

Evaluando el soporte al awareness del sistema de programación colaborativa COLLECE con técnicas de seguimiento ocular

Ana Isabel Molina Univ. de Castilla-La Mancha Paseo de la Universidad, 4 13071 Ciudad Real +34 926 295 300 Analsabel.Molina@ uclm.es

Jesús Gallardo Universidad de Zaragoza Ciudad Escolar, s/n 44003 Teruel +34 978 645 387 Jesus.Gallardo@ unizar.es

Miguel Ángel Redondo 13071 Ciudad Real +34 926 295 300 Miguel.Redondo@ uclm.es

Crescencio Bravo Univ. de Castilla-La Mancha Univ. de Castilla-La Mancha Paseo de la Universidad, 4 Paseo de la Universidad, 4 13071 Ciudad Real +34 926 295 300 Crescencio.Bravo@ uclm.es

RESUMEN

El área del aprendizaje de la programación puede beneficiarse de los principios del CSCW. Existen varios sistemas que han abordado esta problemática, siendo el soporte al awareness limitado en la mayor parte de ellos. Con el objetivo de soportar tareas de programación colaborativa distribuida síncrona se creó el sistema COLLECE. En este trabajo se describe una experiencia de evaluación de este sistema, incidiendo especialmente en las técnicas de soporte al awareness que incorpora. En dicho estudio se combinan varias técnicas de evaluación de sistemas interactivos (cuestionarios, testing en laboratorio, evaluación heurística y técnicas de seguimiento ocular), unas más subjetivas y otras más obietivas. El uso combinado de todas estas técnicas nos ha permitido realizar un análisis más completo del sistema COLLECE y, especialmente, de su soporte al awareness.

Categorías y Descriptores

H.5.3 [Group and Organization Interfaces]

Términos Generales

Measurement, Design, Experimentation, Human Factors.

Palabras clave

Programación colaborativa, evaluación, usabilidad, groupware, awareness, eye tracking.

1. INTRODUCCIÓN

Los avances que se han dado en los últimos años en el área de las telecomunicaciones e Internet han permitido la aplicación de los principios del CSCW (Computer Supported Cooperative Work) [1] en multitud de ámbitos. Una de las actividades que puede beneficiarse del trabajo en grupo es el ámbito de la programación. Con el objetivo de dar soporte a esta actividad se creó la aplicación COLLECE (COLLaborative Edition, Compilation and Execution of programs). COLLECE permite la edición, compilación y ejecución colaborativa síncrona de programas en Java y C. Este sistema ha sido evaluado en distintas ocasiones y con distintos objetivos. Así, por ejemplo, se ha estudiado la calidad del *producto* obtenido por parte de un grupo de programadores, junto con el proceso que daba lugar a la solución grupal [2]. También se han realizado evaluaciones basadas en la opinión de usuarios del sistema, que han sido tanto estudiantes de programación como profesionales del ámbito del desarrollo software [3].

Uno de los aspectos más destacados a considerar en el desarrollo de cualquier sistema groupware es el awareness [4]. Se define awareness como la percepción de la actividad desarrollada por el resto de miembros del grupo de trabajo y el conocimiento sobre dicho grupo. El uso de las técnicas de awareness permite proporcionar un contexto para la propia actividad, mejorando la efectividad y eficiencia del trabajo en grupo [5].

En este trabajo nos centramos en analizar y evaluar el soporte al awareness proporcionado por el sistema COLLECE. A diferencia de evaluaciones anteriores, principalmente basadas en la opinión subjetiva de los usuarios, en este trabajo se propone complementar dicha información con la proporcionada por un dispositivo de seguimiento ocular (eve tracker). Dicha técnica ha sido empleada con éxito en varios estudios de evaluación de la usabilidad [6][7]. Su aplicación permite a los evaluadores obtener medidas más objetivas (de naturaleza fisiológica) sobre el comportamiento visual de los usuarios mientras interaccionan con una aplicación interactiva.

Este artículo se estructura en las siguientes secciones. En la próxima sección se revisan los principales trabajos relacionados con el ámbito de la programación colaborativa, así como de las técnicas de evaluación de sistemas colaborativos. A continuación. en la sección 3, se describen las características del sistema COLLECE, centrándonos en comentar el soporte al awareness que dicha aplicación incorpora. En la sección 4 se pasa a describir los detalles del estudio empírico realizado, así como los resultados obtenidos. Finalmente, en la sección 5, se exponen las conclusiones extraídas del presente trabajo y las líneas de continuación del mismo.

2. TRABAJOS RELACIONADOS

Existe un gran número de trabajos que han afrontado el reto de soportar la programación colaborativa distribuida. Así, por ejemplo, RECIPE (REal-time Collaborative Interactive Programming Environment) [8] permite a programadores distribuidos geográficamente participar de manera concurrente en el diseño, codificación, pruebas, depuración y documentación de un programa. Para conseguir esto, RECIPE facilita la conversión de compiladores y depuradores monousuario en aplicaciones colaborativas, así como la integración de editores colaborativos existentes en el sistema. Sin embargo, no ofrece herramientas especializadas para la comunicación entre los programadores y también carece de herramientas adecuadas de soporte al awareness debido a su alto acoplamiento. Otro entorno que soporta la edición, compilación y ejecución de programas, y que



también incorpora canales para comunicación basada en texto y audio es DPE [9]. Sin embargo, este sistema presenta también un soporte limitado para la coordinación de tareas y el soporte al awareness.

Otras aproximaciones similares han intentado integrar soporte colaborativo en el entorno Eclipse mediante *plug-ins*. Uno de los trabajos más maduros en este campo es *Jazz Sangam* [10], que incorpora mensajería instantánea y control de versiones, entre otras funcionalidades. Los sistemas de este tipo suelen también tener limitaciones en cuanto al *awareness* debido a su alto acoplamiento y también a la falta de herramientas de coordinación y comunicación especializadas. Así, la presencia de elementos de soporte al *awareness*, aspecto esencial para mejorar la experiencia de trabajo colaborativo, es una de las carencias más relevantes de la mayoría de sistemas que soportan la programación colaborativa. Por tanto, a la hora de desarrollar el sistema COLLECE, uno de los puntos en el que más se centraron los esfuerzos fue en que incorporara un adecuado soporte al *awareness*, tal y como se comentará en la sección 3.

Una de las principales necesidades que existen, una vez que un sistema colaborativo ha sido implementado, es evaluar el soporte que da a la actividad colaborativa [12]. Esta evaluación puede hacerse en distintas dimensiones. Por ejemplo, es especialmente relevante para nuestro trabajo la evaluación del soporte al awareness. Así, Mangan et al. [13] destacan la importancia y la complejidad de la evaluación del soporte al trabajo en grupo y en particular del soporte al awareness. En ese trabajo se destaca la importancia de realizar evaluaciones de este tipo, a ser posible con evaluaciones previas antes de los estudios definitivos. Otros trabajos, como el de Convertino et al. [14], han realizado evaluaciones del awareness de actividad, desarrollando tareas en las cuales se pongan a prueba los mecanismos, tanto para percibir el trabajo síncrono como para hacer lo propio con el asíncrono. La adecuada selección de estas actividades, replicando el trabajo real con las herramientas, se muestra entonces clave para llevar a cabo una evaluación de interés. En general, existen distintos problemas que se ha estudiado que deben considerarse a la hora de evaluar el soporte al awareness. Dos de estos problemas son, por un lado, el hecho de que la recepción de información de awareness pueda generar una interrupción en el trabajo del usuario y, por el otro, la posible intromisión en la privacidad del usuario que puede producirse al calcular y visualizar esta información. Ambos problemas se analizan en trabajos como el de Röcker y Magerkurth [15], en el cual se intenta buscar una aproximación distinta al diseño de interfaces de usuario que evite dichos problemas. Finalmente, destacamos la checklist propuesta por Antunes et al. [16], de gran utilidad para que los desarrolladores software puedan inspeccionar la calidad del soporte al awareness de sus desarrollos.

Por otro lado, existen distintos métodos, ampliamente utilizados, para la *evaluación de la usabilidad* de sistemas monousuario. Tal es el caso de las técnicas de análisis de la consistencia o inspección de estándares, el empleo de recorridos cognitivos, así como el uso de heurísticas de evaluación [12]. Sin embargo, estas técnicas, que han resultado válidas e útiles en su aplicación, a pesar de no contar con bases teóricas sólidas, no resultan fáciles de adaptar y aplicar al evaluar sistemas *groupware*. En este sentido, se han propuesto nuevas técnicas de evaluación de la usabilidad para estos sistemas. Destacamos las aportaciones de Pinelle y Gutwin [17], quienes proporcionan una definición de "*groupware usability*" definido "*extent to which a groupware*

system allows teamwork to occur –effectively, efficiently, and satisfactorily– for a particular group and a particular group activity". Entre las nuevas técnicas de evaluación de usabilidad del groupware encontramos métodos de inspección [18], técnicas de recorrido cognitivo adaptado a sistemas colaborativos [17][19] y una adaptación de las heurísticas de Nielsen para su aplicación a este tipo de software [20].

Sin embargo, la mayoría de estas técnicas de evaluación se basan en frameworks heurísticos y el uso de cuestionarios, cumplimentados por expertos (en el caso de la evaluación heurística) o por usuarios del sistema. La mayor parte de estas técnicas son altamente subjetivas y, por tanto, los resultados obtenidos al aplicarlas están muy sujetos a sesgos. En este trabajo nos proponemos complementar dicha evaluación con técnicas más objetivas, en particular con técnicas de seguimiento ocular (eye tracking) [6]. Las sesiones de eye tracking permiten extraer conclusiones sobre el comportamiento de exploración visual que realizan los usuarios al interaccionar con un sistema software. Puesto que se trata de medidas de naturaleza fisiológica, los resultados obtenidos no están tan sujetos a sesgos ni pueden ser controlados por los usuarios. Dicha técnica de análisis ha sido empleada de forma exitosa en estudios de usabilidad de sistemas interactivos monousuario, principalmente en el ámbito web [6][7]. En los últimos años también está despertando interés su uso en situaciones multiusuario, apareciendo, así, nuevos conceptos como el llamado dual eye tracking [21] o la transferencia en remoto de los puntos en los que está fijando un usuario su atención visual (el llamado gaze-transfer [22]).

Durante una sesión de *eye tracking* se recopila gran cantidad de *métricas* [7], que pueden emplearse para conocer las áreas de la interfaz en las que los usuarios centran más su *atención*, el *patrón de escaneo* de los distintos elementos que componen la interfaz o el *esfuerzo cognitivo* que les supone tratar de entender la información visual proporcionada. La mayoría de estas métricas están relacionadas con el número y duración de las llamadas *fijaciones*, que se obtienen por la estabilización de los ojos en un área de la imagen visualizada por un periodo de tiempo determinado. Sin embargo, algunas de las métricas que se obtienen en una sesión de este tipo pueden tener más de una interpretación, por lo que se recomienda complementarlas con otras fuentes de información.

Vemos, por tanto, que existe un gran potencial en la utilización de esta nueva fuente de información para evaluar sistemas interactivos, en general, y sistemas *groupware*, en particular. El uso de las medidas proporcionadas por el *eye tracker* permite contrastar y complementar otras fuentes de información habitualmente utilizadas para testear este tipo de sistemas. En nuestro caso la emplearemos para averiguar el uso que hacen los usuarios de las distintas técnicas de *awareness* incluidas en el sistema COLLECE, cuyas principales características se detallan en la siguiente sección.

3. COLLECE

El entorno **COLLECE** (*COLLaborative Edition, Compilation and Execution of programs*) [3] es un sistema que da soporte a la edición, compilación y ejecución de código fuente de manera colaborativa distribuida síncrona. Actualmente, el entorno es capaz de dar soporte al trabajo con los lenguajes de programación Java y C. En COLLECE, como en otros sistemas similares, los usuarios se agrupan en sesiones de trabajo en las cuales resuelven un problema de programación. Este problema puede implicar desarrollar código desde cero o modificar código existente.



COLLECE está pensado principalmente para fines educativos. Por este motivo, se distinguen dos *roles*: profesores y alumnos. Los profesores son capaces de definir sesiones de trabajo y de distribuir a los usuarios participantes en ellas. Una vez que el usuario elige la sesión en la cual entrar, el usuario pasa a ver la interfaz de usuario principal de COLLECE (Figura 1).

El área principal de la interfaz de usuario de COLLECE es el editor colaborativo de código (1), en el cual se elabora el programa que debe resolver el problema propuesto. Ese programa posteriormente se compilará y ejecutará, obteniendo la salida en la ventana de consola (2). Para poder editar el código, se establece un protocolo de coordinación que regula el turno de edición. Esto se consigue mediante el panel de turno (3). Se trata de un componente gráfico mediante el cual los usuarios pueden manifestar su intención de tomar el turno de edición del código. Cuando un usuario ha realizado una petición de toma de turno, los demás dan su opinión afirmativa o negativa mediante otros botones de este mismo panel. Cuando todos los usuarios están de acuerdo en dar el turno al usuario que lo solicitó, se produce el cambio.

A la hora de decidir si compilar y ejecutar el programa que se está editando se sigue un procedimiento similar al antes descrito, pero utilizando otros dos paneles (4 y 5) que sirven para tomar las decisiones en cuanto a cuándo compilar y cuándo ejecutar, respectivamente. Aparte de los ya comentados, otros elementos gráficos reseñables de la interfaz de usuario de COLLECE son el chat (6) y el panel de sesión (7). El chat es un chat estructurado que permite clasificar los mensajes gracias a un conjunto de introductores de sentencias predefinidos (por ejemplo: "Pienso que...", "Veo un fallo en...", "Falta un...") [23]. En cuanto al panel de sesión, éste incluye la información de nombre y foto de cada uno de los usuarios participantes en la sesión de trabajo, así como su estado (editando, compilando, etc.). Tanto el nombre como el estado están escritos en un color característico que, como se verá más adelante, identifica a ese usuario en ciertos elementos de soporte al awareness.

Además del soporte a la coordinación (panel de turno de edición, panel de compilación, panel de ejecución) y del soporte a la comunicación (chat estructurado), hay que destacar en COLLECE su soporte al awareness. Según [24], la ausencia de awareness es uno de los principales problemas del desarrollo de software distribuido. Para evitar este problema, en COLLECE se provee de información referente acerca de las personas, su estado y sus acciones. En la Figura 1 se resaltan los siguientes elementos de soporte al *awareness*: (a) ruta del archivo que se está editando. (b) línea que se está editando resaltada con el color característico del usuario que lo está haciendo (telecursor), (c) barra de desplazamiento colaborativa (barra multiscroll) para indicar dónde está trabajando el usuario que edita, también usando el color que le identifica, (d) línea que se está editando en un momento dado, (e) último evento que se ha reflejado en el panel de turno (también en los otros paneles), (f) semáforo que se pone de color verde cuando otro usuario ha llevado a cabo una petición que requiere respuesta, (g) nombre del usuario que está editando en ese momento, (h) el propio panel de sesión, que incluye (i) el usuario que edita, compila o ejecuta con el estado correspondiente v (i) el usuario al que pertenece la interfaz dentro de un recuadro. v (k) estado actual del trabajo en la sesión.

Para terminar esta descripción de COLLECE, cabe reseñar también su *sub-sistema de análisis de la colaboración e interacción*. Este sub-sistema parte del marco de análisis proceso-

solución propuesto en [3] y se basa en el manejo de *indicadores* de análisis que dan información sobre la actividad individual, la forma en la que se lleva a cabo la colaboración y la calidad del producto desarrollado.

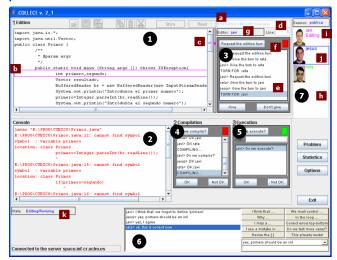


Figura 1. Interfaz de usuario de COLLECE.

Tal y como se ha comentado en la introducción, COLLECE ha sido evaluado en distintas ocasiones y con distintas perspectivas, pero en la mayor parte de ellas la fuente de información utilizada ha sido de naturaleza subjetiva, esto es, basada en la opinión de alumnos o desarrolladores software, siendo el uso de cuestionarios la técnica de evaluación más empleada. Proponemos, en este trabajo, complementar dicha evaluación con otra información, de naturaleza más objetiva, que puede recopilarse durante la actividad de edición colaborativa. Así, en dicha evaluación se combinarán el uso de las técnicas de cuestionarios de percepción subjetiva, evaluación heurística, técnicas de testing, thinking aloud retrospectivo (RTA) [25] y, finalmente, técnicas de eye tracking, realizadas en el contexto de un laboratorio de usabilidad.

4. EVALUACIÓN DE COLLECE

En esta sección se describirán los detalles de la experiencia realizada para evaluar las distintas técnicas de soporte al awareness que incorpora el sistema COLLECE. Nos centraremos en evaluar la utilidad que cada una de ellas tiene para los usuarios durante una sesión de programación colaborativa. Para ello, se usarán distintas fuentes de información, tanto indirectas y subjetivas como directas y más objetivas (las métricas proporcionadas por un dispositivo de seguimiento ocular, entre otras). A continuación, se describirá la muestra que participó en esta experiencia, así como la tarea que tuvieron que realizar, pasando, después, a comentar el diseño del experimento y los resultados obtenidos en el mismo.

4.1 Participantes

En la experiencia participaron un total de diez sujetos, todos ellos alumnos de la asignatura *Sistemas para la Colaboración* de quinto curso de la titulación de Ingeniería Informática de la Universidad de Castilla-La Mancha (UCLM), que accedieron a participar de forma voluntaria en la misma. Los alumnos cumplimentaron un *pretest* que nos permitió conocer su *perfil*. Todos ellos tenían buenos conocimientos en programación Java y conocían la terminología propia del ámbito del CSCW, así como



el concepto de *awareness* y las principales técnicas de soporte al mismo.

4.2 Tarea experimental

La tarea que tuvieron que realizar los alumnos consistía en modificar un programa Java suministrado en la sesión. Dicho programa utilizaba el framework ICE (*Internet Communication Engine*¹) para crear diferentes procesos cliente y servidor, por lo que la tarea tenía una complejidad media. Los alumnos disponían de un tiempo máximo de 15 minutos para completar la actividad.

Se crearon cinco grupos de dos alumnos cada uno, siendo la distribución de los alumnos en dichos grupos totalmente aleatoria.

4.3 Equipamiento

La experiencia se realizó en el *Laboratorio de Usabilidad* con el que cuenta el grupo CHICO [26], de la UCLM. Para la realización de las pruebas se hizo uso de un equipamiento de seguimiento ocular (sistema Tobii, modelo X60)². Para el diseño, realización y posterior análisis de la sesión de *eye tracking* se hizo uso del software *Tobii Studio* versión 3.0.2.

En las pruebas se contó con dos equipos, uno para cada miembro de la pareja. Ambos equipos estaban conectados a Internet. A pesar de que los dos usuarios estaban en el mismo laboratorio, no podían ver la pantalla del otro miembro del grupo, ni podían hablar entre ellos, sino que debían hacerlo usando las herramientas de comunicación y coordinación de COLLECE, de manera que se reprodujese de la forma más fiel posible el escenario de edición colaborativa distribuida soportada por este sistema. Uno de los miembros del grupo usaba el computador que incluía el equipamiento de seguimiento ocular, mientras que la actividad del segundo miembro del grupo era grabada mediante una cámara de vídeo que apuntaba a su pantalla.

4.4 Diseño experimental

En el desarrollo de este estudio empírico se siguieron tres etapas: cumplimentación del *pretest*, *fase de intervención o test* (realización de la tarea experimental) y *postest*. Cada pareja de participantes fue citada a distinta hora para realizar la experiencia, siendo la duración máxima de la prueba para cada grupo de 15 minutos de duración. En el diseño y posterior desarrollo de las pruebas de *eye tracking* se siguieron las *recomendaciones metodológicas* de Nielsen y Pernice [27].

Antes de pasar a la realización de la tarea experimental (o fase de *test*), los participantes cumplimentaban el *pretest* en papel. Dicho cuestionario pretendía determinar el *perfil* del usuario, para lo cual se incluyeron varias cuestiones que debían ser puntuadas en una escala *Likert* (de 1 a 5). Para cada ítem los sujetos indicaban su nivel de acuerdo o desacuerdo con respecto a seis afirmaciones que pretendían medir su nivel de conocimientos en programación y su nivel de conocimientos en Java y en ICE; su nivel de conocimientos teóricos sobre *sistemas colaborativos* y técnicas de soporte al *awareness*; y, finalmente, su experiencia en el uso de herramientas colaborativas, herramientas de programación colaborativa y en el uso de la herramienta COLLECE. A continuación, los alumnos debían puntuar, en una escala de 1 a 5, el nivel de *facilidad de uso percibida* (**PEU**), *utilidad percibida*

(PU) e intención de uso (ITU) de cada una de las técnicas de awareness incluidas en la interfaz del sistema COLLECE (y descritas en la sección 3). Estas tres características son las que pueden medirse al aplicar el framework de evaluación TAM (Technology Acceptance Model) propuesto por Davis [28].

Concluido el *pretest*, las parejas de estudiantes pasaban a la *fase* de intervención o test. Antes de empezar, se realizaba la calibración del dispositivo de eye tracking para uno de los integrantes de la pareja. Es necesario comentar que no todas las personas pueden participar en una experiencia de este tipo. Así, existen problemas de calibración cuando el participante usa gafas bifocales, lentillas o si las condiciones de iluminación no son las apropiadas [29]. Estos factores afectan a la fiabilidad y validez de los datos obtenidos, por lo que los sujetos que presenten alguno de estos problemas deben ser eliminados de la muestra final.

Una vez que las parejas de alumnos finalizaron la tarea de modificación y ampliación del programa planteado, pasaron a completar de forma individual el *posttest*. En esta última fase los alumnos debían puntuar, utilizando la misma escala que se les suministró antes de la prueba (y adaptada del modelo de evaluación TAM), la *facilidad de uso percibida* (PEU) y la *utilidad percibida* (PU) de los principales elementos de soporte al *awareness* soportados por COLLECE, pero teniendo en cuenta, en este caso, el uso que de los mismos habían hecho durante la realización de la tarea experimental.

Finalmente, y teniendo en cuenta el perfil de los participantes en la prueba (alumnos de la asignatura de *Sistemas para la Colaboración*), se pidió a los diez participantes que adoptaran el rol de expertos en interfaces de usuario *groupware*, pasando a realizar una *evaluación heurística* del sistema COLLECE, empleando para ello las dimensiones y sub-dimensiones que incorpora el *framework* de Gutwin y Greenberg [30].

4.5 Resultados y discusión

En esta sección se analiza e interpreta toda la información recopilada durante la experiencia. Tal y como se ha comentado, se incluyen valores obtenidos de distintas fuentes. Algunos de ellos son de naturaleza objetiva (tiempo total empleado para realizar la tarea y las métricas proporcionadas por el *eye tracker*), mientras que otros se obtienen a partir de los cuestionarios suministrados (conocimientos previos, utilidad percibida, intención de uso, etc.).

En la Tabla 1 se muestran los valores relativos al *perfil* de los usuarios que participaron en la experiencia. Se muestran los valores medios de las respuestas dadas por los participantes. Tal y como podemos ver en esta tabla, los participantes tenían poca experiencia en el uso de la herramienta objeto de evaluación (M = 1,70), aunque tenían un conocimiento medio respecto a los fundamentos teóricos sobre CSCW (M = 2,80) y *awareness* (M = 2,60). La mayor parte de ellos tenían la percepción subjetiva de que eran buenos en programación (M = 3,90) y en el uso del lenguaje Java (M = 4,00), aunque no tanto en ICE (M = 1,60). No obstante, al tratarse de una tarea de comprensión y modificación de un programa dado, 2 de los 5 grupos fueron capaces de resolver la tarea en el tiempo establecido, por lo que los niveles de conocimiento de Java y ICE no son un factor que afecte negativamente al experimento.

Tabla 1. Perfil de los participantes en el estudio empírico

Puntuación*

¹ http://www.zeroc.com/ice.html

² http://www.tobii.com



Nivel de conocimiento en programación	3,90 (0,57)
Nivel de conocimiento en lenguaje Java	4,00 (0,82)
Nivel de conocimiento en ICE	1,60 (0,52)
Nivel de conocimientos teóricos sobre sistemas colaborativos	2,80 (0,63)
Nivel de conocimientos teóricos sobre <i>técnicas de</i> awareness	2,60 (0,84)
Nivel de experiencia práctica en el uso de la herramientas colaborativas	2,80 (0,63)
Nivel de experiencia práctica en el uso de la herramientas de programación colaborativa	1,90 (0,88)
Nivel de experiencia práctica en el uso de la herramienta COLLECE	1,70 (0,82)

^{*} Se muestra la media y, entre paréntesis, la desviación típica

Aunque, tal v como se ha comentado en la sección 3, COLLECE incluye una gran cantidad de elementos de soporte al awareness, vamos a centrar el análisis tan solo en algunos de ellos. En la figura 2 se resaltan los principales elementos que consideramos en el presente estudio. Las nueve áreas de la interfaz resaltadas en la figura son lo que, en terminología de la técnica de eye tracking, se conocen como áreas de interés (AOI), es decir, las secciones de la imagen cuyo análisis es de interés para los evaluadores y, por tanto, para las que se extraerán las métricas correspondientes. En la imagen se resaltan las áreas de la interfaz que incluyen la visualización del contexto compartido (AOI-Código), esto es, el código que está siendo editado de forma colaborativa, así como las áreas ocupadas por las herramientas de comunicación (AOI-Chat) y coordinación (AOI-Panel de turnos). Además de estas tres áreas de interés se incluyen aquellas que delimitan seis de los elementos de la interfaz que más explícitamente soportan el awareness. En la Tabla 2 se definen cada una de ellas.

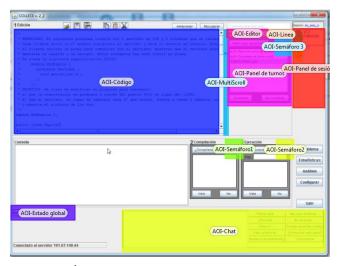


Figura 2. Áreas de interés (AOI) definidas en la interfaz de COLLECE.

Tabla 2. Definición de las áreas de interés (AOI) asociadas a los principales elementos de *awareness* de COLLECE

AOI	Herramienta de soporte al awareness incorporada COLLECE
AOI-Semáforos	Herramientas de toma de decisión sobre la tarea (Área de semáforos)
AOI-MultiScroll	Telepuntero del usuario remoto (usuario editor) en la barra multiscroll
AOI-Panel de sesión	Visualización de los participantes en la actividad

	(Panel de sesión)
AOI-Editor	Información visual sobre quién es el editor
AOI-Estado global	Indicador del estado actual de la actividad (editando, compilando, ejecutando)
AOI-Línea	Indicador del <i>número de línea</i> que se está editando actualmente

En la Tabla 3 se muestran las valoraciones que los diez participantes hicieron de las principales técnicas de awareness, antes (pretest) y después (posttest) de la tarea experimental, según las dimensiones del framework de evaluación subjetiva TAM [28]. Las dos dimensiones que más nos interesa contrastar son las que miden la utilidad percibida (PU) y la intención de uso (ITU). Teniendo en cuenta las respuestas dadas por los usuarios en el pretest, podemos ver que la mayoría de ellos consideran que los elementos que les resultarán más útiles (PU) y, por tanto, consultarán más (ITU) son el panel de sesión, el número de la línea que está siendo editada por el otro miembro del grupo y los semáforos, que les permiten coordinarse y tomar decisiones; siendo la barra *multiscroll* el elemento que consideran que menos usarán durante la experiencia. Comparando las respuestas dadas en el pretest con las dadas en el posttest podemos ver que, una vez realizada la tarea, los usuarios siguen considerando como elementos más útiles (PU) el panel de sesión y los semáforos (M =4,50). Sin embargo, consideran que el número de línea en el que está trabajando el otro integrante del grupo no les ha aportado información tan útil durante la actividad (M = 3,30). En cuanto a la barra multiscroll, la valoración sobre su utilidad mejora una vez que han pasado la fase de test.

Tabla 3. Valoraciones de los elementos de *awareness* de COLLECE (*framework* TAM)**

	Pretest			Posttest	
	PEU*	PU*	ITU*	PEU*	PU^*
AOI-Panel de	4,70	4,50	4,50	4,70	4,50
sesión	(0,48)	(0,71)	(0,71)	(0,48)	(0,85)
AOI-Editor	4,50	4,20	4,30	4,20	4,30
	(0,71)	(1,03)	(1,06)	(0,92)	(0,82)
AOI-MultiScroll	4,30	4,00	3,70	4,50	4,40
	(0,95)	(1,33)	(1,34)	(1,27)	(1,35)
AOI-Semáforos	4,40	4,50	4,30	4,40	4,50
	(0,84)	(0,71)	(0,82)	(0,84)	(0,71)
AOI-Estado	4,00	3,90	3,80	3,90	3,50
global	(1,15)	(0,88)	(1,03)	(0,99)	(0,97)
AOI-Línea	4,50	4,50	4,40	3,70	3,30
	(0,42)	(0,71)	(0,70)	(1,42)	(1,64)

^{*} Se muestra la media y, entre paréntesis, la desviación típica

A continuación se comentarán los resultados y **métricas** obtenidos mediante el dispositivo de seguimiento ocular. En este caso sólo se pueden comentar los datos correspondientes a los cinco participantes cuyo comportamiento fue registrado por el *eye tracker*. Tal y como se comentó en la sección 2, la mayoría de métricas calculadas en una sesión de *eye tracking* se calculan a partir de las *fijaciones*. Así, por ejemplo, un mayor *número de fijaciones* en una determinada área de la imagen indica un mayor interés por parte del sujeto en la información suministrada en dicha área. Las cuatro métricas que permiten determinar el nivel de *interés* y, por tanto, *utilidad*, de una determinada parte de la

^{**} Valores para n=10 participantes



interfaz son [7]: el número de fijaciones en un AOI (#Fij), la duración total de las fijaciones (DTF), la duración total de las visitas a una determinada AOI (DTV) y el tanto por ciento del tiempo total de inspección (%Insp) dedicado a mirar una determinada AOI. Las duraciones y tiempos calculados se miden en segundos. En la Tabla 4 podemos ver los valores de todas estas métricas para las nueve áreas de interés definidas en el interfaz de usuario de COLLECE.

Tabla 4. Métricas proporcionadas por el dispositivo *eye* tracker***

AOI	#Fij*	DTF*	DTV*	%Insp*
AOI-Código	1693,20	414,59	538,80	45,47
	(287,20)	(76,60)	(105,44)	(3,69)
AOI-Chat	643,40	184,50	236,48	20,00
	(161,13)	(54,99)	(74,16)	(5,09)
AOI-Panel de turnos	115,40	26,23	30,50	2,48
	(66,80)	(17,00)	(18,86)	(1,36)
AOI-awareness	#Fij*	\mathbf{DTF}^*	\mathbf{DTV}^*	%Insp*
AOI-Panel de sesión	26,00	5,55	6,35	0,52
	(17,82)	(3,76)	(4,15)	(0,31)
AOI-Editor	16,20	4,39	4,56	0,41
	(10,43)	(3,28)	(3,30)	(0,30)
AOI-MultiScroll	13,20	4,08	4,23	0,35
	(13,66)	(4,73)	(4,78)	(0,38)
AOI-Semáforos	11,40	1,97	2,01	0,16
	(8,17)	(1,36)	(1,39)	(0,10)
AOI-Estado global	5,60	1,09	1,14	0,10
	(2,41)	(0,64)	(0,65)	(0,06)
AOI-Línea	2,00	0,30	0,32	0,03
	(2,35)	(0,31)	(0,33)	(0,03)

^{*} Se muestra la media y, entre paréntesis, la $desviaci\'on\ t\'ipica$

Considerando los valores calculados para cada AOI (Tabla 4), podemos comprobar como los usuarios dedican la mayor parte del tiempo a trabajar en el área de edición colaborativa (AOI-Código), así como a hacer uso de las herramientas de comunicación (chat) y coordinación (Panel de turnos). En relación a los elementos de awareness soportados por COLLECE, y ordenando según el valor de las cuatro métricas de interés consideradas (Figura 3), podemos ver como el elemento más consultado es el panel de sesión. Esta métrica coincide con la valoración subjetiva que dieron los participantes en los cuestionarios TAM (Tabla 5), va que éste fue el elemento considerado más útil (PU) y el que tenían más intención de usar (ITU) (M = 4,80). Según las métricas calculadas, los elementos menos consultados por los participantes (a los que dedicaron menos tiempo de inspección) fueron el indicador del estado global y el número de línea en el que está trabajando el otro miembro del grupo (Tabla 4 y Figura 3). Efectivamente, en el posttest (Tabla 5) estos fueron los elementos menos valorados (bajando la valoración de PU para ambos elementos con relación a las valoraciones dadas a los mismos en el *pretest*) (Tabla 5).

Tabla 5. Valoraciones de los elementos de *awareness* de COLLECE (*framework* TAM)**

Pretest		Posttest	
 PU*	ITU*	PU*	

AOI-Panel de sesión	4,80	4,80	4,60
	(0,45)	(0,45)	(0,89)
AOI-Editor	4,00	4,00	4,00
	(1,22)	(1,22)	(1,00)
AOI-MultiScroll	3,60	3,60	3,80
	(1,67)	(1,67)	(1,79)
AOI-Semáforos	4,80	4,60	4,60
	(0,45)	(0,55)	(0,55)
AOI-Estado global	4,00	4,00	3,00
	(1,00)	(1,00)	(1,00)
AOI-Línea	4,40	4,20	3,60
	(0,89)	(0,84)	(1,95)

^{*} Se muestra la media y, entre paréntesis, la desviación típica

^{**} Valores para n=5 participantes (cuyo comportamiento fue registrado con el eye tracker)

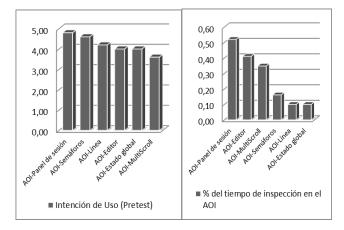


Figura 3. Ordenación de los elementos de *awareness* según ITU (izquierda) y %Insp (derecha).

Finalmente, tal y como se comentó en la sección 4.4, se pidió a las alumnos que valoraran el soporte que la interfaz de COLLECE proporcionaba a cada una de las dimensiones incluidas en el framework de Gutwin y Greenberg [30]. En la Tabla 6 podemos ver los valores que asignaron a cada una de ellas. Los participantes valoraron que el soporte a la dimensión quién de COLLECE era muy bueno, destacando principalmente la valoración de la sub-dimensión identidad (M = 4,70). Dicha información se muestra en el panel de sesión de la aplicación que, tal y como se ha comprobado, es el elemento más consultado (mayor %Insp) y considerado más útil (mayor valor de PU) por los participantes en la actividad. En cuanto a la dimensión qué, que permite definir la acción que están realizando los otros miembros del grupo y qué objetos están manipulando, fue valorada de forma positiva, aunque no tanto como la anterior dimensión. En cuanto a las sub-dimensiones relacionadas con el qué, aquellas sub-dimensiones que hacen referencia a los cambios que están realizando los integrantes del grupo (M = 4,30) y sobre qué artefactos los están realizando (M = 4.00) fueron las más valoradas, siendo la que indica el nivel de actividad (M = 2,80) la menos soportada por la interfaz de COLLECE. La dimensión dónde fue la considerada como la peor soportada por el sistema. De hecho, las sub-dimensiones peor puntuadas fueron las que hacen referencia a dónde están mirando el resto de integrantes del grupo. Dicha dimensión no es soportada por COLLECE (ni por la mayoría de sistemas groupware existentes) ya que, precisamente, se necesitaría contar con un equipo de seguimiento ocular para poder registrar dicha información y transmitirla a los otros miembros del grupo. Esa es la idea en la que se basan trabajos

^{**} Valores para n=5 participantes (cuyo comportamiento fue registrado con el eye tracker)



recientes que abordan el llamado *gaze-transfer* en el contexto de sistemas de trabajo en grupo e, incluso, en contextos de programación en grupo [31][32].

Tabla 6. Soporte a las dimensiones de awareness del sistema COLLECE según el framework de Gutwin y Greenberg*

Dim.	Sub-dimensión	Puntuación*
	Presencia - ¿Hay alguien en el espacio de trabajo?	4,50 (0,85)
QUIÉN	Identidad - ¿Quién está participando en la actividad? ¿Quiénes son?	4,70 (0,67)
	Autoría - ¿Quién está haciendo qué cosa?	4,50 (0,71)
	Acción - ¿Qué están haciendo los otros usuarios? ¿Cuáles son sus tareas o actividades actuales?	3,60 (1,07)
	Intención - ¿De qué objetivo forma parte la acción que están realizando?	3,30 (1,16)
QUÉ	Artefactos - ¿En qué objetos están trabajando?	4,00 (0,94)
	Cambios - ¿Qué cambios están realizando? ¿Dónde están realizando dichos cambios?	4,30 (0,95)
	Nivel de actividad - ¿Están activos en el espacio de trabajo? ¿Con qué rapidez trabajan?	2,80 (1,14)
	Localización - ¿Dónde están trabajando los otros usuarios?	3,50 (1,43)
DÓNDE	Mirada - ¿Dónde están mirando? ¿En qué parte del contexto compartido está su foco de atención?	2,90 (1,20)
	Vista - ¿Qué están viendo?	2,90 (1,20)
	Alcance - ¿Qué pueden llegar a hacer?	3,30 (1,42)

^{*} Se muestra la media y, entre paréntesis, la desviación típica

Una vez concluida la fase de *test* y *posttest* los alumnos volvieron a visionar las grabaciones realizadas durante la tarea experimental, realizándose así una sesión de thinking aloud retrospectivo (RTA) [25] en el que se buscaba conocer las impresiones de los participantes respecto al sistema COLLECE. A continuación se realizó una breve entrevista con los participantes. Esta fase permitió identificar cuáles eran considerados los puntos fuertes y débiles del sistema, así como posibles propuestas de mejoras. Entre los elementos más valorados por los participantes se encuentra el soporte a la coordinación basada en la cesión de turnos y entre los más criticados el chat estructurado. También varios participantes propusieron mejorar la identificación de dónde está editando el otro integrante del grupo. Aunque visualmente esta información se muestra mediante el AOI-Línea, quizás su localización en la interfaz, formato de visualización o tamaño no sea el más adecuado, ya que, tal y como se ha comprobado, a pesar de ser uno de los elementos con mejor valoración en intención de uso (ITU) en el pretest, fue de los menos atendidos visualmente por los alumnos (%Insp) y considerado como menos útil durante la tarea (PU en posttest). Los participantes también propusieron incluir un canal adicional de audio, mostrar en el contexto compartido (AOI-Código) la numeración de las líneas de código y flexibilizar la política de toma de decisiones en grupo, que en la versión actual de COLLECE se basa en el consenso de todos los miembros. Esta última mejora será de especial interés cuando el número de integrantes de cada grupo sea mayor que dos.

5. CONCLUSIONES

En el presente trabajo se ha descrito una experiencia de evaluación del sistema de programación colaborativa distribuida

síncrona COLLECE. A diferencia de evaluaciones anteriores, la evaluación realizada combina varias técnicas de evaluación de sistemas interactivos multiusuario. Así, se han aplicado, de forma conjunta, técnicas de *inspección* (evaluación heurística), *indagación* subjetiva (cuestionarios, entrevistas), así como técnicas de *testing* en un laboratorio de usabilidad (RTA y *eye tracking*). Mientras que la evaluación heurística basada en *checklists* es una técnica común en la literatura [16], la utilización de técnicas de *eye tracking* resulta más novedosa para la evaluación del soporte al *awareness*.

Somos conscientes del tamaño reducido de la muestra que ha participado en esta primera experiencia, por lo que los resultados obtenidos deben ser tomados con cautela y considerados como preliminares. El tamaño, en gran medida, está condicionado por la dificultad que supone analizar información dinámica cuando se aplican técnicas de *eye tracking*. Nos planteamos, por tanto, como continuación de este trabajo, replicar esta experiencia con una muestra mayor, que permita obtener resultados más concluyentes.

6. AGRADECIMIENTOS

El presente trabajo ha sido parcialmente soportado por el proyecto coordinado EDUCA-Prog del Ministerio de Ciencia e Innovación (TIN2011-29542-C02-02). Los autores quieren agradecer su participación a los alumnos de la UCLM que accedieron a participar en esta experiencia.

7. REFERENCIAS

- [1] Greif, I., 1988. *Computer-Supported Cooperative Work: A Book of Readings*. San Mateo, CA, Morgan Kaufmann.
- [2] Bravo, C., Redondo, M.A., Verdejo, M.F., Ortega, M., 2008. Framework for process and solution analysis in synchronous collaborative learning environments. *International Journal of Human-Computer Studies* 66 (11), 812–832.
- [3] Bravo, C., Duque, R., Gallardo, J., 2013. A groupware system to support collaborative programming: Design and experiences. *Journal of Systems and Software* 86 (7), pp. 1759-1771.
- [4] Dourish, P., Bellotti, V., 1992. Awareness and Coordination in Shared Workspaces. Proceedings of the Conference on Computer Supported Cooperative Work CSCW'92, Toronto, Canada, ACM Press, New York.
- [5] Gallardo, J., Molina A.I., Bravo, C., Redondo, M.A., Collazos, C.A., 2011. An ontological conceptualization approach for awareness in domain-independent collaborative modeling systems: Application to a model-driven development method. *Expert Systems with Applications* 38 (2011) 1099–1118.
- [6] Nielsen, J., Pernice, K., 2010. Técnicas de Eye Tracking para usabilidad Web. ANAYA Multimedia. New Riders, 2010.
- [7] Poole, A., Linden, J.B., 2004. Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects.
- [8] Shen, H., Sun, C., 2000. : RECIPE: a prototype for Internet-based real-time collaborative programming. Proceedings of the 2nd Annual International Workshop on Collaborative Editing Systems. Philadelphia, Pennsylvania, USA.

^{**} Valores para n=10 participantes



- [9] Jo, C.H., Arnold, A.J., 2003. A portable and Collaborative Distributed Programming Environment. *International Conference on Software Engineering*. Las Vegas, Nevada, USA, 198-203.
- [10] Devide, J., Meneely, A., Ho, C-w, Williams, L., and Devetisikiotis, M., 2008. Jazz Sangam: A Real-time Tool for Distributed Pair Programming of a Team Development Platform. Infrastructure for Research on Collaborative Software Engineering (IReCoSE) workshop at ACM SIGSOFT Foundations of Software Engineering (FSE), Atlanta, GA.
- [11] Gallardo, J., Molina, A.I., Bravo, C., Redondo, M.A., Collazos, C., 2011. Empirical and heuristic-based evaluation of collaborate modeling systems: An evaluation framework. *Group Decision and Negotiation* 20 (5).
- [12] Rubin, J., Chisnell, D., 2008. Handbook of Usability Testing. How to Plan, Design and Conduct Effective Tests. Wiley Publishing, Inc., Indianapolis, Indiana.
- [13] Mangan, M.A.S., de Araujo, R.M., Kalinowski, M., Borges, M., Werner, C.M.L., 2002. Towards the evaluation of awareness information support applied to peer reviews of software engineering diagrams. CSCWD 2002.
- [14] Convertino, G., Neale, D., Hobby, L., Carrollo, J., Rosson, M. 2004. A laboratory method for studying activity awareness. *Proceedings of the Third Nordic Conference on Human-Computer interaction*, Tampere, Finland (2004), 313-322.
- [15] Röcker, C., Magerkurth, C., 2007. Privacy and Interruptions in Team Awareness Systems. Universal Acess in Human Computer Interaction. Coping with Diversity Lecture Notes in Computer Science Volume 4554, 2007, pp 273-283.
- [16] Antunes, P., Herskovic, V., Ochoa, S.F., Pino, J.A., 2013. Reviewing the quality of awareness support in collaborative applications, Journal of Systems and Software, *in press*.
- [17] Pinelle, D., Gutwin, C., 2002. Groupware walkthrough: Adding context to groupware usability evaluation. *In:* Proceedings of the 2002 SIGCHI Conference on Human Factors in Computing Systems. Minneapolis, Apr. ACM Press, pp. 455–462.
- [18] Steves, M.P., Allen, R.H. 2001. Evaluating Collaborative Enterprises – A Workshop Report. In: Proceedings of the 10th Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. MIT, Cambridge, USA.
- [19] Ereback, A.L., Hook, K., 1994. Using Cognitive Walkthrough for Evaluating a CSCW Application. *In: Proceedings of ACM CHI '94*, pp. 91-92.
- [20] Baker, K., Greenberg, S., Gutwin, C., 2001. Heuristic evaluation of groupware based on the mechanics of

- collaboration. *In Proceedings of the 8th IFIP working conference on engineering for human-computer interaction (EHCI'01)*, pp 123–140.
- [21] Hennessey, C., 2012. Framework for colocated synchronous dual eye tracking. *DUET 2012. CSCW'12*, February 11–15, 2012, Seattle, Washington, USA. ACM 978-1-4503-0556-3/12/02.
- [22] Mueller, R., Helmert, J.R., Pannasch, S., Velichkovsky, B.M., 2011. Following closely? The effects of viewing conditions on gaze versus mouse transfer in remote cooperation. *DUET 2011*.
- [23] Lund, K., Baker, M.J., Baron, M., 1996. Modelling Dialogue and Beliefs as a Basis for Generating Guidance in a CSCL Environment, In: *Proceedings of the International Conference on Intelligent Tutoring Systems*, pp. 206-214.
- [24] Herbsleb, J., Grinter, R., 1999. Architectures, coordination, and distance: Conway's law and beyond. *IEEE Software* 16 (5), 63-70.
- [25] Tobii, 2009. Retrospective Think Aloud and Eye Tracking. Whitepaper by Tobii Technology http://www.tobii.com/.
- [26] Ortega, M., Molina, A.I., Redondo, M.A. et. al., 2009. CHICO (Computer – Human Interaction and Collaboration), UCLM. *IE Comunicaciones (Revista Iberoamericana de Informática Educativa)*, Núm. 15, Enero-Junio 2012, pp 11-17.
- [27] Nielsen, J., Pernice, K., 2009. Eyetracking Methodology. How to Conduct and Evaluate Usability Studies Using Eyetracking. *Nielsen Norman Group* 2009.
- [28] Davis, F. D., 1993. User acceptance of information technology: System characteristics, user perceptions and behavioral impacts. *International Journal of Man–Machine Studies*, 38(3), 475–487. 1993.
- [29] Tobii, 2011. Accuracy and precision. Test report, Tobii T60 XL Eyetracker, http://www.tobii.com/.
- [30] Gutwin, C., & Greenberg, S., 2002. A descriptive framework of workspace awareness for real-time groupware. Computer supported cooperative work (CSCW), Special issue on awareness in CSCW, *The Journal of Collaborative Computing*, 11(3–4), 411–446.
- [31] Bednarik, R., Shipilov, A, 2011. Gaze cursor during distant collaborative programming: A preliminary analysis. In Proceedings of the DUET 2011: Dual Eye Tracking in CSCW, 2011.
- [32] Bednarik, R., Shipilov, A. Pietinen, S. 2011. Bidirectional gaze in remote computer mediated collaboration: setup and initial results from pair-programming. *In Proceedings of the ACM 2011 conference on Computer supported cooperative work, CSCW '11*, pages 597–600, New York, NY, USA,. ACM.