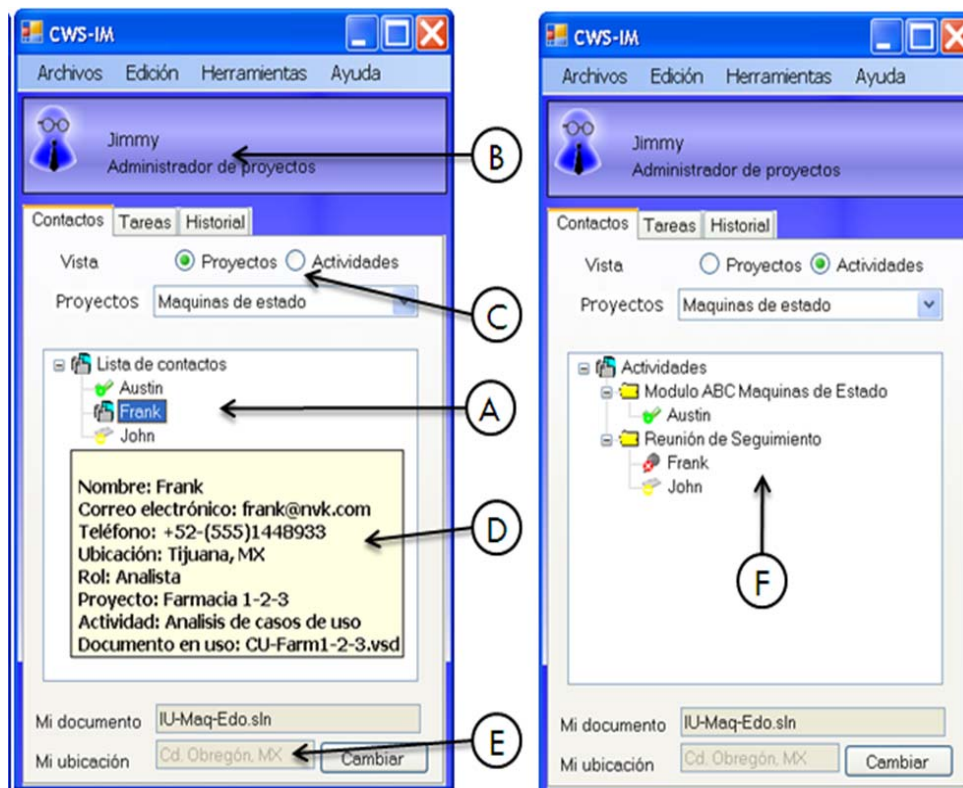


Figura 6.1. Componentes GUI de CWS-IM: Vista de contactos.



Figura 6.2. Ventana para conversación en grupo.

Panel de tareas (6.3B): Presenta diferentes perspectivas de la tarea que se está realizando, es decir a que proyecto y/o actividad pertenece, cuales son los recursos que se han generado de manera individual para la presente tarea.

Panel de recursos (6.3C): es utilizado para vincular documentos digitales generados para una actividad que fue previamente asignada. Adicionalmente desde aquí se puede compartir un recurso a un usuario o inclusive a un grupo de usuarios. Para realizarlo no es necesario que estén conectados, dicho recurso permanece guardado en el servidor en un buzón del usuario buscado y se le envía una notificación a su historial donde se informará que tiene un recurso compartido. Cuando éste es descargado, se notifica al usuario que compartió dicho recurso que éste ya fue descargado. Asimismo en el mismo panel se puede consultar qué recursos han sido compartidos a nivel individual y quiénes son los usuarios que los compartieron.

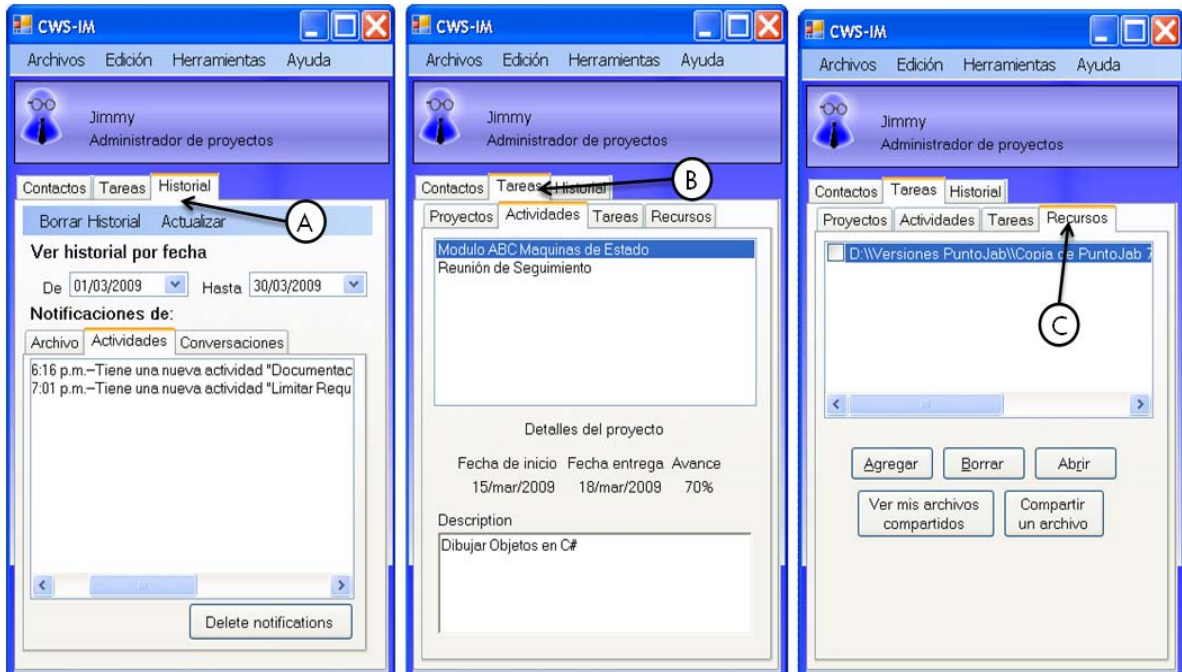


Figura 6.3. Componentes GUI de CWS-IM: Vista de tareas.

Con los componentes GUI ya descritos, en las siguientes secciones se describe el sistema multi-agente para la recolección de información en el entorno de trabajo del DSD, algunos aspectos de la implantación y la funcionalidad del sistema en un escenario de trabajo.

6.2 Obtención de información

Para la recolección de información de manera transparente para el usuario se propone considerar las siguientes fases:

- *Monitoreo:* estar alerta e identificar la unidad de trabajo (p. ej. actividad) en la cual el colaborador está trabajando.
- *Identificación:* relacionar los archivos que están siendo manipulados por el colaborador con las unidades de trabajo que le fueron asignadas.
- *Solicitud:* realizar las peticiones de información del estado actual de uno o más colaboradores, basados en la situación actual de su actividad individual en un determinado momento.

- *Formalización:* recibir las solicitudes de la fase anterior para formalizarla a través de la interpretación de los datos. En esta fase se debe verificar si la solicitud es válida o no (p. ej. si la información del proyecto está activa).
- *Procesamiento:* realizar las consultas necesarias al repositorio de la organización basado en solicitudes válidas.
- *Notificación:* informar los resultados de la petición.

Para satisfacer dichas fases mencionadas, se propone un Sistema Multi-Agente (ver Figura 6.4), donde existen básicamente un agente por cada fase, así como un agente auxiliar. Cabe mencionar que un sistema multi-agente esta formado por un conjunto de agentes de software, los cuales son un proceso computacional o pieza de software autónomo, con iniciativa y capacidad de explorar y modificar su entorno y posibilidad de comunicarse con otros agentes [118].

La descripción de cada uno de los agentes del sistema multi-agente propuesto se presenta a continuación:

Agente Monitor, apoya en la fase de monitoreo. Es el responsable de conocer información (p. ej. nombre, tipo, fecha de creación, etc.) de los artefactos (p. ej. archivos) que manipula el usuario en su computadora de trabajo. Esto demanda la ejecución de un proceso de monitorización proactivo para captar la información generada durante la interacción del usuario y las aplicaciones de la computadora (procesadores de texto, lenguajes de programación, aplicaciones de diseño, etc.).

Agente Identificador, apoya en la fase de identificación. Este agente estará encargado de identificar si el archivo que es manipulado por el usuario tiene que ver con alguna actividad y/o tarea de los proyectos de la organización. Para lograr esto se requiere de la ejecución de un proceso de búsqueda proactivo para identificar y relacionar la información generada durante la fase de monitoreo.

Agente Interfaz, Recoge la información que se envía desde la instancia de proyecto con el objetivo de actualizar la información de la interfaz gráfica de usuario (GUI).

Agente solicitante, apoya la fase de solicitud. Dicho agente estará dedicado a realizar la petición de información mediante la ejecución de un proceso de envío de datos a la instancia de proyecto mediante el uso de un canal de comunicación.

Agente Gestor, apoya en la fase de formalización. Dicho agente tendrá conocimiento de los proyectos válidos. Para esto, se solicita un proceso de búsqueda en los repositorios de los proyectos de la organización que ayuden a decidir si procede la petición o no. En caso de proceder la petición se pasa a la siguiente fase. En caso contrario se debe notificar al Agente Interfaz el motivo por el cual no procedió su petición.

Agente Proyecto, apoya la fase de procesamiento. Es un agente que tendrá a su cargo la información de un determinado proyecto. Éste solicitará un proceso de búsqueda para un proyecto en particular de la organización y el resultado será enviado al Agente Notificador.

Agente Notificador, apoya la fase de notificación. Es el responsable de informar los resultados de la petición al Agente Interfaz (mediador entre el usuario y los agentes).

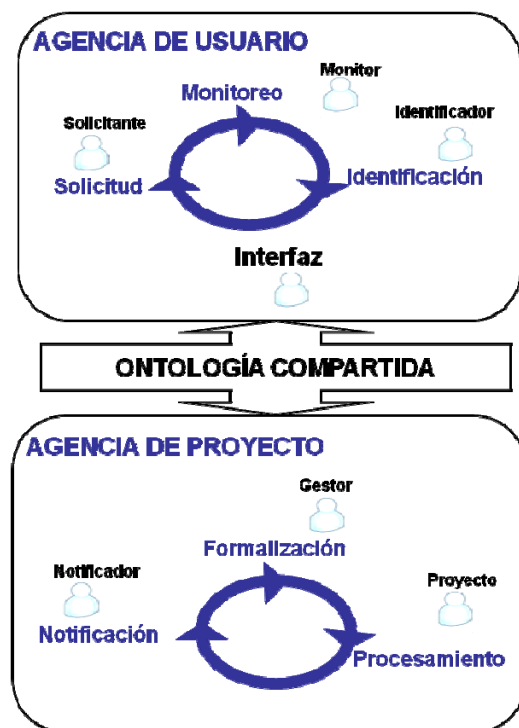


Figura 6.4. Distribución de agentes.

Como se puede notar, la Figura 6.4 muestra como los agentes están estructurados en dos instancias: *instancia de usuario* e *instancia de proyecto*. La instancia de usuario es la encargada de dar el soporte a las peticiones de información por parte del usuario interesado. Está compuesta de los agentes *Monitor*, *Identificador* y *Solicitante*. Cabe mencionar que existe un agente auxiliar dentro de esta instancia, el cual es llamado *Agente Interfaz*. Este agente recibe la información que es enviada de la instancia de proyecto para actualizar los elementos de información de la interfaz gráfica de usuario. Por otro lado, la instancia de proyecto, tiene como finalidad ofrecer la información del contexto de trabajo basado en la información obtenida de los repositorios de los proyectos y de usuarios de la organización. Esta instancia la forman los agentes *Gestor*, *Proyecto* y *Notificador*.

Además de las instancias, la Figura 6.4 tiene un componente intermedio entre las instancias llamado *Ontología compartida*. Ésta es necesaria para que haya comunicación consistente entre los agentes de las distintas instancias.

6.3 Implantación

Para describir la implantación de CWS-IM, en este apartado se presenta la descripción de un diagrama de arquitectura que muestra cada componente del sistema. Esta arquitectura está basada en un modelo cliente-servidor. La Figura 6.5 resalta 3 partes principales: el cliente CWS, un servidor de presencia de actividad y un servidor de presencia de usuarios y artefactos.

El *cliente CWS* proporciona la interfaz gráfica de usuario, la cual requiere de módulos para la detección de presencia de documentos, la presencia de usuarios y de actividad. Con estos módulos la interfaz de usuario proporciona los elementos de información actualizados, además de soportar la instancia de usuario del sistema multi-agente previamente descrito. El cliente CWS se desarrolló con el lenguaje C# .NET 2008 utilizando la biblioteca agsxmpp [119]. Esta biblioteca ofrece la funcionalidad requerida para conectarse al servidor de colaboración en tiempo real llamado Openfire 3.5.1., el cual utiliza el protocolo de mensajería instantánea XMPP (por su siglas en inglés, Extensible Messaging and Presence Protocol). Con esto se hace posible que este componente obtenga la lista de los usuarios y artefactos registrados.

El *servidor de presencia de actividad* es el que recibe las notificaciones de documentos que están relacionados con alguna unidad de trabajo de la organización. Con esta información se crea una bitácora de actividad de los clientes CWS, la cual es transmitida para actualizar la información de usuario en el servidor XMPP (Openfire) y la información de presencia de actividad en los clientes. A través de esto se actualiza la presencia de usuarios y artefactos basados a la actividad reportada en la bitácora de actividad (LOG de actividad). Esta bitácora es informada mediante un servidor de notificaciones, el cual tiene la información del documento que el usuario está trabajando. Con ello se valida si corresponde a algún proyecto y actividad, para posteriormente solicitar a través del servidor XMPP (Openfire) los clientes conectados. Este componente está soportado por el servidor de aplicaciones de Internet Information Server (IIS 5.1) mediante servicios web (webservices) y el servidor de base de datos SQL Server 2000.

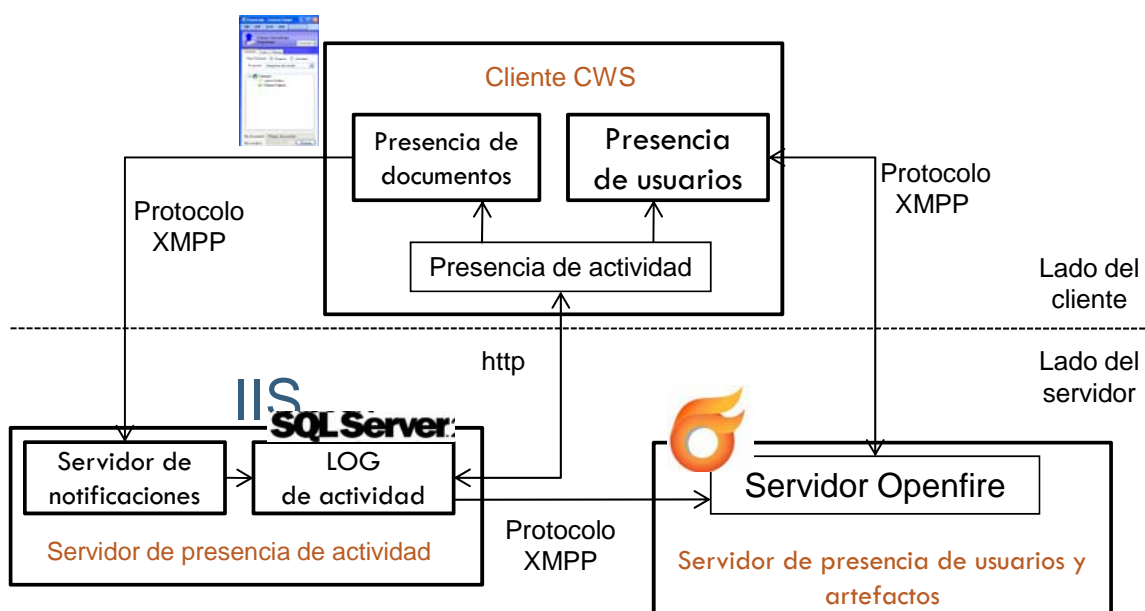


Figura 6.5. Diagrama arquitectura.

El *servidor de presencia de usuarios y artefactos* distribuye el estatus de disponibilidad de los clientes CWS. Para esto cada usuario debe ser registrado en este servidor para que el cliente CWS pueda distribuir su presencia y la de los artefactos involucrados (p. ej. documentos compartidos). Este servidor es afectado por la información

de la bitácora de actividad para realizar la distribución de presencia. Este componente fue soportado con servidor de colaboración Openfire 3.5.1. Como se mencionó previamente, el cliente CWS puede acceder a este servidor mediante el uso de la biblioteca agsxmpp.

Cabe mencionar que esta implementación utiliza datos del sistema de administración de proyectos DIGITÉ [120].

Para determinar automáticamente que archivos han sido recientemente cambiados en el lado del cliente, el *Cliente CWS* escucha los eventos de los archivos mediante la clase `FileSystemWatcher` de la librería `.NET Framework`. Además, para determinar qué aplicación y archivo está actualmente en uso (recibiendo el estado activo en la pantalla del usuario), el cliente CWS utiliza el método `GetForegroundWindow` de la librería de `Windows User32.dll` (dynamic link library). La ruta del archivo (classpath) que está en uso o modificado recientemente, es enviado al *servidor de presencia de actividad* por medio de un mensaje basado en el protocolo XMPP. En ese momento, el archivo es comparado con la información del repositorio (SQLServer) para determinar si dicho archivo está asociado a una actividad o proyecto. Una vez determinada la actividad y/o el proyecto, la información de los colaboradores implicados es obtenida del repositorio (SQLServer). Con esta información, se agrega un registro en la bitácora de actividad (LOG de actividad). De tal manera, que se registra que el usuario “U” esta actualmente trabajando en el documento “D” asociado a la actividad “A” del proyecto “P”. Asimismo, esta información es automáticamente reenviada al *servidor de usuarios y artefactos* mediante un mensaje XMPP, que a su vez actualiza los estatus de usuario y artefactos, notificando también a todos los clientes CWS involucrados con otro mensaje XMPP. Finalmente, los datos actuales para suministrar o soportar los componentes GUI (p. ej. Nombre del proyecto, actividades asignadas, usuarios, documentos, etc.) son tomados por el cliente CWS desde el repositorio (SQLServer) a través de una conexión HTTP (IIS Webservice).

Para entender de manera más clara el funcionamiento de esta herramienta, en la siguiente sección se presenta un escenario donde se explica de manera detallada como la aplicación apoya a los trabajadores del DSD.

6.4 Funcionalidad de CWS-IM

Para ilustrar de manera clara el funcionamiento de CWS-IM, se presenta la Figura 6.6, la cual representa un espacio de colaboración compartida para mostrar un escenario de aplicación que ejemplifica una situación de trabajo con el apoyo de la herramienta.

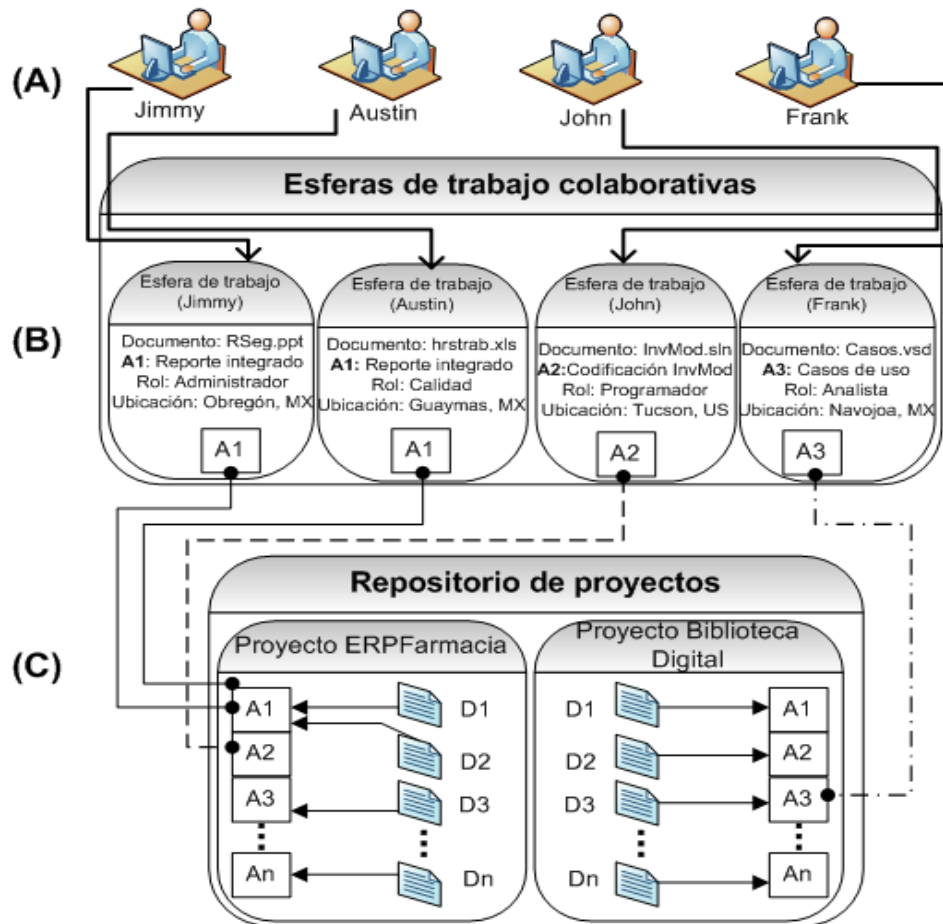


Figura 6.6. Representación de un espacio de colaboración compartido.

Se puede notar que la Figura 6.6 está dividida en tres partes: un grupo de trabajo virtual (6.6A), las esferas de trabajo colaborativas (6.6B) y el repositorio de proyectos (6.6C).

El *grupo de trabajo virtual* se compone de cuatro desarrolladores que trabajan en distintas ubicaciones geográficas. Los miembros del grupo centran sus esfuerzos en llevar a cabo una determinada actividad con un conjunto de recursos asociados y colaboradores en un proyecto en particular. Cada conjunto representa una Esfera de Trabajo [46] (p. ej.

Jimmy trabaja en la actividad A1 del proyecto ERPFarmacia mediante la modificación del documento D1)

Las *esferas de trabajo colaborativas* son el conjunto de esferas de trabajo (6.6B) que son categorizados para identificar quiénes trabajan en la misma actividad/mismo proyecto (MA/MP), en diferente actividad/mismo proyecto (DA/MP) y en diferente actividad/diferente proyecto (DA/DP).

El *repositorio de proyectos* contiene la información referente a las actividades de los proyectos, en los que la organización está comprometida. Las actividades tienen asociados documentos (p. ej. el proyecto *ERPFarmacia* contiene la actividad *A1*, la cual tiene asociados los documentos *D1* y *D2*).

En este caso, Jimmy (administrador del proyecto) está trabajando en el documento *D1:RSeg.ppt*, el cual es un documento que está asociado a la actividad *A1: reporte integrado* del proyecto *ERPFarmacia*. De manera similar, Austin (control de calidad) trabaja en la actividad *A1*, el cual es identificado durante la manipulación del documento *D2: hrstrab.xls*. Por lo que la esfera de trabajo colaborativa categoriza a las esferas de trabajo de Jimmy y Austin en (MA/MP). En cambio, John (programador) trabaja en el mismo proyecto que Jimmy y Austin pero en una actividad distinta *A2: Codificación de InvMod* mediante la manipulación del documento *D3: InvMod.sln*. La esfera de trabajo colaborativa distingue que John está trabajando en una actividad diferente pero del mismo proyecto (DA/MP) en relación a Jimmy y Austin. Además Frank (analista) está trabajando en la actividad *A3: Casos de uso* del proyecto *Biblioteca Digital*, el cual fue identificado por la manipulación del documento *D3: casos.vsd*. Por lo que la esfera de trabajo colaborativa categoriza a la esfera de trabajo de Frank en un proyecto distinto al resto del grupo (Diferente Actividad/Diferente Proyecto).

6.4.1 Interpretación del código de colores y el rol

Para interpretar la presencia y función de los contactos potenciales desde el listado de contactos de CWS-IM (ver Figuras 6.7A y 6.7F), es necesario entender el código de

colores y los íconos que se proponen para la herramienta. Dicho código de colores fue creado a partir de una nueva forma para desplegar estatus de disponibilidad denominado *Disponibilidad Selectiva* (ver sección 4.3.6).

Para explicar la interpretación de esto se presentan dos perspectivas: de lo individual a lo grupal y de lo grupal a lo individual. Para esto se describe el siguiente escenario:

Jimmy (administrador del proyecto) al estar abriendo el documento “RSeg.ppt” recibe una lista de contactos potenciales que están relacionados con el proyecto que está trabajando en ese momento. Al mismo tiempo el sistema le proporciona información de los roles que cada contacto está desempeñando mediante íconos ilustrativos (lentes =administrador de proyectos; marca de verificación=control de calidad; lupa=analista; teclado=programador; y el signo de interrogación=Sin rol específico).

A través de los colores asociados al ícono, en este caso Jimmy puede establecer si sus compañeros de trabajo están trabajando en la misma actividad y el mismo proyecto (representado por el color verde), si están trabajando en diferente actividad pero el mismo proyecto (representado por el color amarillo) o si están trabajando en un proyecto diferente (representado por el color rojo). De igual manera Jimmy puede considerar la información del rol para decidir iniciar una interacción con su colega o no.

Es importante resaltar que el ícono de rol que aparece en el listado de contactos representa el rol actual de la actividad que cada quién realiza.

Tomando en cuenta la perspectiva de lo individual a lo grupal (“cómo veo al grupo”), Jimmy es capaz de conocer los roles y actividades de sus contactos como se ilustra en la Figura 6.7A.

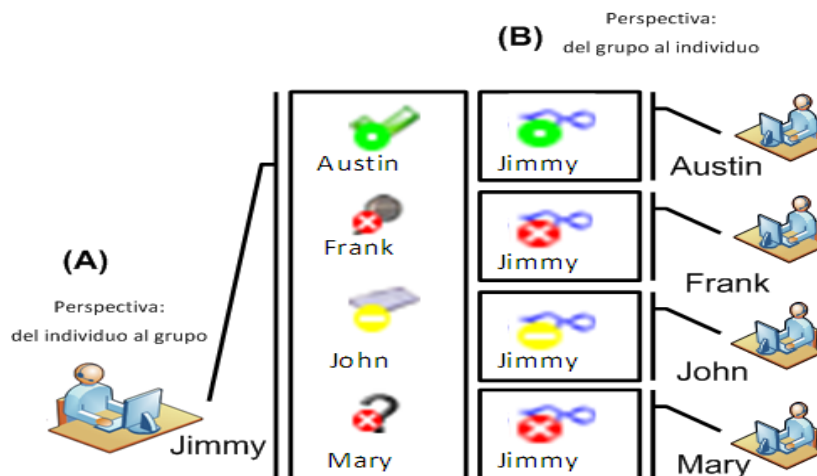


Figura 6.7. Lista de contactos de Jimmy.

Por otro lado, considerando la perspectiva de lo grupal a lo individual (“cómo me ve el grupo”), como es ilustrado en la Figura 6.7B, el rol de Jimmy (administrador de proyectos) es mostrado a sus colegas (icono de lentes) pero de distintos colores de acuerdo a la actividad de cada integrante del grupo. Por ejemplo, Austin ve a Jimmy en la misma actividad y el mismo proyecto (código verde), mientras que John lo visualiza trabajando en una actividad diferente pero del mismo proyecto (código amarillo) y Frank lo ubica trabajando en un proyecto diferente (código rojo).

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:

- *Código verde:* Disponible para aquellos colegas que están trabajando en la misma actividad y el mismo proyecto (p. ej. Jimmy y Austin).
- *Código amarillo:* Alcanzable pero ocupado para los colegas que trabajan en diferente actividad del mismo proyecto (p. ej. Jimmy y John).
- *Código rojo:* Ocupado para quienes realizan actividades de distintos proyectos (p. ej. Jimmy y Frank).

En este caso, Jimmy necesita consultar con Frank una duda referente a la actividad que realiza en ese momento (reporte integrado). Sin embargo, dada la actividad que Frank está realizando, CWS-IM lo ilustra con el “icono en color rojo”. Asimismo, Jimmy sabe

que alguien está trabajando en su misma actividad y proyecto (Austin es ilustrado con su icono en “color verde”). Con esta información Jimmy decide no contactar a Frank en ese momento pues sabe que está ocupado para quienes realizan actividades de proyectos diferentes. En cambio Jimmy decide iniciar interacción con Austin quién está disponible desde la perspectiva de CWS-IM, es decir, al estar trabajando en la misma actividad y proyecto.

6.5 Resumen del capítulo

En este capítulo se ha proporcionado una descripción de la herramienta CWS-IM, la cual surgió de los requerimientos identificados en el capítulo 4. Como parte de los resultados de este capítulo se presentan:

- La descripción de los elementos de información que proporciona la interfaz gráfica de usuario de CWS-IM.
- Se relacionaron los elementos de información con los mecanismos que apoyarían la extracción de dicha información en la herramienta.
- Como parte del resultado del diseño, se presentan los componentes GUI de CWS-IM asociados a los elementos de información.
- Además se diseñó un sistema multi-agente para la recolección transparente de información relacionada a la actividad actual del grupo de trabajo virtual.
- Se presentó el diagrama de arquitectura de CWS-IM para su implantación.
- Se describió la funcionalidad de CWS-IM mediante la representación de un espacio de colaboración compartido.
- A partir de la definición de Disponibilidad Selectiva, se definió un código de colores y la identificación de roles de los contactos potenciales para la interpretación de las perspectivas de la actividad del grupo.

En el siguiente capítulo se presenta la evaluación de CWS-IM por un grupo de trabajadores de una fábrica de software.

Capítulo 7

*“La verdadera grandeza de la ciencia acaba valorándose
por su utilidad”.*

Gregorio Marañón

7. Evaluación de un caso de estudio

En el capítulo anterior se describió el diseño e implantación de un prototipo llamado CWS-IM, el cual es un mensajero instantáneo extendido con soporte para inicios de interacción informados. A través de CWS-IM los colaboradores pueden ser notificados de la actividad actual que realiza cada colega. Con esto un emisor puede decidir si el momento es adecuado para iniciar la interacción, no sólo para sí mismo sino también para el receptor.

En este capítulo se presenta la evaluación de dicha herramienta a través de un caso de estudio, el cual consistió en introducir la herramienta CWS-IM en una fábrica de software. Uno de los objetivos de esta evaluación es captar la percepción de utilidad y facilidad de uso de dicha herramienta por parte de un grupo de trabajadores de desarrollo distribuido de software (DSD).

Además, 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. 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.

Para recabar la información percibida por los participantes, se utilizó el modelo de aceptación de la tecnología (TAM, Technology Acceptance Model) propuesto en [112] y la escala de usabilidad del sistema (SUS, System Usability Scale) propuesta en [121].

Este caso de estudio se dividió en cuatro pasos: la delimitación de los objetivos de la evaluación, el establecimiento de las hipótesis, la validación de las hipótesis y el análisis de los resultados. Las siguientes secciones describen las cuatro fases realizadas para este caso de estudio.

7.1 Objetivos

Los objetivos de este caso de estudio consisten en evaluar:

1. el impacto de CWS-IM en términos de utilidad y facilidad de uso en el ambiente de trabajo de DSD;
2. la interpretación de los estados de actividad para iniciar colaboración con un colaborador; y
3. conocer el impacto de la interpretación de los estados de actividad en términos de percepción de beneficio de la actividad.

7.2 Hipótesis

Para guiar esta evaluación se estableció un conjunto de hipótesis de trabajo, las cuales se agrupan en tres partes: Utilidad y facilidad de uso, Apoyo a Inicio de colaboración, y Nivel de interrupción de la herramienta.

En cuanto a la *utilidad y facilidad de uso* se establecen:

H0: No existe diferencia en la utilización de las herramientas CWS-IM y la IM tradicional.

H1a: Los sujetos percibirán que CWS-IM es más útil que el IM para la realización de sus actividades.

H1b: Los sujetos percibirán que CWS-IM es más fácil de utilizar que el IM para actividades de DSD.

H1c: Los sujetos del estudio percibirán que CWS-IM es más adecuado para ser adoptado que el IM para el trabajo de DSD.

En cuanto a *iniciar colaboración* se establece que:

H0: No existe diferencia en la utilización de la información proporcionada por CWS-IM respecto a la proporcionada por el IM para iniciar colaboración.

H2a: Los sujetos en posición de emisor tenderán a solicitar información adicional referente al estado del receptor (mediante el componente *ToolTip de información del contacto*) cuando el estado del receptor potencial esté en “diferente actividad / mismo proyecto” o “diferente proyecto” que cuando los estados sean “misma actividad / mismo proyecto”.

En cuanto a percepción de la *interrupción* con el uso de los estatus basados en actividad actual:

H0: No existe diferencia en la percepción de interrupción de los sujetos en posición de receptor cuando los estados del receptor y del emisor se encuentren en la misma actividad y el mismo proyecto.

H3a: El nivel de beneficio percibido de las interacciones será más alto cuando los estados del receptor y del emisor se encuentren en la misma actividad y el mismo proyecto.

En cuanto a percepción de la *interrupción* con el uso de los sistemas IM y CWS-IM:

H0: No existe diferencia en la percepción de interrupción de los sujetos en cuanto al uso del sistema.

H4a: Los sujetos percibirán que las interrupciones fueron más perjudiciales con el uso de IM que con CWS-IM.

7.3 Diseño del estudio

El paradigma de diseño para este estudio fue WithIn Subjects. Para evitar el efecto de aprendizaje, los participantes del estudio fueron divididos en dos grupos. Uno de ellos inició con el uso de CWS-IM durante una semana y posteriormente cambiaron la herramienta por IM durante una semana más. El otro grupo inició con la herramienta IM en la primera semana y continuó con CWS-IM durante la segunda semana.

En cuanto a la comunicación mediante mensajería instantánea, durante el estudio los participantes sólo utilizaron las herramientas proporcionadas (CWS-IM o IM tradicional). En relación a otros medios de comunicación, los participantes tenían acceso a otras herramientas (p. ej. correo electrónico, teléfono, etc.). Cabe mencionar que los participantes realizaban sus actividades de desarrollo desde sus espacios de trabajo.

7.4 Participantes

La evaluación fue realizada con un grupo de desarrolladores distribuidos de software de una fábrica de software mexicana. El proceso de producción de software de esta fábrica, está apegado al Proceso Unificado Racional (RUP) y emplea el Lenguaje Unificado de Modelado (UML) para el modelado visual y diseño del software. Además la fábrica cuenta con el nivel de madurez 3 en CMMI (Capability Maturity Model Integration) del SEI (Software Engineering Institute). La metodología de administración de proyectos utilizada está basada en los estándares del PMI (Project Management Institute) y tiene un grupo de administradores de proyectos en proceso de certificación como PMP (Project Management Professional).

Para la evaluación de la herramienta, se involucraron a 16 trabajadores de esta fábrica, incluyendo:

- cuatro (4) administradores de proyecto,
- dos (2) ingenieros de control o aseguramiento de calidad,

- cuatro (4) programadores y
- seis (6) arquitectos de software.

Los participantes fueron divididos en 4 equipos de trabajo acorde a sus actividades asignadas. Durante las tres semanas del estudio los participantes trabajaron en 33 actividades diferentes de 11 proyectos de la fábrica de software. El promedio de edad de los participantes fue de 24.5 años, donde 10 de ellos fueron de sexo femenino y 6 de sexo masculino. En cuanto a la experiencia laboral, los participantes promediaban 2.9 años de experiencia en DSD. Además, el promedio de proyectos asignados por colaborador era de 2.1 proyectos.

Durante el estudio, los participantes estaban particularmente involucrados en las fases de definición y ejecución de sus respectivos proyectos. Las actividades de desarrollo estaban encaminadas a aplicaciones .NET (ASP y C#) para escritorio y dispositivos móviles con acceso a bases de datos SQL Server 2000/2005. Su distribución fue:

- en diferentes construcciones del mismo sitio de trabajo (separados entre 40 y 80 metros),
- entre ciudades del mismo país (separados entre 60 y 120 kilómetros aproximadamente).
- entre ciudades de dos países diferentes (separados por 700 kilómetros aproximadamente)

En cuanto a la diferencia horaria, era prácticamente la misma y no había diferencias culturales importantes (Sonora, México y Arizona, USA) ya que los trabajadores son bilingües y se mueven entre los sitios por propósitos del proyecto mismo.

7.5 Procedimiento

En total se llevaron a cabo 7 actividades para realizar la evaluación, las cuales se describen a continuación:

1. *Reunión inicial.* Se realizó una reunión general con los participantes con el fin de informar la intención del estudio y explicarles cuál sería su participación y la forma en que se llevaría a cabo.

2. *Entrevista individual.* Se les realizó una entrevista de manera personal a cada participante, para captar la forma en que ellos organizan sus actividades y su percepción frente a las interrupciones de sus colegas.
3. *Aplicación de cuestionario inicial.* Se les pidió que respondieran el cuestionario de entrada, donde se les solicitaron los siguientes datos: edad, sexo, años de experiencia y cantidad de proyectos en lo que están involucrados.
4. *Instalación de sistemas.* Se instalaron tanto el sistema CWS-IM como el IM. Para esto, primero se instaló el servidor de cada uno de los sistemas. Posteriormente los sistemas clientes se instalaron en cada una de las computadoras de los participantes.
5. *Entrenamiento.* Se capacitó a los sujetos de manera individual en el uso de cada uno de los sistemas. Durante la sesión de entrenamiento se les asignó una identificación (login) para el uso del sistema.
6. *Uso del Sistema.* Se les pidió que utilizaran el sistema correspondiente durante una semana laboral. Posteriormente cambiaron de sistema durante una semana adicional.
7. *Evaluación de usabilidad y facilidad de uso.* Se les pidió al final del uso de cada uno de los sistemas que llenaran el cuestionario SUS (System Usability Scale) (ver Apéndice C) [121] y el cuestionario TAM (Technology Acceptance Model) (ver Apéndice A) [112].

7.6 Limitaciones

El caso de estudio descrito y los métodos utilizados para evaluar la herramienta propuesta, podrían haber tenido algunas debilidades. La influencia de estas debilidades en los resultados y la forma en que los abordamos son:

- *Los estudios within subject tienen el problema de aprendizaje.* Se intentó evitar este efecto mediante la interacción por subgrupos con las herramientas en orden diferente. Sin embargo, es imposible establecer si este efecto fue totalmente controlado o evitado [122].

- *La gente usualmente responde los cuestionarios con lo que ellos piensan que el investigador quiere oír. Se intentó evitar esto mediante el llenado de cuestionarios de manera anónima. La gente usualmente es más sincera en este tipo de estudios cuando no necesita dar su nombre. Además se aplicaron dos cuestionarios de usabilidad de diferentes [123].*
- *El estudio fue realizado con personal de una fábrica de software con el nivel de madurez 3 en CMMI del SEI. Los resultados podrían ser válidos sólo para fábricas de software mexicanas similares, sin embargo no es posible garantizar que estos resultados podrían ser generalizables a otros entornos de trabajo de DSD.*

7.7 Resultados

En este apartado se presentan y discuten los resultados del estudio para determinar la percepción de los desarrolladores, en términos de utilidad y facilidad de uso de la herramienta CWS-IM durante la realización de sus actividades.

Utilidad y facilidad de uso:

Puesto que partimos de que los emisores tendrán información de la actividad actual del receptor consideramos las siguientes hipótesis en cuanto a la utilidad y facilidad de uso:

H₀: No existe diferencia en la utilización de las herramientas CWS-IM y la IM tradicional.

- *H_{1a}: Los sujetos percibirán que CWS-IM es más útil que el IM durante la realización de sus actividades.*

Para abordar esta hipótesis se utilizaron los cuestionarios SUS y las primeras 6 preguntas del cuestionario TAM.

Los resultados arrojados por el cuestionario 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 $t_{01} = -2.147$ se comprueba que esta diferencia es significativa. Por lo que los resultados arrojados del SUS indican que la H1a es aceptada.

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. El cuestionario TAM utiliza una escala Likert de 1 (extremadamente en desacuerdo) a 7 (extremadamente de acuerdo), por lo que se considera un nivel aceptable de usabilidad en un rango de 25 a 42.

Las medias para el cuestionario TAM en su factor usabilidad (primeros 6 reactivos) indican una tendencia aceptable favorable para CWS-IM (media=35.31, desv. est.=4.51248) contra IM (media=24.69, desv. est.=8.81452). Para confirmar esta información se realizó la prueba t de student a nivel de significancia $p < 0.01$. Dado que $t_{01} = -4.292$ se valida que si hay diferencia significativa, lo que indica que utilizando el cuestionario TAM (usabilidad) H1a también es aceptada.

- *H1b: Los sujetos percibirán que CWS-IM es más fácil de utilizar que IM para actividades de DSD.*
- *H1c: Los sujetos del estudio percibirán que CWS-IM es más adecuado para ser adoptado que el IM tradicional en el trabajo de DSD.*

Para validar estas hipótesis se obtuvieron los datos de las últimas 6 preguntas del cuestionario TAM. El rango de aceptación para determinar si realmente se percibió que CWS-IM podría ser adoptado es de 25 a 42, puesto que TAM utiliza una escala Likert de 1 (extremadamente en desacuerdo) a 7 (extremadamente de acuerdo). Las medias para el cuestionario TAM en su factor facilidad de uso indican una tendencia aceptable muy similar entre CWS-IM (media=35.69, desv. est.= 4.55659) y IM (media=32.75, desv est. = 5.97216). Para confirmar esta información se realizó la prueba t de student a nivel de significancia $p < 0.01$. Dado que $t_{01} = -1.564$ se establece que no hay diferencia significativa, lo que nos indica que ambos sistemas (IM y CWS-IM) son sistemas percibidos como fáciles de usar y que pueden ser adoptados de manera favorable. En este caso la H1b es rechazada.

Puesto que CWS-IM es considerada más útil que IM (h1a) y además la herramienta CWS-IM es considerada fácil de usar al igual que IM, se cuenta con suficiente evidencia para aceptar H1c.

Apoyo al inicio de interacción:

Una función importante de la herramienta, es la forma en que despliega los estatus de usuario en base a la actividad actual. Puesto que CWS-IM utiliza con un código de colores para indicar la actividad actual se consideran las siguientes hipótesis en cuanto a percepción de la interrupción con el uso de los estatus basados en actividad actual:

H0: No existe diferencia en la utilización de la información proporcionada por CWS-IM para iniciar colaboración.

- *H2a: Los sujetos en posición de emisor tenderán a solicitar información adicional referente al estado del receptor (mediante el componente ToolTip de información del contacto) cuando el estado del receptor potencial esté en “diferente actividad / mismo proyecto” o “diferente proyecto” que cuando los estados estén en “misma actividad / mismo proyecto”.*

Para obtener datos y probar esta hipótesis se guardaron en una bitácora (archivo de texto) los siguientes datos: fecha, hora y acción (solicitar información de un contacto en particular y/o interactuar con un colaborador) del contacto. Esta información se recabó siempre que el usuario que estaba en posición de emisor realizara una acción sobre el listado de contactos. Cabe mencionar que la información recabada fue tomada del sistema CWS-IM. La colecta fue de 455 datos, de los cuales 354 fueron colectados en el estatus “misma actividad / mismo proyecto”, 62 en estatus “diferente actividad / mismo proyecto” y 39 en estatus de “diferente actividad / diferente proyecto”. Para interpretar los datos de la acción tomada por el usuario ésta fue codificada de la siguiente forma:

- Si la acción era recabar información del usuario se le asignaba un 1.
- Si la acción era iniciar una interacción con dicho usuario se le asignaba un 2.

Los resultados del análisis de interacción con *el componente ToolTip de información del contacto* (con un rango de 1 a 2) 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 $t_{.01} = 4.268$ se puede establecer que esta diferencia es significativa, por lo que hay evidencia suficiente para aceptar H2a. En este caso, se argumenta que los usuarios entendieron la información proporcionada por CWS-IM.

Nivel de interrupción de la herramienta:

El entendimiento de la actividad actual de los colegas de trabajo por parte del emisor, le ayuda a determinar el momento en que ambos trabajan bajo un interés común. En este caso se ha encontrado en la literatura que cuando esto es así, los receptores podrían tener más “voluntad para interactuar” [19, 41]. Por lo que consideramos las siguientes hipótesis en cuanto a percepción de la interrupción con el uso de los estatus basados en la actividad actual:

H0: No existe diferencia en la percepción de interrupción de los sujetos en posición de receptor, aún cuando los estados del receptor y del emisor se encuentren en la misma actividad y el mismo proyecto.

- *H3a: El nivel de beneficio percibido de las interacciones será más alto cuando los estados del receptor y del emisor se encuentren en la misma actividad y el mismo proyecto.*

Para esta evaluación, CWS-IM aplicaba un mini-cuestionario de manera aleatoria al atender la interacción (receptor) con las preguntas ¿qué tanto te perjudica la interacción? Y ¿qué tanto te beneficia? con una escala Likert de 1 (extremadamente en desacuerdo) a 5 (extremadamente en de acuerdo). De esta forma se guardó en una bitácora (archivo de

texto) la siguiente información: identificador, fecha, hora, nivel de perjuicio, nivel de beneficio y estatus. Esta información se recabó siempre que el sujeto estaba en posición de receptor. Para validar la hipótesis se analizaron 105 instancias de interacción. Los grupos se realizaron según el estatus en el que fueron realizadas las interacciones. Cabe mencionar que para valorar el Nivel de “Percepción de beneficio” (PB) de cada interacción por parte del receptor fue necesario aplicar la siguiente fórmula:

$$PB = \text{Beneficio-Perjuicio}$$

Para identificar diferencias significativas entre los diferentes estatus de usuario, aplicamos la prueba t de Student. El análisis de las diferencias entre los estatus “misma actividad / mismo proyecto” (media=-0.6667, desv. est. = 2.21989) y “diferente actividad / mismo proyecto” (media = -1.7647, desv. est. = 1.25147) no fue significativo ($t_{.01}=1.964$). El análisis de diferencia entre los estatus “misma actividad / mismo proyecto” (media=-0.6667, desv. est. = 2.21989) y “diferente actividad / diferente proyecto” (media=-1.2308, desv. est.= 2.08782) tampoco nos arrojó una diferencia significativa ($t_{.01}=0.853$). Finalmente entre los estatus “diferente actividad / mismo proyecto” (media = -1.7647, desv. est. = 1.25147) y “diferente actividad / diferente proyecto” (media=-1.2308, desv. est.= 2.08782) no resultó en una diferencia significativa ($t_{.01}=0.872$).

Estos resultados proporcionan evidencia para rechazar la hipótesis H2b y aceptar la hipótesis nula (H0) para esta hipótesis en particular, por lo que se puede establecer que “*No existe diferencia en la percepción de interrupción de los sujetos en posición de receptor cuando los estados del receptor y del emisor se encuentren en la misma actividad y el mismo proyecto*”.

En cuanto a percepción de la interrupción con el uso de los sistemas IM y CWS-IM.

H0: No existe diferencia en la percepción de interrupción de los sujetos en cuanto al uso del sistema.

- *H4a: Los sujetos percibirán que las interrupciones fueron más perjudiciales con el uso de IM que con CWS-IM.*

Para la evaluación de esta hipótesis utilizamos los datos del mini-cuestionario tanto los capturados por el sistema IM como los de CWS-IM. En este caso tomamos el valor arrojado para el nivel 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.

7.8 Discusión

Una vez que los resultados han sido obtenidos, hemos tratado de analizarlos y reflexionar sobre las razones que podrían explicarlos. Por lo tanto, se proporciona una "posible explicación" de cada resultado.

En cuanto a *utilidad y facilidad de uso*, para el caso de h1a se cree que los participantes percibieron a la herramienta CWS-IM más útil que el 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. Esta explicación surge de los comentarios realizados por algunos de los participantes del estudio en su entrevista preliminar, los cuales se refieren a lo que les implica organizar sus actividades o tareas asignadas. Tales comentarios son:

Cuando tengo un pendiente con mi jefe [administrador de proyecto] entonces dejo la pestaña o la ventana abierta del mensajero instantáneo, igual también utilizo en ocasiones el bloc de notas o el Excel donde estoy apuntando todos los pendientes que vayan saliendo... [Participante 6].

...fíjate que generalmente yo en particular hago anotaciones, ahí tengo mis pendientes, para mi es más fácil tenerlo no se en algún "post-it" pegado en mi escritorio,

que pendientes debo de sacar para que día o que debo de hacer primero. Eso lo hago en vez de meterme a Digité y estar revisando... [Participante 9].

...un problema que yo veo es identificar claramente a las personas, mis, mis compañeros de proyecto y tenerlos así como clasificados por lo que hacen... [Participante 7].

CWS-IM proporciona elementos para organizar las actividades asignadas de manera automática, tomando la información desde el sistema gestor de proyectos (Digité). También, con CWS-IM las actividades que son asignadas en el gestor de proyectos (Sistema Digité) 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 agrupados de usuarios que se presentan de acuerdo al proyecto y/o en el 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 (h1b) resultó que tanto el sistema IM como CWS-IM fueron percibidos como fácil de usar. Puesto que CWS-IM es considerada más útil que IM (h1a) y además CWS-IM fue percibida fácil de usar al igual que IM, se tienen elementos suficientes para afirmar que CWS-IM es una herramienta que puede ser adoptada de manera favorable en DSD (h1c).

En lo que respecta a *apoyo al inicio de interacción*, los resultados para h2a 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 [54] y la sobrecarga mental [6]. Esta explicación se presenta en base a los comentarios referentes a la atención de solicitudes o inicios de colaboración:

... es muy sencillo, si para lo que me buscan está alineado al esfuerzo que estoy realizando en ese momento, lo atiendes “de volada” (en el momento), si hay una dependencia para una entrega en ese momento también lo atiendes, simplemente si existe una dependencia... [Participante 11].

... sería más sencillo si en el mensajero solo tratara con personas del trabajo, la mayoría aquí utilizamos el mensajero, pero el público, es decir el social donde también tenemos compañeros fuera del trabajo, amigos, familiares, etc., entonces si quieres mostrarte disponible para los de tu trabajo, porque estas en el trabajo, eso no puede ser, pues es un estado para el público en general, entonces si un familiar te ve en disponible te distrae, sientes el compromiso moral de contestarle porque es tu pariente...[Participante 2].

...para mí sería bueno que si estoy en estado ausente o algo así no me lleguen muchas llamadas... [Participante 8].

En cuanto a *nivel de interrupción de la herramienta*, para el caso de h3a se considera que aún 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 [19], 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. Considerando la conclusión de [124], la cual expresa que si un cliente no es capaz o está dispuesto a participar activamente en la comunicación cuando es requerido, cualquier mecanismo de comunicación será superfluo. Esto es porque la voluntad para interactuar es considerada como un obstáculo para la coordinación entre sitios, ya que reduce de manera considerable el establecimiento de relaciones de confianza entre los miembros del grupo [41]. Dicha falta de voluntad para

interactuar está vinculada a la disposición de la colaboración, y en particular por las situaciones de trabajo entre múltiples sitios. Esto es debido a que estas interacciones solicitadas (por ser un "experto") frecuentemente no son recompensadas por las organizaciones [125]. Esta explicación es por comentarios referente al manejo de interrupciones como los siguientes:

... si tengo el trabajo encima, y estoy muy ocupada o estoy muy entretenida con lo que estoy haciendo, no lo tomo en cuenta, lo más que hago lo cierro o evito estar ahí ... [Participante 1].

...Me molesta bastante cuando estoy bien concentrada haciendo un, un documento y que tengo la idea y que de repente me hablan pues ahí, como que me atraso un poco... [Participante 7].

...depende de la persona que me está hablando igual si es mi jefe directo, o si es algún involucrado del entregable que voy o que estoy haciendo pues a lo mejor ahí yo si le doy bastante importancia... eh le doy prioridad en base a eso, si es otra persona incluso si es mi analista que me está preguntando una cosa del proyecto no lo tomo en cuenta en ese momento y lo omito, aunque me este insistiendo... [Participante 6].

... si te está hablando un Arquitecto que está a cargo de ti, entonces le tienes que dar una respuesta, no me molesta, igual y debo solucionarlo... [Participante 5].

Finalmente lo referente a interrupción según la herramienta utilizada en el estudio, los resultados para h4a 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 decidir si el momento es adecuado para iniciar colaboración, lo cual fue soportado por los resultados de las hipótesis H1a (utilidad), H1b (facilidad de uso), H1c (adopción) y H2a (entendimiento de la actividad actual).

7.9 Resumen del capítulo

En este capítulo se han presentado los resultados de un caso de estudio realizado en una fábrica de software. El estudio fue realizado para evaluar la herramienta CWS-IM en términos de:

- *Utilidad, facilidad de uso y adopción.* Se reportó evidencia favorable en cuanto a que CWS-IM es percibido como una herramienta adecuada para las actividades de DSD comparado con la mensajería instantánea tradicional.
- *iniciar colaboración.* Se reportó que los participantes entendieron de manera satisfactoria los elementos de información proporcionados por CWS-IM para iniciar colaboración de manera informada.
- *percepción de la interrupción con el uso de los estatus basados en actividad actual.* Los resultados fueron evidencia complementaria a trabajos que consideran que cuando el esfuerzo entre las partes (emisor y receptor) es de beneficio común existe más voluntad para interactuar, sin embargo esto no es una regla que indique que siempre es así. Lo que resulta en que aunque haya voluntad para interactuar el receptor podría verse perturbado o dañado con la interrupción.
- *percepción de la interrupción con el uso de los sistemas IM y CWS-IM.* Se encontró evidencia para afirmar que se percibieron como más perjudiciales los inicios de interacción con IM que con CWS-IM.

Como parte de los resultados este caso de estudio, se identificaron algunos puntos que es importante tener en cuenta para mejorar y facilitar los inicios de interacción informados con CWS-IM. Este tipo de observaciones dan pauta para definir en el siguiente capítulo las principales conclusiones, aportaciones y trabajo futuro derivados de la presente tesis.

Capítulo 8

“La puerta mejor cerrada es aquella que puede dejarse abierta”.

Proverbio chino

*“El futuro tiene muchos nombres. Para los débiles es lo inalcanzable.
Para los temerosos, lo desconocido. Para los valientes es la oportunidad”.*

Víctor Hugo

8. Conclusiones, aportaciones y trabajo futuro

En este trabajo de tesis se atiende la problemática del inicio de colaboración informada en el desarrollo distribuido de software (DSD). Esto fue abordado mediante el concepto de *esferas de trabajo colaborativas*, el cual fue caracterizado y descrito en el capítulo 4. Esto conllevó al desarrollo de un mensajero instantáneo extendido con soporte para esferas de trabajo colaborativas, con el cual fue posible realizar inicios de colaboración informados entre personas que realizan actividades de DSD. El funcionamiento de los estatus de usuario estuvo basado en Disponibilidad Selectiva, el cual se derivó del concepto de esferas de trabajo colaborativas.

Con lo anterior también fue posible realizar una prueba de concepto mediante un caso de estudio realizado en un entorno de trabajo real de DSD. Por lo que particularmente, este trabajo es valioso para equipos de trabajo virtuales, en los cuales sus integrantes tienen la necesidad de conocer la actividad actual de sus colegas y difundir entre el grupo de

trabajo información respecto a distintos tipos de disponibilidad con base en la actividad que se realiza en un determinado momento.

En las siguientes secciones se presentan las conclusiones de este trabajo, las principales aportaciones del mismo, y los principales puntos de trabajo futuro.

8.1 Conclusiones

El DSD ofrece muchos beneficios que lo hacen una forma de trabajo atractiva para la industria del software. Estos beneficios incluyen trabajo continuo en los proyectos, reducir los costos de mano de obra, y la creación rápida de equipos de desarrollo que están geográficamente cerca del cliente, entre otros.

Sin embargo, DSD también crea un conjunto de retos inherentes a sus características, incluyendo la falta de oportunidades de interacción cara a cara, gestión distribuida del conocimiento, la escasez de conciencia y relaciones de confianza entre el equipo distribuido, los cuales pueden dar lugar a malentendidos. Estos retos exigen nuevos métodos de coordinación, herramientas y procesos. Como parte de ello, este trabajo se enfocó en la necesidad de distribución de información de conciencia entre los miembros de un grupo de trabajo virtual que apoya la coordinación de actividades de DSD, particularmente en la entrada en colaboración de manera síncrona.

Para lo anterior, en 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 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 estos requerimientos e 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 auxilie al emisor a tomar una mejor de decisión respecto a cuándo y cómo abordar a un colega en un momento

determinado. En este momento se crea la propuesta que se denomina como 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.

Conforme se iba avanzando con la propuesta, se estableció la necesidad de realizar una evaluación preliminar. Dicha evaluación tuvo como objetivo validar si el diseño de una primera versión de interfaces de usuario era percibido adecuado por un grupo de desarrolladores. Para llevarla a cabo fue necesario realizar la evaluación mediante escenarios de uso donde se les presentaron escenarios actuales y escenarios proyectados con la propuesta de interfaces. La evidencia de esta evaluación preliminar mostró indicios alentadores en términos de utilidad y facilidad de uso, lo cual motivó a continuar con la implantación de la propuesta en un prototipo de software.

De lo anterior se derivó el desarrollo de un prototipo que denominamos CWS-IM, el cual es un mensajero instantáneo extendido con soporte para Esferas de Trabajo Colaborativas. El desarrollo de este prototipo está soportado por un sistema multi-agente para 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 caso de estudio en un ambiente de DSD real en una fábrica de software con sedes en Sonora, México y Arizona, E.U. Este consistió de un estudio de 3 semanas de trabajo, en el cual se obtuvo la percepción de 16 desarrolladores 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 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 (en la posición de receptor) percibieron las interrupciones utilizando CWS-IM como menos negativas que cuando utilizaron el mensajero instantáneo tradicional (MI).

8.2 Aportaciones

La principal contribución de este proyecto de investigación se centra en la conceptualización y caracterización del concepto esferas de trabajo colaborativas para el apoyo de inicios de colaboración de manera informada, la propuesta y diseño de disponibilidad selectiva, y el desarrollo y evaluación de una herramienta que denotan la factibilidad de instrumentar este tipo de soporte. Otras aportaciones derivadas de este trabajo fueron el modelo de trabajo propuesto para desarrollo de software, la caracterización de las actividades del desarrollo de software y los resultados de la evaluación preliminar. Finalmente, las publicaciones derivadas de este trabajo constituyen una parte muy importante de las contribuciones que esta tesis ha permitido obtener. A continuación se listan las principales aportaciones seguidas de la lista de publicaciones desarrolladas con base en el mismo.

1. *Un modelo de trabajo para desarrollo distribuido de software, que integra los distintos conceptos manejados por administración de proyectos de software y elementos de la actividad de desarrollo de software.* Este modelo fue construido en base a literatura de administración de proyectos e ingeniería de software. En cuanto a administración de proyectos se retomaron principalmente las publicaciones de Gido [37] y de Tobis y Tobis [36]. Estos trabajos discrepan en su forma de conceptualizar los componentes de un proyecto, ya que Gido considera que los proyectos se componen de actividades y Tobis indica que los proyectos están formados por tareas. Ambos trabajos tratan metodologías y conceptos referentes a planeación, ejecución, control y cierre de las actividades o tareas del proyecto. Por lo que este modelo integra una unidad de trabajo flexible, donde los proyectos pueden estar jerarquizados en los niveles que sea necesario, de tal manera que cualquier trabajo de desarrollo realizado por proyectos pueda adaptarse a este modelo. De esta manera el trabajo de Gido y el de Tobis y Tobis pueden ser adaptados al modelo e integrarse como parte de las instancias y sus relaciones. Con respecto a la ingeniería de

software se encontraron metodologías para el proceso de desarrollo de software, las cuales proporcionaron el establecimiento y confirmación de algunas relaciones entre instancias del modelo sobre todo en la parte operativa o ejecución del modelo (ver Sección 2.2.2). Pressman [11] presenta el ciclo de vida de desarrollo de software donde ilustra las fases o pasos para que el software sea creado. Con esto se incluyó la parte dinámica del proceso de trabajo al modelo mediante la instancia operación. Cabe mencionar que esta instancia en particular tuvo que ser relacionada con la instancia estado, la cual fue retomada de Morán et al. [126] para indicar el avance de una unidad de trabajo desde la operación.

Como se puede notar este modelo aporta la visión integrada de las instancias o elementos que se involucran en el trabajo de desarrollo de software. Por lo que este modelo provee una visión de gestión de proyectos y de proceso de trabajo. En cambio, los autores de administración de proyectos [36, 37] se refieren en gran parte en sus trabajos a proporcionar buenas prácticas o recomendaciones a seguir para que el proyecto llegue a buen término. Pressman [11] por su parte ofrece una visión práctica del proceso y el software como producto. Es por ello que con este modelo se logra obtener argumentos suficientes para la comprensión de la especificación, administración y operación de actividades en el trabajo de desarrollo de software. Esta aportación puede ser considerada como parte complementaria a las literaturas tanto de administración de proyectos como de ingeniería de software.

2. *Definición de un conjunto de hallazgos a considerar para lograr inicios de interacción informados en DSD.* Para determinar estos hallazgos se analizaron trabajos de los campos de estudio de manejo de interrupciones y desarrollo distribuido de software (DSD). Estos trabajos describen las principales aproximaciones disponibles en la literatura y que revela cómo la investigación en estas áreas aportan ideas de diseño para iniciar colaboración. Los trabajos de manejo de interrupciones aportan en cuanto a

la detección del tipo de actividad o tarea que motiva la colaboración, la relevancia que tiene para los involucrados de un grupo de trabajo y las formas de mitigar el efecto negativo de una interrupción antes o al momento de iniciar la interacción. En el análisis presentado por Cutrell et al. [5] demuestra que para que la interrupción sea percibida menos negativa o dañina a la actividad es importante determinar qué relevancia tendrá el motivo de la interacción para la persona que es buscada. En este sentido Mockus y Herbsleb [82] presentan una herramienta para localizar expertos dentro de un grupo de trabajo. Esta herramienta basa el conocimiento de los expertos en la información del sistema de administración de cambios. Con esta herramienta se beneficia en buena manera a la persona que desea iniciar colaboración, por lo que resuelve de cierta manera el escoger a la persona más indicada para colaborar. Asimismo Ye [19] argumenta que al contactar a un colega (receptor) para tratar una tarea de interés compartido, existe más voluntad por parte del receptor para que ésta se lleve a cabo. Por lo que el argumento de Ye busca beneficiar la parte receptora para colaborar. También es importante considerar la propuesta de Bailey y Iqbal [6], que indican que las interrupciones pueden ser consideradas relevantes o importantes para la persona que es contactada (receptor) siempre y cuando el motivo de dicha interrupción vaya en función a la tarea que realiza en ese momento. La integración de las perspectivas de estos autores contribuyen a los inicios de colaboración informados con ideas de diseño que beneficie a ambas partes de la interacción (emisor y receptor).

En los trabajos referentes a DSD se encontró que el uso de elementos de conciencia ayudan a mantener informado a un grupo en cuanto a la actividad que realizan. En este caso el sistema YooHoo [75] utiliza de manera interesante las notificaciones basada en intervalos de tiempo personalizados para informar los eventos relevantes realizados por los integrantes del grupo. El sistema ProjectWatcher [69] indica las acciones que realiza el desarrollador sobre los artefactos. De esta forma ambos sistemas informan al grupo de trabajo lo que sucede en torno al trabajo de

desarrollo. Sin embargo estos trabajos y los citados en la Sección 3.2.2 ofrecen informar al grupo de manera rígida. Es decir, la distribución de información de conciencia, da a conocer el mismo estatus de trabajo o actividad al resto del grupo, sin tomar en cuenta aspectos como los propuestos por Bailey y Iqbal [6], ya que esto ayudaría a informar la tarea que realiza el receptor en determinado momento para motivar la interacción si fuera necesario. Esto es debido a que las herramientas citadas fueron creadas para fines distintos al de iniciar colaboración.

Por lo tanto, estos hallazgos aportan claramente ideas de diseño para la creación de 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.

3. *Caracterización de las actividades de desarrollo de software.* Para crear esta caracterización se analizaron trabajos de investigación referentes a ingeniería de software. Estos trabajos proporcionaron aspectos con respecto al producto y al proceso de desarrollo de software. Para realizar esta caracterización se retomó el trabajo de Kraut y Streeter [2], el cual presenta las características del desarrollo de software con la finalidad de hacer notar los problemas de coordinación. Por ello aspectos como escala, manejo de riesgos o incertidumbre, dependencia de actividades y comunicación fueron importantes para argumentar la caracterización de este tipo de actividad. Es por ello que para complementar la caracterización se agregó a la descripción de cada característica propuestas por Kraut y Streeter elementos que componen este tipo de actividad. Para el caso de la característica de escala se integraron aspectos como el substrato social de manera similar como lo proponen Munir y Lorne [108] para los requerimientos de groupware. También se incluyó distribución geográfica para considerar si el desarrollo era realizado de manera colocalizada o distribuida como lo argumenta Prikladnicki et al. [14]. Finalmente en escala se agregó el esfuerzo para denotar la magnitud del proyecto pero desde el punto de vista de la duración

de su desarrollo de manera similar como lo proponen Kraut y Streeter. Para la característica de incertidumbre también se retomó el argumento de de Kraut y Streeter y además se fusionó con la descripción que proponen Stinchcombe y Heimer para expresar la poca certeza que pueden tener los trabajadores respecto a conocer el avance real de una unidad de trabajo del proyecto. Para la característica de interdependencia, se agregó la propuesta de McChesney [34], la cual explica las razones por las que se presentan las actividades interdependientes. Finalmente para la característica de comunicación se agregan los tipos de comunicación (formal e informal) descritos por Kraut et al. [97] y Isaacs et al. [18].

Como se puede notar, esta caracterización de la actividad resulta ser complementaria a los trabajos de ingeniería de software, pues se agrega una perspectiva diferente a la presentada por Pressman [11] para describir la naturaleza del software y su ciclo de vida.

4. *Definición de un conjunto de requerimientos para el logro de inicio de interacciones informados en DSD.* Para obtener estos requerimientos, las áreas de investigación interacción humano computadora y sistemas colaborativos fueron muy importantes. Esto fue porque en interacción humano computadora se encontraron los aspectos involucrados con el manejo personal de actividades. Particularmente del trabajo referente a esfera de trabajo propuesto por González y Mark [46], se retomaron elementos para entender la actividad de los desarrolladores desde su espacio de trabajo. Esta propuesta contribuyó a la identificación de áreas de oportunidad para iniciar colaboración desde la perspectiva del trabajo individual.

En el área de sistemas colaborativos se identificaron elementos para colaboración en grupo, de manera particular se retomó la propuesta colaboración potencial de Morán [111]. Con esto es posible descubrir o crear oportunidades para colaborar en momentos apropiados.

Tomando como fundamento los trabajos antes mencionados, se contribuyó a la creación de requerimientos para inicios de colaboración informados desde la perspectiva de la actividad individual [46] y la colaboración actual y potencial [111], los cuales apoyan a la comprensión de cómo éstos deben ser llevados a cabo en ambientes de DSD y además proporcionan una base para la creación de ideas de diseño para desarrollar sistemas que soporten inicios de colaboración informados.

5. *Caracterización y modelado de Esferas de Trabajo Colaborativas.* Las esferas de trabajo propuestas por González y Mark [46] proporcionaron elementos relacionados con la actividad individual que pueden ser utilizados para iniciar colaboración desde la perspectiva del manejo de las unidades de trabajo. Para poder monitorizar la esfera de trabajo de cada individuo agregaron elementos de colaboración potencial [117], para descubrir o crear oportunidades de interacción en momentos oportunos o apropiados. Las esferas de trabajo poseen una visión personal de actividades pero existen elementos en la esfera (p. ej. artefacto en uso, tipo de artefacto, objetivo, etc.) que son de utilidad para los colegas de trabajo. Para ello se integraron elementos de colaboración potencial, con lo cual se distribuye información de la esfera a los colegas. Con esto se le ofrece al iniciador (antes o al momento de iniciar colaboración) información acerca de la esfera de trabajo en que se encuentra el colega que se quiere contactar. Con esto se beneficia al receptor de la interacción, pues el iniciador obtiene elementos de información que le indican que tan relevante es lo que quiere tratar con su colega en ese momento de manera similar como lo propone Ye [19] con el argumento de detectar el momento con más voluntad para interactuar. Con esta fusión de elementos se logra obtener la caracterización y modelado de un sistema tecnológico para soportar inicios de colaboración informados, lo cual fue parte importante para lograr el objetivo general de esta investigación. Asimismo 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. Yohoo, ProjectWatchet, etc.), 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.).

6. *El resultado de la preferencia del usuario con respecto a la funcionalidad de las esferas de trabajo colaborativas.* Previo al desarrollo de una herramienta prototipo para soportar esferas de trabajo colaborativas, fue necesario validar las interfaces de usuario propuestas y su función en el DSD. Por lo que esta validación se realizó mediante la creación de escenarios de uso como son propuestos por Carroll [20]. Estos resultados del caso de estudio proporcionaron indicios favorables de aceptación en cuanto a funcionalidad y utilidad de las Esferas de Trabajo Colaborativas en DSD. Además de esto, se obtuvo un conjunto de modificaciones a la propuesta que contribuyeron a mejorar el diseño de los inicios de interacción informados dentro de este tipo de ambientes de trabajo. Con estos resultados se confirmaron elementos de diseño y se agregaron otros para contribuir en buena manera al desarrollo de un prototipo funcional. La preferencia de los participantes por utilizar la propuesta en DSD aportan nuevas formas por mitigar el efecto negativo de la interrupción en el área de manejo de interrupciones. Estos es porque a diferencia de la propuesta de la detección de la carga mental propuesta por Bailey y Iqbal [6], los resultados de percepción aquí propuestos son en base a un sistema que ayuda a un grupo de trabajo a interrumpir en momentos oportunos y no en determinar características que pudieran servir para ello. De tal manera que estos resultados ya incluyen una propuesta que le ayuda al desarrollador a determinar el momento apropiado para abordar a un colega.

7. *Un prototipo basado en la funcionalidad de la mensajería instantánea con soporte para Esferas de Trabajo Colaborativas (CWS-IM).* El desarrollo de este prototipo fue soportado la caracterización de las esferas de trabajo colaborativas, con lo cual se aportaron implicaciones de diseño para iniciar colaboración de manera informada. Además se consideraron los resultados del estudio preliminar para redefinir los mecanismos y servicios que proporciona el prototipo. A diferencia de la herramienta ProjectView-IM [107] que integra en el diseño de un mensajero instantáneo que facilita el trabajo en múltiples proyectos, CWS-IM ofrece elementos de información relacionados con la actividad de los colegas en un tiempo dado. Además CWS-IM proporciona los estatus de usuario en base al criterio de Disponibilidad Selectiva descrito en la Sección 4.3.6. con el cual los usuarios pueden determinar que tanto puede interesarle al colega el motivo de la interacción como sugiere Ye [19]. Además CWS-IM distribuye elementos de información de la actividad que realizan los colegas lo cual ayuda también a disminuir la incertidumbre de lo que hacen los colegas [38]. Este prototipo en buena medida fue parte esencial para la evaluación del caso de estudio en un ambiente de trabajo real, el cual se integró con información del sistema gestor de proyectos. Este prototipo contribuye de manera diferente a las herramientas propuestas para realizar actividades de coordinación y comunicación en DSD (ver Sección 3.2), pues no se encontraron herramientas o propuestas que promuevan los inicios de colaboración informados.

8. *Evaluación del prototipo en un caso de estudio en el ambiente de DSD.* Este caso de estudio permitió evaluar el prototipo en un ambiente real de DSD por distintos roles de trabajo. Con respecto a trabajos de comunicación, Raghvinder y Johannes [62] realizan un estudio para promover la poca interacción entre los sitios mediante la distribución de información referente a las modificaciones realizadas a un diseño de software, pero su estudio demanda facilitar y gestionar tareas o actividades mediante un rol de trabajo

especial. Las herramientas propuestas para DSD como WikiWiki [88], OntoWiki [127], CodeSaw [85], entre otros no reportan resultados de percepción de trabajadores de DSD ni tampoco se reporta si estos han sido evaluados en un caso real. También los estudios de DSD han estado contribuyendo en la parte de coordinación como el trabajo de Cataldo et al. [25], Wiredu [13], Tiwana [81], en el cual identifican un conjunto de problemas de coordinación y presentan una serie de implicaciones de diseño que deben tomarse en cuenta para desarrollar una herramienta en apoyo a este tipo de actividad.

Esta evaluación es relevante pues particularmente en la literatura de DSD no se ha reportado el estudio de un prototipo en ambiente real como el que se describe en el Capítulo 7, pues la mayoría de ellos presentan casos de estudio en ambientes controlados en el mejor de los casos. Por lo que este caso de estudio puede ser considerado como un conjunto inicial de resultados que pueden servir como una base a partir de la cual otros investigadores comenzarán a explorar más a fondo este tipo de temas.

Un resultado importante de este trabajo de investigación, lo constituyen las publicaciones realizadas. A continuación se enlistan las referencias a los trabajos publicados.

Publicación en revistas:

Publicadas

1. Ramón R. Palacio, Alberto L. Morán, Víctor M. González, “CWS: An Awareness Tool to Support Starting Collaboration in Global Software Development”. The Open Software Engineering Journal. Special Issue on Global Software Development and its Challenges. ISSN: 1874-107X - Volume 3, 2010.
2. Ramón R. Palacio, Aurora Vizacino, Alberto L. Morán, Víctor M. González, “A tool to facilitate appropriate communication in Global Software Development”. IET Software. (Aceptada para publicación).

Publicación en congresos internacionales:

1. Ramón R. Palacio, Alberto L. Morán, Víctor M. González, Aurora Vizcaíno, "Providing Support for Starting Collaboration in Distributed Software Development: A Multi-agent Approach," CSIE, vol. 7, pp.397-401, 2009 WRI World Congress on Computer Science and Information Engineering, 2009. IEEE Press ISBN: 978-0-7695-3507-4. Los Angeles, USA, March 31 - April 2, 2009.
2. Ramon R. Palacio, Alberto L. Moran, Victor M. Gonzalez, Aurora Vizcaino, "Collaborative Working Spheres as support for starting collaboration in distributed software development," CSCWD, pp.636-641, 2009 13th International Conference on Computer Supported Cooperative Work in Design, 2009. IEEE Press ISBN: 978-1-4244-3534-0. Santiago, Chile April 22-April 24.
3. Gabriela Aranda, Aurora Vizcaíno, Ramón R. Palacio, Alberto L. Morán, "What Information would you like to know about your Co-worker?: A Case Study", ICGSE, 2010, the IEEE International Conference on Global Software Engineering. Princeton, USA August 23-26 (Accepted for publication).

Publicación en congresos nacionales:

1. Palacio R., Morán A., González V., "Soporte para la entrada en colaboración en el desarrollo distribuido de software: implicaciones de diseño", 9o Encuentro Internacional Mexicano de Ciencias de la Computación, Simposio de Ingeniería de Software 2008, Ensenada, BC México, 6-10 Octubre de 2008, Mexicali, BC México.
2. Palacio Ramón R., "Soporte a la entrada en colaboración en el desarrollo distribuido de software mediante esferas de trabajo colaborativas", 9o Encuentro Internacional Mexicano de Ciencias de la Computación, Consorcio Doctoral, Mexicali, BC México, 6-10 Octubre de 2008.

Es importante mencionar, que la funcionalidad proporcionada por CWS-IM, aunque fue bien percibida e interpretada por los usuarios potenciales, no significa que no pueda ser mejorada en más aspectos referentes al DSD. Por ello, en la siguiente sección se presentan

y describen un conjunto de propuestas de investigación para trabajo futuro que surgen a partir de esta tesis.

8.3 Trabajo futuro

Durante la realización de este trabajo de tesis surgieron algunas observaciones o ideas que podrían haber mejorado algunos aspectos de la tesis, o que a su vez proponían enfoques diferentes a nuestra línea de investigación. No obstante dichas ideas resultaban interesantes para profundizarse y no pudieron ser atendidas debido al alcance de esta tesis. Estas ideas las utilizamos para describir tres líneas de trabajo futuro:

- *Funcionalidad del prototipo CWS-IM y el concepto esferas de trabajo colaborativas.* Aunque la herramienta fue bien recibida por un grupo de usuarios potenciales, se identificaron aspectos para mejorar la usabilidad y utilidad del prototipo. Por lo que se propone incluirle mecanismos que permitan al emisor estar informado de eventos importantes para el receptor (p. ej. el receptor potencial está a punto de dejar el trabajo después de terminar su jornada laboral, o que el receptor está a punto de entrar a una reunión de trabajo, o incluso interrumpir una jornada de trabajo por una cuestión personal/familiar importante para el receptor sin dárselo a conocer al emisor). Un caso especial es la de incluir información sobre la jornada de trabajo donde se traslapan los horarios de trabajo de los miembros del equipo, los cuales se encuentran en distintos puntos geográficos con diferencias importantes entre sus respectiva hora local. También pueden agregarse información o algún distintivo que relacione al individuo con su cultura para tener en cuenta cuales son los valores y conductas del contacto (p. ej. conocer si la persona es de cultura colectivista o individualista). Además de lo anterior, se plantea la idea de integrar el concepto de Esferas de Trabajo Colaborativas a una red social vía web para trabajadores de DSD, en la cual se puedan incluir y/o adaptar los mecanismos y servicios propuestos para esferas de trabajo colaborativas. Con esto se pueden complementar los beneficios que ofrece CWS-IM al manejar comunicación síncrona y asíncrona con el grupo de trabajo, y porque se incluiría

información más detallada de la persona (contacto) o su perfil de trabajo (p. ej. conocer su nivel de experto, intereses personales, etc.).

- *Estudio de usabilidad de CWS-IM por desarrolladores co-localizados y distribuidos.* Se plantea realizar un estudio que permita investigar si hay diferencia en la percepción de la utilidad y facilidad de uso de CWS-IM entre desarrolladores co-localizados y distribuidos de un mismo equipo de trabajo, ya que esto pudieran generar resultados para conocer el efecto de la proximidad artificial [8] con el nivel de interacción mediante el uso de la herramienta. Además se propone considerar un mecanismo que permita medir el impacto de la interrupción antes de que el receptor conozca el motivo de la interrupción y después de conocer dicho motivo. Esto daría una mejor perspectiva para determinar si la información proporcionada por CWS-IM es motivo suficiente para darle valor a la interacción.
- *Manejo personalizado de las interrupciones con CWS-IM.* Durante la evaluación de la herramienta en el caso se estudio, se observó que hay aspectos como la personalidad que pudieran considerarse para manejar de manera más personalizada la preferencia para ser abordado por un compañero del grupo. Se propone como trabajo futuro incluir elementos en la interfaz gráfica de usuario (GUI) para personalizar las interrupciones, lo cual permita al receptor configurar o programar manualmente su estado de disponibilidad, independientemente de su actividad actual (p. ej. en determinado horario estar disponible sólo para el administrador del proyecto, y no disponible para todos los demás, incluidos los que realizan la misma actividad en el mismo proyecto). También se propone manejar interrupciones mediante el reconocimiento de los rasgos de personalidad de los individuos. Por lo que surge la idea de agregar al prototipo un mecanismo que reconozca rasgos de personalidad para determinar la preferencia para ser abordado. Por ello se propone como trabajo futuro identificar, desarrollar y evaluar mecanismos que permitan medir rasgos de personalidad mediante la forma en que usan el equipo de cómputo.

Referencias

- [1] J.D. Herbsleb and A. Mockus, "An Empirical Study of Speed and Communication in Globally Distributed Software Development," *Software Engineering*, vol. 29, no. 6, 2003, pp. 481-494.
- [2] R. E. Kraut and L.A. Streeter, "Coordination in software development," *Commun. ACM* vol. 38, no. 3, 1995, pp. 69-81.
- [3] D. Damian and D. Moitra, "Guest editors' Introduction: Global Software Development: How far have we come?," *IEEE Software*, vol. 23, no. 5, 2006, pp. 17-19.
- [4] U. Bellur, "An academic perspective on globalization in the software industry," *Proc. The 30th annual international computer software and applications conference (COMPSAC'06)*, 2006, pp. 52-54.
- [5] E. B. Cutrell, M. Czerwinski and E. Horvitz, "Effects of instant messaging interruptions on computing tasks," *Proc. CHI '00 Extended Abstracts on Human Factors in Computing Systems* ACM, 2000, pp. 99-100.
- [6] B.P. Bailey and S.T. Iqbal, "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, 2008, pp. 1-28; DOI <http://doi.acm.org/10.1145/1314683.1314689>.
- [7] M. Czerwinski, E. Horvitz and S. Wilhite, "A diary study of task switching and interruptions," *Proc. the SIGCHI Conference on Human Factors in Computing Systems*, 2004, pp. 175-182.
- [8] R. Kraut, C. Egidio and J. Galegher, "Patterns of contact and communication in scientific research collaboration," *Proc. Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, ACM, 1988.
- [9] G. Calvary, J. Coutaz and L. Nigay, "From single-user architectural design to PAC*: a generic software architecture model for CSCW," *Proc. Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 1997.
- [10] C. A. Ellis, S.J. Gibbs and G. Rein, "Groupware: some issues and experiences," *Commun. ACM*, vol. 34, no. 1, 1991, pp. 39-58; DOI <http://doi.acm.org/10.1145/99977.99987>.
- [11] R.S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw Hill, 2005, p. 873.

- [12] L. D.Panjer, D. Damian and M. Storey, "Cooperation and coordination concerns in a distributed software development project," *Proc. The 2008 international Workshop on Cooperative and Human Aspects of Software Engineering* ACM Press, 2008, pp. 77-80.
- [13] G.O. Wiredu, "A framework for the analysis of coordination in global software development," *Proc. The 2006 international Workshop on Global Software Development For the Practitioner* 2006, pp. 38-44.
- [14] R. Prikładnicki, J.L.N. Audy and J.R. Evaristo, "Distributed Software Development: Toward an Understanding of the Relationship between Project Team, Users and Customers," *Proc. 5th Int'l Conf. Enterprise Information Systems (ICEIS 03)*, 2003, pp. 417-423.
- [15] L. Dabbish and R.E. Kraut, "Controlling interruptions: awareness displays and social motivation for coordination," *Proc. The 2004 ACM Conference on Computer Supported Cooperative Work* 2004, pp. 182-191.
- [16] J. D. Herbsleb, A. Mockus, T. A. Finholt and R.E. Grinter, "Distance, dependencies, and delay in a global collaboration," *Proc. Proceedings of the 2000 ACM conference on Computer supported cooperative work*, ACM, 2000.
- [17] L. Layman, L . Williams, D. Damian and H. Bures, "Essential communication practices for Extreme Programming in a global software development team," *Information and software technology*, vol. 48, no. 9, 2006, pp. 781-794.
- [18] E. Isaacs, S. Whittaker, D. Frohlich and B. O'Conaill, *Informal Communication Re-examined: New Functions For Video in Supporting Opportunistic Encounters*, Lawrence Erlbaum, 1997.
- [19] Y. Ye, "Supporting software development as knowledge-intensive and collaborative activity," *Proc. Proceedings of the 2006 international workshop on Workshop on interdisciplinary software engineering research*, ACM, 2006.
- [20] J. Carroll, *Making Use*, MIT Press, 2000.
- [21] A. Mockus and J. Herbsleb, "Challenges of Global Software Development," *Proc. Seventh International Software Metrics Symposium, METRICS 2001*, 2001, pp. 182-184.
- [22] I. Richardson, V. D. Casey, D. Zage and W. Zage, *Global Software Development – The challenges* University of Limerick, Ball State University, 2005.
- [23] D. Damian and D. Zowghi, "The Impact of Stakeholders'? Geographical Distribution on Managing Requirements in a Multi-Site Organization," *Proc. The 10th Anniversary IEEE Joint international Conference on Requirements Engineering*, 2002, pp. 319-330.

- [24] C. Ebert and P.D. Neve, "Surviving Global Software Development," *IEEE Software*, vol. 18, 2001, pp. 62-69.
- [25] M. Cataldo, M. Bass, J. D. Herbsleb and L. Bass, "On Coordination Mechanisms in Global Software Development," *Proc. The international Conference on Global Software Engineering* IEEE Computer Society, 2007, pp. 71-80.
- [26] B. Bruegge, A. H. Dutoit and T. Wolf, "Sysiphus: enabling informal collaboration in global software development," *Proc. IEEE International Conference on Global Software Engineering (ICGSE '06)*, 2006, pp. 139-148.
- [27] W. Lloyd, M. Rosson and J. Arthur, "Effectiveness of elicitation techniques in distributed requirements engineering," *Proc. The 10th anniversary IEEE Joint International Conference on Requirements Engineering, RE'02*, 2002, pp. 311-318.
- [28] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," *IEEE Software*, vol. 18, no. 2, 2001, pp. 22-29.
- [29] J. Herbsleb and D. Moitra, "Guest Eds.' Introduction: Global Software Development," *IEEE Software*, vol. 18, no. 2, 2001, pp. 16-20.
- [30] E. Conchúir, H. Holmstrom, P. Agerfalk and B.Fitzgerald, "Exploring the assumed benefits of global software development," *Proc. The IEEE International Conference on Global Software Engineering (ICGSE'06)*, 2006, pp. 159-168.
- [31] D. Damián, "Stakeholders in global requirements engineering: Lessons learned from practice," *IEEE Software*, vol. 24, 2007, pp. 21-27.
- [32] M. Babar, J. Verner and P. Nguyen, "Establishing and maintaining trust in software outsourcing relationships: An empirical investigation," *Systems and Software*, vol. 80, no. 9, 2007, pp. 1438-1449.
- [33] A. Boden, A. Nett and V. Wulf, "Coordination Practices in Distributed Software Development of Small Enterprises," *Book Coordination Practices in Distributed Software Development of Small Enterprises*, Series Coordination Practices in Distributed Software Development of Small Enterprises, ed., Editor ed.^eds., IEEE Computer Society, 2007, pp.
- [34] I. McChesney, "Effective coordination in the software process - historical perspectives and future directions," *Software Quality Control*, vol. 6, no. 3, 1997, pp. 235-246.
- [35] R.S. Pressman, *Ingeniería del Software Un enfoque Práctico*, McGraw Hill, 2001.
- [36] Michael Tobis and I. Tobis, *Managing Multiple Projects*, McGraw-Hill, 2002.

- [37] Jack Gido and C. James, *Administración exitosa de proyectos*, Thomson 2003.
- [38] A. Stinchcombe and C. Heimer, *Organization Theory and Project Management: Administering Uncertainty in Norwegian Offshore Oil*, A Scandinavian University Press Publication, 1986.
- [39] H. Spanjers, M. Huurne, B. Graaf, M. Lormans, D. Bendas and R. Solingen, "Tool Support for Distributed Software Engineering," *Proc. Global Software Engineering, 2006. ICGSE '06. International Conference on*, 2006, pp. 187-198.
- [40] I. Oshri, J. Kotlarsky and L.P. Willcocks, "Global software development: Exploring socialization and face-to-face meetings in distributed strategic projects," *J. Strateg. Inf. Syst.*, vol. 16, no. 1, 2007, pp. 25-49; DOI <http://dx.doi.org/10.1016/j.jsis.2007.01.001>.
- [41] P. J. Ågerfalk, B. Fitzgerald, H. Holmström, B. Lings, B. Lundell and E.Ó. Conchúir, "A Framework for Considering Opportunities and Threats in Distributed Software Development," *Proc. the International Workshop on Distributed Software Development (DiSD 2005)*, Austrian Computer Society, 2005, pp. 47-61.
- [42] J. Gibson, J. Ivancevich and J. Donnelly, *Las Organizaciones*, McGraHill, 2003.
- [43] A. Strauss, "Work and the division of labor," *The Sociological Quarterly*, vol. 26, no. 1, 1985, pp. 1-19.
- [44] Gerrit van der Veer and M.v. Welie, "Task based groupware design: putting theory into practice," *Proc. 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, 2000, pp. 326-337.
- [45] Alberto Morán, Jesús Favela, Ana Martinez and D. Decouchant, "Document Presence Notification Services for Collaborative Writing," *Proc. Seventh International Workshop on Groupware*, 2001, pp. 125-135.
- [46] V. Gonzalez and G. Mark, "Constant, Constant, Multi-tasking Crazy: Managing Multiple Working Spheres," *Proc. The SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, 2004, pp. 2004.
- [47] A. B. Brush, B. R. Meyers, D. S. Tan and M. Czerwinski., "Understanding memory triggers for task tracking," *Proc. SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2007, pp. 947-950.
- [48] V. González, L. Galicia and J. Favela, "Supporting the Planning and Organization of Multiple Activities in the Workplace," *Proc. INTERACT 2007*, 2007, pp. 235-238.
- [49] A. Strauss, "The articulation of project work: An organizational process.," *The Sociological Quarterly*, vol. 29, no. 2, 1988, pp. 163-178.

- [50] Andrew Monk and S. Howard, "Methods & tools: the rich picture: a tool for reasoning about work context," *interactions*, vol. 5, no. 2, 1998, pp. 21-30; DOI <http://doi.acm.org/10.1145/274430.274434>.
- [51] G. Mark, D. Gudith and U. Klocke, "The cost of interrupted work: more speed and stress," *Proc. The Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, 2008, pp. 107-110.
- [52] P. D. Adamczyk, S. T. Iqbal and B.P. Bailey, "A method, system, and tools for intelligent interruption management," *Proc. Proceedings of the 4th international workshop on Task models and diagrams*, ACM, 2005.
- [53] C. Speier, I. Vessey and J.S. Valacich, "The effects of interruptions, task complexity, and information presentation on computer-supported decision-making performance," *Decision Sciences*, vol. 34, no. 4, 2003, pp. 771-797.
- [54] J. Ellis and L. Kvavilashvili, "Prospective memory in 2000: Past, present and future directions," *Applied Cognitive Psychology*, vol. 14, 2000, pp. 1-9.
- [55] E. Ó. Conchúir, P. J. Ågerfalk, H. H. Olsson and B. Fitzgerald, "Global software development: where are the benefits?," *Communication ACM*, vol. 58, no. 2, 2009, pp. 127-131.
- [56] S. Cherry and P.N. Robillard, "Communication Problems in Global Software Development: Spotlight on a New Field of Investigation," *Proc. Int'l Workshop on Global Software Development, International Conference on Software Engineering (ICSE 2004)*, 2004, pp. 48-52.
- [57] M. Gloria, V. M. Gonzalez and J. Harris, "No task left behind?: examining the nature of fragmented work," *Proc. Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 2005.
- [58] B. P. Bailey and J.A. Konstan, "On the need for attention-aware systems: Measuring effects of interruption on task performance error rate and affective state," *Computers in Human Behavior*, vol. 22, no. 4, 2006, pp. 685-708.
- [59] D. McFarlane, "Comparison of four primary methods for coordinating the interruption of people in human-computer interaction," *Hum.-Comput. Interact.*, vol. 17, no. 1, 2002, pp. 63-139; DOI http://dx.doi.org/10.1207/S15327051HCI1701_2.
- [60] E. Horvitz, C. Kadie, T. Paek and D. Hovel, "Models of attention in computing and communication: from principles to applications," *Commun. ACM*, vol. 46, no. 3, 2003, pp. 52-59; DOI <http://doi.acm.org/10.1145/636772.636798>.

- [61] J. Gluck, A. Bunt and J. McGrenere, "Matching attentional draw with utility in interruption," *Book Matching attentional draw with utility in interruption*, Series Matching attentional draw with utility in interruption, ed., Editor ed.^eds., ACM, 2007, pp.
- [62] S.S. Raghvinder and R. Johannes, "Architecture leadership and management in globally distributed software development," *Book Architecture leadership and management in globally distributed software development*, Series Architecture leadership and management in globally distributed software development, ed., Editor ed.^eds., ACM, 2008, pp.
- [63] M. Cataldo and J.D. Herbsleb, "Communication networks in geographically distributed software development," *Proc. Proceedings of the 2008 ACM conference on Computer supported cooperative work*, ACM, 2008.
- [64] A. Repenning, A. Ioannidou, M. Payton, W. Ye and J. Roschelle, "Using components for rapid distributed software development.," *IEEE Software*, vol. 18, no. 2, 2001, pp. 38-45.
- [65] P. Dourish and V. Bellotti, "Awareness and Coordination in Shared Workspaces.," *Proc. the ACM Conference on Computer-Supported Cooperative Work CSCW'92*, New York: ACM, 1992, pp. 107-114.
- [66] D. Damian, J. Chisan, P. Allen and B. Corrie, "Awareness meets requirements management: Awareness needs in global software development.," *Proc. Proceedings of the International Workshop on Global Software Development 2003*, 2003.
- [67] R. Kobylinski, O. Creighton, A. Dutoit and B. Bruegge, "Building Awareness in Global Software Engineering: Using Issues as Context," *Proc. International Workshop on Distributed Software Development (GSD'02)*, 2002.
- [68] A. Sarma and v.A.d. Hoek, "Palantír: Increasing Awareness in Distributed Software Development," *Proc. Workshop on Global Software Development (ICSE 2002)*, 2002, pp. 28-32.
- [69] K. A. Schneider, C. Gutwin, R. Penner and D. Paquette, "Mining a software developer's local interaction history," *Proc. 1st International Workshop on Mining Software Repositories (MSR'04)*, 2004.
- [70] K. Elliot, C. Neustaedter and S. Greenberg, *Sticky Spots: A Location-Based Messaging System for the Home*, 2006.
- [71] H. Schnädelbach, A. Penn, P. Steadman, S. Benford, B. Koleva and T. Rodden, "Moving office: inhabiting a dynamic building," *Proc. Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, ACM, 2006.

- [72] N. Romero, G. McEwan and S. Greenberg, "A field study of community bar: (mis)-matches between theory and practice," *Proc. Proceedings of the 2007 international ACM conference on Supporting group work*, ACM, 2007.
- [73] J. T. Biehl, M. Czerwinski, G. Smith and G.G. Robertson, "FASTDash: a visual dashboard for fostering awareness in software teams," *Proc. Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 2007.
- [74] J. T. Biehl, W. T. Baker, B. P. Bailey, D. S. Tan, K. M. Inkpen and M. Czerwinski, "Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development," *Proc. Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ACM, 2008.
- [75] R. Holmes and R.J. Walker, "Promoting developer-specific awareness," *Proc. Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, ACM, 2008.
- [76] M. Mamei and F. Zambonelli, "Programming pervasive and mobile computing applications: The TOTA approach," *ACM Trans. Softw. Eng. Methodol.*, vol. 18, no. 4, 2009, pp. 1-56; DOI <http://doi.acm.org/10.1145/1538942.1538945>.
- [77] J. A. Espinosa, S. A. Slaughter, R. Kraut and J.D. Herbsleb, "Team knowledge and coordination in geographically distributed software development," *Journal of Management Information Systems*, vol. 24, no. 1, 2007, pp. 135-169.
- [78] M. Cataldo and J.D. Herbsleb, "Communication patterns in geographically distributed software development and engineers' contributions to the development effort," *Proc. Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, ACM, 2008.
- [79] P. Ovaska, M. Rossi and P. Marttiin, "Architecture as a coordination tool in multi-site software development," *Software Process: Improvement and Practice*, vol. 8, no. 4, 2003, pp. 233-247.
- [80] M. Paasivaara and C. Lassenius, "Collaboration Practices in Global Inter-Organizational Software Development Projects," *J. Software Process: Improvement and Practice, special issue on global software development*, vol. 8, no. 4, 2003, pp. 183-200.
- [81] A. Tiwana, "Impact of classes of development coordination tools on software development performance: A multinational empirical study," *ACM Trans. Softw. Eng. Methodol.*, vol. 17, no. 2, 2008, pp. 1-47.

- [82] A. Mockus and J.D. Herbsleb, "Expertise browser: a quantitative approach to identifying expertise," *Proc. Proceedings of the 24th International Conference on Software Engineering*, ACM, 2002.
- [83] S.E. Dossick and G.E. Kaiser, "CHIME: A Metadata-Based Distributed Software Development Environment," *Proc. 7th European Software Engineering Conference and 7th ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, 1999.
- [84] C. Souza, J. Froehlich and P. Dourish, "Seeking the source: software source code as a social and technical artifact," *Proc. Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, ACM, 2005.
- [85] E. Gilbert and K. Karahalios, "CodeSaw: A Social Visualization of Distributed Software Development," *Proc. INTERACT 2007*, 2007, pp. 303–316.
- [86] S. Sarkar, R. Sindhgatta and K. Pooloth, "A collaborative platform for application knowledge management in software maintenance projects," *Proc. Proceedings of the 1st Bangalore annual Compute conference*, ACM, 2008.
- [87] A. Sarma, L. Maccherone, P. Wagstrom and J. Herbsleb., "Tesseract: Interactive visual exploration of socio-technical relationships in software development," *Proc. Proceedings of the 2009 IEEE 31st International Conference on Software Engineering*, IEEE Computer Society, 2009, pp. 23-33.
- [88] A. Aguiar and G. David, "WikiWiki weaving heterogeneous software artifacts," *Proc. Proceedings of the 2005 international symposium on Wikis*, ACM, 2005.
- [89] W. Xiao, C. Chi and M. Yang, "On-line collaborative software development via wiki," *Proc. Proceedings of the 2007 international symposium on Wikis*, ACM, 2007.
- [90] N. Boulila, AH Dutoit and B. Brügge, "D-Meeting: An Object-Oriented Framework for supporting distributing modeling of software,," *Proc. ICSE 2003 International Workshop on Global Software Development*, 2003.
- [91] A. Fernández, B. Garzaldeen, I. Grützner and J. Münch, "Guided support for collaborative modeling, enactment and simulation of software development processes," *Software Process: Improvement and Practice*, vol. 9, no. 2, 2004, pp. 95–106.
- [92] B. Bruegge, A. Lucia, F. Fasano and G. Tortora, "Supporting Distributed Software Development with fine-grained Artefact Management," *Proc. Proceedings of the IEEE international conference on Global Software Engineering*, IEEE Computer Society, 2006.

- [93] H. Spanjers, M. ter Huurne, B. Graaf, M. Lormans, D. Bendas and R. van Solingen, "Tool support for distributed software engineering," *Proc. IEEE International Conference on Global Software Engineering (ICGSE '06)*, 2006, pp. 187-198.
- [94] J. A. Espinosa and E. Carmel, "The Impact of Time Separation on Coordination in Global Software Teams: a Conceptual Foundation," *Journal of Software Process: Practice and Improvement*, vol. 8, no. 4, 2004, pp. 249-266.
- [95] P. Keil, D. J. Paulish and R.S. Sangwan, "Cost estimation for global software development," *Proc. Proceedings of the 2006 international workshop on Economics driven software engineering research*, ACM, 2006.
- [96] N. Ramasubbu and R.K. Balan, "Globally distributed software development project performance: an empirical analysis," *Proc. Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, ACM, 2007.
- [97] R. Kraut, R. Fish, B. Root and B. Chalfonte, "Informal communication in organizations: Form, function and technology," *Proc. S. Oskamp & S. Spacapan, (Eds.), People's reactions to tech-nology in factories, offices and aerospace, The Claremont Symposium on Applied Social Psychology, Sage Publications*, 1990, pp. 145-199.
- [98] J. D. Herbsleb, D. J. Paulish and M. Bass, "Global software development at siemens: experience from nine projects," *Proc. Proceedings of the 27th international conference on Software engineering*, ACM, 2005.
- [99] N. S. Shami, N. Bos, Z. Wright, S. Hoch, K. Y. Kuan, J. Olson and G. Olson, "An experimental simulation of multi-site software development," *Proc. Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research*, IBM Press, 2004.
- [100] L. Kiel, "Experiences in Distributed Development: A Case Study," *Proc. Workshop on Global Software Development at ICSE*, 2003.
- [101] K.J. Stewart and S. Gosain, "Impacts of ideology, trust, and communication on effectiveness in open source software development teams," *Proc. Twenty-Second International Conference on Information Systems*, 2001, pp. 507-512.
- [102] A.L. Morán, J. Favela, A. Martinez and D. Decouchant, "On the Design of Potential Collaboration Spaces," *The International Journal of Computer Applications and Technology (IJCAT)* vol. 19, no. 3/4, 2004, pp. 261-271.
- [103] D. Damian, F. Lanubile and T. Mallardo, "The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations," *Proc. Proceedings of the 28th international conference on Software engineering*, ACM, 2006.

- [104] V. Bellotti, N. Ducheneaut, M. Howard and I. Smith, "Taskmaster: recasting email as task management," *Proc. Position paper for the CSCW'02 workshop on Redesigning Email for the 21st Century*, 2002.
- [105] M. Handel and J.D. Herbsleb, "What is chat doing in the workplace?," *Proc. The 2002 ACM Conference on Computer Supported Cooperative Work* ACM Press, 2002, pp. 1-10.
- [106] E. Isaacs, A. Walendowski and D. Ranganathan, "Hubbub: a sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions," *Proc. Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, ACM, 2002.
- [107] S. Fussell, S. Kiesler, L. D. Setlock and P. Scupelli, "Effects of instant messaging on the management of multiple project trajectories," *Proc. The SIGCHI Conference on Human Factors in Computing Systems*, 2004, pp. 191-198.
- [108] M. Munir and O. Lorne, "What do groups need? A proposed set of generic groupware requirements," *ACM Trans. Comput.-Hum. Interact.*, vol. 1, no. 3, 1994, pp. 245-268; DOI <http://doi.acm.org/10.1145/196699.196715>.
- [109] R. Morteo, V. González, J. Favela and G. Mark, "Sphere Juggler: fast context retrieval in support of working spheres," *Proc. ENC 2004*, IEEE Press, 2004, pp. 361-367.
- [110] J. Camacho, J. Favela and V. González, "Supporting the Management of Multiple Activities in Mobile Collaborative Working Environments," *Proc. CRIWG 2006*, 2006, pp. 381-388.
- [111] A.L. Morán, J. Favela, R. Romero, H. Natsu, C. Perez, O. Robles and A. Martinez, "Potential and actual collaboration support for distributed Pair-Programming," *Computación y sistemas* vol. 11, no. 3, 2008, pp. 211-229.
- [112] F. Davis, "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," *MIS Quarterly* vol. 13, 1989, pp. 319-340.
- [113] Microsoft, "Windows Live Messenger," 2007; <http://windowslive.com/desktop/messenger>.
- [114] O. G. León and I. Montero, "Diseño de investigaciones," *Introducción a la lógica de la investigación en Psicología y Educación*, 2a ed., McGraw-Hill Interamericana de España, 1997.
- [115] S. Inc., *SPSS 15 for Windows*, SPSS Inc, 2006.

- [116] M.J. Schervish, “P Values: What They Are and What They Are Not,” *The American Statistician* vol. 50, no. 3, 1996, pp. 203-206.
- [117] A.L. Morán, “Potential and Actual Collaboration Support in the design of distributed Groupware,” PhD Thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 2005.
- [118] J. Pavón and J.J. Gómez-Sanz, “Agent Oriented Software Engineering with INGENIAS,” *Proc. The Multi-Agent Systems and Applications III: CEEMAS 2003 (LNAI)*, 2003, pp. 394-403.
- [119] Jabber-Clients, “The power of presence,” 2008; <http://www.jabber.com/CE/JabberClients>.
- [120] Digité, “Globally Distributed Agile ALM from Digité,” 2003; <http://www.digite.com/products/digite-products.htm>.
- [121] J. Brooke, “SUS: A quick and dirty usability scale,” *Proc. Usability evaluation in industry* London, UK: Taylor & Francis, 1996, pp. 189–194.
- [122] J.W. Cotton, *Analyzing within-subjects experiments*, Lawrence Erlbaum Associates, 1998, p. 331.
- [123] M. Tohidi, W. Buxton, R. Baecker and A. Sellen, “Getting the Right Design and the Design Right: Testing Many Is Better Than One,” *Proc. CHI 2006*, ACM, 2006, pp. 1243-1252.
- [124] M. Korkala and P. Abrahamsson, “Communication in Distributed Agile Development: A Case Study,” *Proc. 33rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE Press, 2007.
- [125] F. Détienne, “Collaborative design: Managing task interdependencies and multiple perspectives,” *Interacting with Computers*, vol. 18, no. 1, 2005, pp. 1-20.
- [126] A. Morán, J. Favela, A. Martinez and D. Decouchant, “Document Presence Notification Services for Collaborative Writing,” *Proc. Seventh International Workshop on Groupware*, 2001, pp. 125-135.
- [127] S. Auer, S. Dietzold, J. Lehmann and T. Riechert, “Ontowiki: A tool for social, semantic collaboration,” *Proc. Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at WWW 2007*, 2007.

APÉNDICE A.

CUESTIONARIO TAM (Technology Acceptance Model)

Sistema _____ Fecha _____ / _____ / _____

Escala de percepción de utilizad y facilidad de uso.

Instrucciones: Para cada uno de los enunciados marque con una cruz el recuadro que mejor describa su reacción al Sistema que utilizaste. Asume que dicho Sistema (prototipo) será desarrollado como un producto funcional.

	En desacuerdo							De acuerdo
1. Utilizando este sistema en mi trabajo podría llevar a cabo mis tareas más rápidamente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
2. Utilizando este sistema mejoraría mi desempeño laboral.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
3. Utilizar este sistema haría que fueran más fáciles las tareas de mi trabajo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
4. Utilizando este sistema mejoraría mi efectividad en el trabajo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
5. Utilizando este sistema incrementaría mi productividad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
6. Encontraría útil este sistema en mi trabajo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
7. Encontraría fácil que el sistema hiciera lo que yo quiero que haga.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
8. Interactuar con el sistema sería claro y entendible.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
9. Aprender a operar este sistema sería fácil para mí.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
10. Es fácil llegar a ser hábil en el uso del sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
11. Encontraría fácil de usar el sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	
12. Encontraría flexible interactuar con el sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Extremadamente	Muy	Ligeramente	Neutral	Ligeramente	Muy	Extremadamente	

Por favor escribe cualquier comentario que tengas en relación al sistema.

APÉNDICE B.

Proceso de análisis y diseño del sistema multi-agente basado en la metodología ingenias.

Para realizar el proceso de análisis y diseño del sistema multi-agente se utilizó el INGENIAS Development Kit 2.8.

Para ayudar a los usuarios a determinar cuándo el momento para iniciar una interacción apropiada, se ha implementado un sistema multi-agente que cuenta con autonomía y que sea útil para la obtención y distribución de información de las diversas actividades de un proyecto sin la intervención de los usuarios. Para desarrollar el sistema multi-agente, se utilizó la metodología INGENIAS [1], a través de la cual se guió proceso de análisis y diseño y modelado de cada agente y las interacciones entre ellos.

Un ejemplo de modelo resultante del análisis y diseño de agentes se ilustra en la Figura 1. En este caso, el Agente de Monitor lleva a cabo dos tareas:

- i) el seguimiento de los archivos que el usuario está trabajando; e
- ii) informar que un archivo de trabajo está en uso.

Esta figura también muestra el rol de este agente, el cual es monitorizar actividad (representado por Activity Monitor) y su objetivo específico Identificar los archivos de trabajo en uso (Identify Working Files in Use).

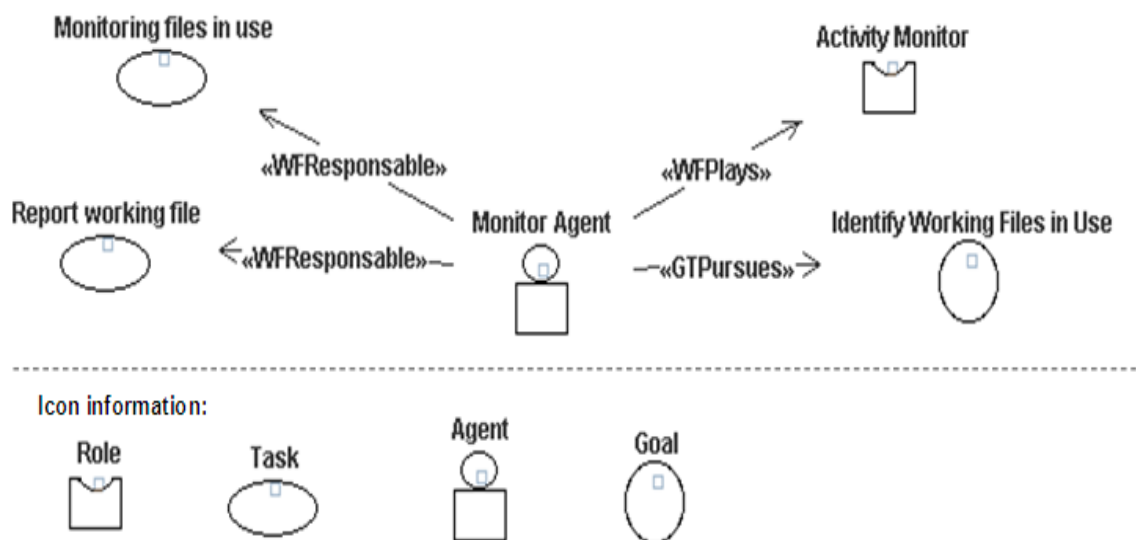


Figure 1. Modelo del agente monitor

Otra forma de visualizar el análisis y diseño de los agentes es mediante el modelo de interacción. En la Figura 2 se muestra un ejemplo de este tipo de modelos. En esta figura podemos ver los objetivos de cada agente y la interacción entre ellos.

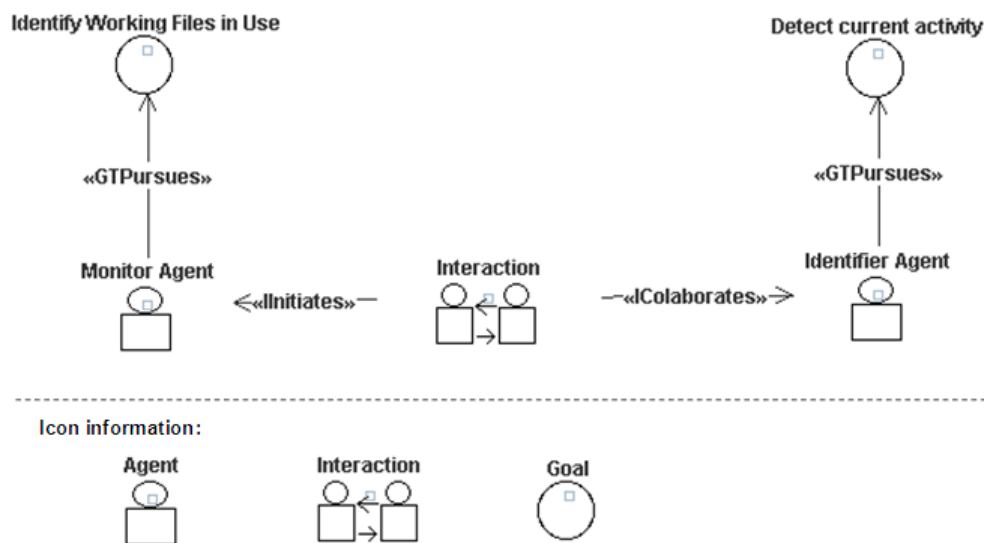


Figure 2. Modelo de interacción entre los agentes monitor e identificador.

Finalmente, para ilustrar como trabaja el sistema multi-agente se diseña un diagrama escenario. Para ejemplificar esto se presenta el siguiente escenario:

En una organización de Desarrollo Distribuido de Software, un diseñador del sistema tiene acceso a un archivo de UML a través de una aplicación de diagramación. Este archivo ha sido enviado al diseñador como parte de una tarea de diseño de interfaces.

Cuando el diseñador tiene alguna duda sobre el contenido del archivo UML, por lo general intenta contactar al analista responsable mediante el uso de un determinado medio de comunicación (p. ej. teléfono o una aplicación de mensajería instantánea). El diseñador por lo tanto generalmente interrumpe la actividad del analista.

Para explicar cómo funciona el sistema multi-agente, la figura 3 presenta un escenario propuesto. Una breve explicación es la siguiente:

En el momento en que el archivo UML está siendo editado (p. ej. "designInterface.vsd") (representado con el icono ApplicationEvent), un agente de Monitor observa el evento por medio de la tarea MonitoringFiles, y añade un registro al respecto en la bitácora (p. ej. archivo nombre, tipo de documento, fecha, hora y estado).

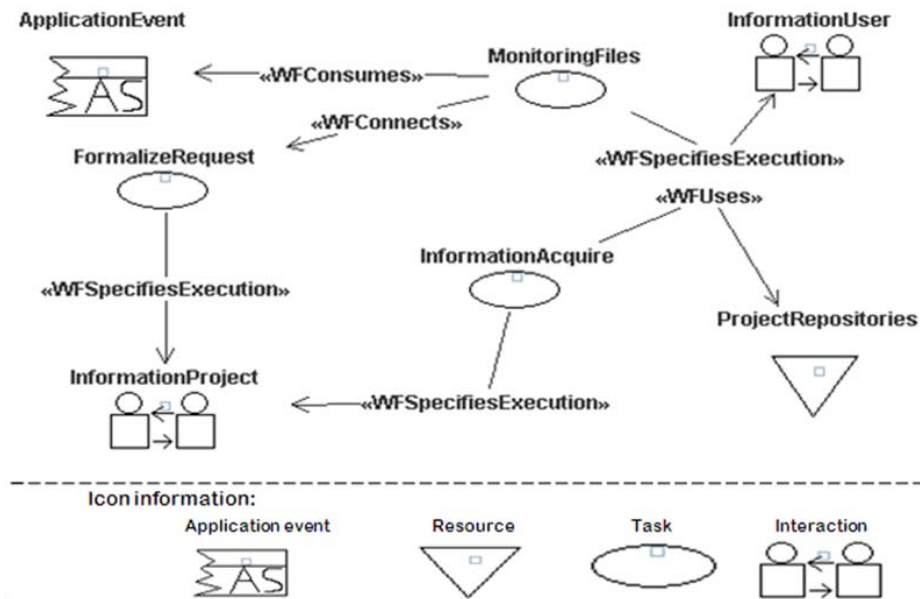


Figure 3. Diagrama escenario

Luego, esto desencadena una interacción entre un grupo de agentes en el lado del usuario (representado con el icono InformationUser): Un identificador de agente verifica si el archivo está relacionado con un proyecto. Si este es el caso, entonces verifica si el archivo está asociado a una actividad o proyecto. Además, el agente identificador actualiza la bitácora marcando el archivo como válido para hacer una solicitud de información al lado del servidor. De esta manera, los agentes detectan que el usuario está trabajando en una actividad válida de un proyecto válido (La actividad “Designing Graphical User Interface” corresponde al proyecto “Accounting-Project”).

Además, el lado del cliente realiza una solicitud de información al lado del servidor (representado por el icono de la tarea InformationAcquire). Esta petición desencadena la interacción entre otro grupos de agentes de la siguiente manera: la solicitud de información del lado del usuario es recibido por el Agente Administrador. Luego, el agente administrador revisa si el “Accounting-Project” es un proyecto activo (p. ej. hay personas trabajando en este proyecto) y de esta forma prueba que la petición es correcta. Posterior a esto, el agente del proyecto localizado busca información referente a este proyecto (tal como desarrolladores implicados, estatus de actividad de los desarrolladores, etc.). Después al encontrar esta información, ésta es enviada al agente interfaz (del lado del usuario). De esta manera, la información solicitada sobre el proyecto (p. ej. “Accounting-Project”) es entregada y listo para ser mostrado al usuario final.

Referencias:

- [1] J. Pavón and J.J. Gómez-Sanz, “Agent Oriented Software Engineering with INGENIAS,” Proc. The Multi-Agent Systems and Applications III: CEEMAS 2003 (LNAI), 2003, pp. 394-403.

APÉNDICE C.

CUESTIONARIO SUS (System Usability Scale)

Sistema _____ Fecha ____/____/____

Escala de facilidad de uso del sistema

Instrucciones: Para cada uno de los enunciados marque con una cruz el recuadro que mejor describa su reacción al Sistema que utilizaste. Asume que dicho Sistema (prototipo) será desarrollado como un producto funcional.

	Completamente en desacuerdo			Completamente de acuerdo		
1. Creo que me gustaría usar este sistema a menudo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. El sistema me pareció innecesariamente complejo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Pensaba que el sistema era sencillo de utilizar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Creo que necesitaría la ayuda de un técnico para poder usar este sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. Me pareció que las distintas funciones de este sistema estaban bien integradas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. Me pareció que había demasiadas incoherencias en este sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Creo que casi todo el mundo aprendería a usar este sistema rápidamente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. Me pareció que el sistema es muy complicado de usar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. Me encontré muy a gusto utilizando el sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. Tuve que aprender muchas cosas antes de poder hacerme con este sistema.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Por favor escribe cualquier comentario que tengas en relación al sistema.