

ASSE 2016, 17° Simposio Argentino de Ingeniería en Software

## Sistemas Colaborativos con Awareness: Requisitos para su Modelado

Luis Mariano Bibbo<sup>1</sup>, Roxana Giandini<sup>1</sup>, Claudia Pons<sup>1,2,3</sup>

<sup>1</sup> Facultad de Informática - UNLP,  
1900 La Plata, Bueno Aires, Argentina {lmbibbo, giandini,  
[cpons](mailto:cpons@lifa.info.unlp.edu.ar)}@lifa.info.unlp.edu.ar

<sup>2</sup> CIC: Comisión de Investigaciones Científicas,  
1900 La Plata, Bueno Aires, Argentina

<sup>3</sup> UAI: Universidad Abierta Interamericana,  
Capital Federal, Argentina

**Resumen.** El concepto de awareness, que se define como la percepción y el conocimiento del grupo y sus actividades, constituye uno de los principales aspectos de los sistemas colaborativos. Los desarrolladores de software requieren lenguajes y herramientas de modelado que permitan expresar awareness en este tipo de sistemas. Este trabajo propone revisar publicaciones científicas con el objetivo de describir características que un modelo conceptual de sistemas colaborativos tiene que incorporar para incluir la funcionalidad de awareness. El resultado de esta investigación nos permitió enumerar requisitos que un modelo abstracto para tal fin debe cumplir. Se detectó que es especialmente necesario incluir los conceptos de proceso colaborativo y protocolo para poder modelar la dinámica entre los usuarios finales del sistema. Finalmente, la incorporación de estos conceptos requiere de awareness específico vinculado a cada uno de ellos que debe ser incluido también en el modelo resultante.

### 1 Introducción

Los sistemas colaborativos son aplicaciones donde un grupo de usuarios que persiguen un objetivo común desarrollan distintas actividades en forma conjunta. El software que utilizan les permite comunicarse, colaborar y coordinar actividades entre ellos. El sistema no contempla a un usuario en particular sino que atiende las necesidades del grupo. De acuerdo a Ellis et al. [1], las plataformas de colaboración son: “Sistemas de computadoras que proveen una interfaz a un entorno compartido y que soportan a un grupo de usuarios que tienen un objetivo común”.

Como se mencionó en [1] y [2], la coordinación brinda un equilibrio entre el trabajo individual de los participantes y el esfuerzo que deben realizar los usuarios para lograr un objetivo grupal. En definitiva, una efectiva coordinación mejora las

posibilidades de colaboración ordenando la participación de los usuarios. Esto se implementa por un lado, a alto nivel mediante **procesos colaborativos** los cuales determinan qué tareas colaborativas y en qué orden tienen que llevar adelante los usuarios. Por otro lado se utilizan **protocolos** para definir qué acciones pueden realizar los roles en cada momento en una actividad colaborativa.

Para conseguir una efectiva colaboración y lograr los objetivos que se persiguen, los usuarios tienen que estar informados sobre las acciones que los otros participantes realizan en el ambiente y cómo esas acciones afectan el entorno de trabajo. Esta información que brinda el sistema se llama awareness. El awareness es la percepción o conocimiento del grupo y de las actividades que realizan los usuarios en el sistema compartido. Uno de los aportes más relevantes en la teoría de awareness lo realizaron Gutwin y Greenberg [3] introduciendo un Framework que define diferentes awareness a partir de un conjunto de preguntas relevantes. La principal contribución ha sido identificar 10 conceptos que definieron como “*Workspace Awareness*”.

En este trabajo revisamos propuestas científicas que presenten recursos de ingeniería de software útiles para el modelado desarrollar sistemas colaborativos que incluyan awareness. Estos pueden ser: métodos, modelos, frameworks, guías, prototipos, o cualquier artefacto de ingeniería de software que faciliten la construcción de este tipo de sistemas. Específicamente, estamos interesados en compilar los requisitos específicos que un modelo conceptual de sistema colaborativo tiene que incluir para contemplar el awareness. Para lograr esto, en la sección 2 presentamos la motivación de esta búsqueda. En la sección 3 mostramos los resultados obtenidos y analizamos los trabajos encontrados más destacados. En la sección 4 enumeramos los requisitos obtenidos para modelar awareness. Finalmente la sección 5 presenta conclusiones y líneas de trabajo futuro.

## 2 Motivación

Desarrollar aplicaciones colaborativas es una tarea compleja ya que estos sistemas mantienen a un conjunto de actores/usuarios interactuando con un modelo compartido. Específicamente, es complicado modelar la funcionalidad de awareness que muestra lo que está pasando con los otros usuarios mientras colaboran. Contar con métodos, procesos o lenguajes que guíen la construcción de funcionalidad de awareness es fundamental para sustentar las prácticas repetibles, reusables y técnicas que organizan el desarrollo y mejoran la calidad de los productos.

Los recursos de Ingeniería de Software que estamos interesados en encontrar, tienen que ser utilizados para la construcción de software colaborativo que contenga awareness. En nuestra investigación no consideramos aquellos trabajos que ponderen el uso de awareness en otros dominios como educación o medicina y no focalicen el desarrollo de sistemas colaborativos.

Para realizar nuestra compilación de trabajos, definimos el siguiente texto de búsqueda que contiene las palabras claves que se utilizan en las publicaciones científicas:

**Texto de Búsqueda: (Groupware OR “Collaborative system”) And (Model\* OR Design OR “Software Design” OR “Conceptual Model”) And (Awareness)**

Realizamos búsquedas utilizando los servicios del Ministerio de Educación de Argentina, servicios de ACM digital library y el meta-buscador SCOPUS. Luego de algunas comparaciones entre los resultados obtenidos decidimos trabajar con el resultado brindado por SCOPUS en Octubre de 2015, ya que incluye al de ACM y tiene una representación más amplia que el buscador del Ministerio.<sup>1</sup>

En una búsqueda preliminar se identificaron trabajos que no se referían al desarrollo de software, sino que surgieron algunos que utilizan awareness en algunas disciplinas como Juegos o KMS (Knowledge Management System), entre otras. Por lo tanto, la pregunta nro. 1 nos permitió realizar una primera clasificación para descartar trabajos que no trataban sobre desarrollo de software. Luego distinguimos aquellos que aportan recursos de Ingeniería de Software y los clasificamos en los siguientes grupos, según lo que proponen:

- Especificación textual o semi-formal: ¿El trabajo propone utilizar alguna descripción gráfica, textual o semi-formal para modelar el awareness?. En este grupo incorporamos aquellos que utilizan por ejemplo, notaciones gráficas, guías y formularios.
- Modelo Abstracto: ¿El trabajo propone algún modelo abstracto para representar el awareness?. Por ejemplo encontraremos, modelos conceptuales, metamodelos y ontologías.

### 3. Análisis de los resultados

Usando el texto de la búsqueda se encontraron 134 potenciales trabajos primarios que se hallaron a partir del título, las palabras claves y el contenido del resumen. En esta revisión inicial se excluyeron 2 trabajos que estaban duplicados y 6 que eran proceedings eliminando 8 de los 134 iniciales. Los restantes 126 pasaron a una segunda revisión donde obtuvimos una copia completa de cada uno de ellos, realizando un análisis más detallado en base a preguntas que nos permitan clasificarlos en distintos grupos.

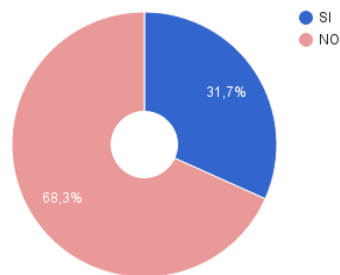
Particularmente, necesitamos identificar trabajos que se refieran a la Ingeniería de Software excluyendo a aquellos correspondientes a otras disciplinas. Para ello se utilizó la pregunta: “¿A qué área se aplica el awareness en los sistemas colaborativos propuestos?”. Quedaron entonces para analizar 82 (el 65,1%) trabajos que tienen por

---

<sup>1</sup> Cabe aclarar que se utilizarán términos en inglés ya que las palabras claves con las cuales están indexados los trabajos están en ese idioma.

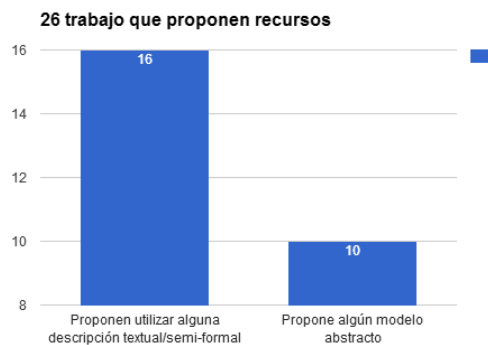
objeto construir Software Colaborativo con awareness. El informe con detalles del análisis se encuentra en: [goo.gl/LPQY6k](http://goo.gl/LPQY6k)

El objetivo es verificar si estos trabajos presentan algún recurso de ingeniería de software para modelar, diseñar, concebir o implementar el awareness en sistemas colaborativos. Hay 26 trabajos que presentan recursos para los desarrolladores de Sistemas Colaborativos y 56 trabajos no los presentan, como se muestra en la Fig.1.



**Fig. 1.** Trabajos que introducen recursos para los desarrolladores.

Seguidamente analizamos los 26 trabajos que son de nuestro interés que están divididos en dos grupos. Por un lado tenemos 16 que proponen describir el awareness de manera textual o de una manera informal o semi-formal. Entre ellos aparecen algunos *checklist* a tener en cuenta en la incorporación de awareness durante el análisis o desarrollo de los Sistemas Colaborativos. Por otro lado, surgen guías para escribir los requerimientos. Incluso hay trabajos que presentan algoritmos para implementar la distribución de información que involucra el awareness. Llegamos finalmente a los 10 trabajos que presentan algún modelo abstracto (ontología, metamodelos o modelos conceptuales) y que además incluyen awareness. Esto nos ayudará a comprender mejor la relación de este concepto con los elementos propios de los sistemas colaborativos.



**Fig. 2.** Distribución de los trabajos que aportan recursos para los desarrolladores.

También nos permitirá clasificar los distintos tipos de awareness que aparecen en estos sistemas. En la siguiente sección analizamos los trabajos que nos interesa destacar. En la Fig. 2 se muestra la distribución de estos trabajos.

### 3.1. Análisis de los Trabajos seleccionados

Consideramos primero un conjunto de trabajos que presentan una notación para especificar los requerimientos de los sistemas colaborativos. Los trabajos comienzan con la presentación del lenguaje CSRML [4] (Collaborative Systems Requirements Modeling Language) como una extensión de I\*. El Lenguaje presenta un conjunto de elementos básicos de los sistemas colaborativos (*goal, role, actor, task, awareness*) y relaciones entre ellos.

Los autores continuaron durante los años 2012 y 2013 realizando diversos experimentos [5] y comparaciones con diferentes técnicas *Goal Oriented* (NFR framework, i\* and KAOS) de especificación de requerimientos para concluir que el CSRML era necesario y construyó el Editor CSRML tool 2012 como una extensión de Visual Estudio [6].

En el 2014 realizaron un estudio de usabilidad del CSRML tool 2012 [7] a través de un experimento con estudiantes que utilizaron la herramienta para responder un cuestionario para evaluar la usabilidad del Editor.

Del análisis de los trabajos de estos autores, no se encontró un metamodelo donde pueda estar basado el lenguaje CSRML. Los conceptos abstractos que se describen son "elementos" y "relaciones". Los elementos incluidos en el lenguaje son: *goal, task, role, resource* y *awareness* cuando en general los sistemas colaborativos suelen incluir otros elementos básicos como *session, tool* y *workspace*.

El lenguaje propuesto no tiene posibilidades de especificar procesos colaborativos. No es posible expresar qué tareas se pueden realizar en cada momento y tampoco se puede determinar que sesiones o actividades colaborativas se pueden realizar antes que otras o qué condiciones se pueden dar para que una sesión se active. Al no incluir procesos colaborativos, no se puede brindar información de awareness relacionada con ellos.

Respecto al CSRML tool 2012 [6], se observa que la herramienta no está basada en estándares y no está integrada con el lenguaje estándar UML, lo que impide utilizar herramientas relacionadas con UML tales como plugins, editores o generadores de código.

El trabajo [8] presenta una revisión de propuestas científicas que brindan diversas clasificaciones de awareness. Comenta distintas definiciones y concluye el análisis puntualizando las siguientes características:

- El awareness es un concepto multifacético. Distintos investigadores han propuesto clasificaciones de awareness (social awareness, mutual awareness, task awareness, workspace awareness, etc.)
- Un sistema colaborativo que provea soporte para awareness deberá

responder a un conjunto de requerimientos relacionados con una lista de preguntas que sirven para identificar tipos de awareness definidos por el autor:

- *social awareness*: Que se refiere preguntas vinculadas con el usuario en relación a su pertenencia a distintos grupos. Entre las preguntas se encuentran: ¿Qué roles otros miembros del grupo pueden asumir? o ¿Cómo puedo interactuar con otros miembros del grupo?
- *task awareness*: Se refiere a preguntas relacionadas con el usuario y las tareas que realiza. Por ejemplo el autor pregunta: ¿Qué pasos tengo que realizar para completar esta tarea? O ¿Cuánto tiempo se requiere para una tarea?
- *workspace awareness*. Aquí el autor se refiere a qué nos preguntamos sobre el usuario en relación a su ubicación dentro del sistema. Por ejemplo ¿Quiénes están aquí o en donde están los usuarios? O ¿Qué cambios hubo desde la última vez que estuve aquí?

El trabajo define “*situation/context awareness*” como la suma de los tres tipos de awareness identificados. A partir de esta definición los autores analizaron propuestas de investigación y resumieron los siguientes conceptos claves frecuentes en los modelos de sistemas colaborativos con awareness.

- *Context element*: Es modelado como un conjunto de elementos compartidos.
- *Task and Activity*: Esto describe qué se espera que se haga o qué es lo que se está realizando para completar una interacción.
- *Resource*: Describe un elemento del contexto. Este elemento es usado durante o para completar una interacción.
- *Interaction*: Es la interacción entre sujetos y objetos usando herramientas colaborativas o es la interacción social a través de la definición de reglas y división de tareas en la comunidad que se modela con un Business Process Modeling.
- *Role*: En situaciones normales de trabajo colaborativo el rol es entendido como el papel que le toca representar a un usuario. Por ejemplo el “Lider” podrá participar de las reuniones de comité y podrá asignar tareas a otros roles.

Seguidamente el trabajo FADYRCOS (Framework for dynamic reconfiguration of networked collaborative systems) [9], tiene por objeto soportar colaboración en ambientes distribuidos donde la sesión puede ser implícita, y además incluir nuevos mecanismos para manejar la evolución de las sesiones y cambios de roles. Sus autores han desarrollado un framework que permite el manejo de la sesión y el desarrollo de componentes dinámicas para sistemas colaborativos que cuenta con una arquitectura compuesta por 3 niveles, que son los siguientes:

*Colaboration Level:* El objetivo de este nivel es el de proveer un esquema de colaboración a los miembros que pertenecen a los distintos grupos que se comunican dentro de una sesión. Los elementos de este nivel están representados en el metamodelo genérico de colaboración (GCM). El principal concepto del metamodelo es la *Session*, que cuenta con un conjunto de flujos de datos representados con el concepto *Flow*. Durante las actividades de colaboración los *flows* son intercambiados entre entidades representadas por el concepto de *Node*.

*Application Level:* Este modelo se basa en los conceptos del dominio. En general este modelo es una especialización del GCM, donde se detallan los nodos que van a participar, los grupos que van a intervenir y las sesiones que existen en el dominio que se está modelando.

*Messaging Level:* Para este nivel se cuenta con dos paradigmas de comunicación. El “*Event Based Communication*” y el “*Peer to Peer Communication*”.

El metamodelo tiene componentes abstractas como *Nodes*, *Session* y *Role/Group*. Este modelo resulta interesante para el modelado de la *Session* y los mecanismos de interacción entre los diferentes roles y participantes. A través de un mecanismo de reglas, se van ejecutando transformaciones entre los niveles, utilizando reglas de refinamiento, selección y adaptación de contexto.

El trabajo [9] no tiene posibilidades de especificar procesos colaborativos. Tampoco se puede determinar que sesiones se deben realizar antes que otra o qué condiciones se pueden dar para que una sesión se active.

Respecto a [10], la propuesta presenta una ontología de awareness para el modelado de sistemas colaborativos. Primero introduce algunos fundamentos teóricos sobre sistemas colaborativos y awareness a partir de una revisión de trabajos. De la revisión realizada, se destaca la contribución de Gutwin y Greenberg (2002) que resulta una de las más destacadas en el campo de la teoría de awareness. Estos autores introducen un framework que define e identifica 10 elementos de awareness. Cada elemento responde a algunas preguntas que sirven para identificar el tipo de awareness que se necesita en cada caso. Las preguntas asociadas con cada tipo de awareness están basadas en la clasificación “*Who, What, Where*” de Gutwin y Greenberg. Además estos autores definen 6 elementos adicionales relacionados con acciones ya realizadas. Así surgen los siguientes tipos de awareness: *location history*, *action history*, *presence history*, *event history*. Previamente Gutwin y Greenberg habían definido otras clasificaciones como “*informal awareness*”, “*social awareness*”, “*group-structural awareness*” y “*workspace awareness*”.

La ontología propuesta se basó en los siguientes tipos de awareness:

- *Group awareness:* Conciencia e información sobre diferentes aspectos del grupo y sus miembros.
- *Workspace Awareness:* Percepción de la interacción de los usuarios en el espacio de trabajo.

- En base a estos dos tipos, los autores presentan los siguientes mecanismos para manipularlos: *Awareness Mechanism*: Mecanismos que informan la ocurrencia de una acción que provee el awareness.
- *Access awareness mechanism*: Este mecanismo informa la acción de acceso de un usuario a distintos elementos del sistema.
- *Entity-Relationship awareness mechanism*: Mecanismo de awareness que informa la acción sobre una entidad o sobre una relación.
- *Communication/Coordination awareness mechanism*: Mecanismo de awareness que informa la acción de comunicación o coordinación.

El trabajo separa la problemática del awareness dentro de los sistemas colaborativos en 2 aspectos. Por un lado aparece el concepto de awareness con dos subclases *Workspace* y *Group Awareness*. Por otro lado introduce el concepto de mecanismo de Awareness que representa el modelado de cómo el awareness va a ser manejado en relación a qué acción activa o dispara el awareness. Por ejemplo cuando se quiere informar que un usuario accede a alguna sesión se utiliza el *Access Awareness Mechanism* para informar a los usuarios involucrados.

Luego el artículo [11] es una continuación del trabajo de Ontología, ya que fue realizado por los mismos autores. El trabajo propone un metamodelo de awareness que define el *Group Awareness* y el *Workspace Awareness*. También tiene un conjunto de *Awareness Support Widgets* para soportar este tipo de awareness.

No se menciona cómo relaciona el awareness con los conceptos tradicionales de los sistemas colaborativos (*Workspace, Session, Collaborative Process, Tool, etc.*). Asimismo hay ciertos tipos de awareness que no son soportados por el metamodelo como por ejemplo los awareness relacionados con las actividades que realizan los usuarios o poder mostrar el grado de avance de una tarea colaborativa.

Finalmente el trabajo [12] presenta un metamodelo llamado *Context Metamodel*. Aunque no es un trabajo específico de awareness, el metamodelo propuesto incluye algunos conceptos interesantes para especificar el awareness en los sistemas colaborativos. El metamodelo tiene como clase principal la clase *ContextualElement*. También tiene un conjunto de enumerativos que resultan interesantes. Primero está el *ContextType* tiene los tipos de awareness que propone Gutwin y Greenberg “*who*” “*what*” “*where*” “*when*” “*why*”. Otro enumerativo es el *UpdateType* que determina cuando se actualiza el awareness y el *AcquisitionType* para detallar el mecanismo por el cual se adquiere la información de awareness.

El metamodelo propuesto en [12] no tiene posibilidades de especificar procesos colaborativos; por lo tanto no se puede brindar información de awareness relacionada con ellos.



#### 4. Requisitos para el modelado conceptual de awareness

Como vimos en esta revisión son pocos los trabajos que presentan modelos abstractos para que los diseñadores de software incluyan el concepto de awareness en sistemas colaborativos. Una de las primeras observaciones realizadas indica que los trabajos analizados en la sección 3 utilizan los mismos conceptos principales que son los siguientes: “*Shared Object*”, “*Tool*”, “*Session*”, “*Workspace*”, “*User/Role*”, “*Group*” lo cual nos permite enunciar el primer requisito.

**Requisito 1:** Contar con un modelo conceptual que contenga los conceptos que intervienen en los sistemas colaborativos.

Continuando con el análisis vemos que estos elementos tienen relación con el concepto de awareness. Por ejemplo una de las preguntas que se utilizan en el trabajo [8] para identificar un awareness específico es: “*What roles will the other members of the group assume?*” en donde aparece el concepto de *Role*. Otro ejemplo el awareness relacionado con el espacio de trabajo donde el autor se pregunta: *How can I help other participants to complete the project? or What are they doing? or where are they?*; donde vincula el espacio con el awareness. Con esta observación dando origen al segundo requisito para el modelado de sistemas colaborativos.

**Requisito 2:** Contar con un modelo que permita expresar el awareness vinculado a los conceptos que intervienen en el sistema. El modelo tiene que cubrir las distintas alternativas que presenta el awareness (*workspace awareness, social awareness, group awareness*, entre otras).

Analizamos ahora las tareas colaborativas que realizan los usuarios en el sistema y que en los trabajos aparecen con el concepto de “*Task*”. Alrededor de este concepto se organizan las actividades que realizan los usuarios de forma colaborativa. Estas tareas requieren algunos recursos de ingeniería de software para poder modelarlas correctamente. Por un lado se requiere algún mecanismo para darle un orden lógico a las tareas. Por ejemplo necesitamos indicar qué tarea se realiza antes que otra o si algunas tareas se pueden hacer en paralelo. Esto nos permitió enunciar el tercer requisito.

**Requisito 3:** Contar con un modelo de proceso colaborativo como un conjunto de actividades colaborativas (sesiones) que desarrollan los usuarios.

A partir de este requisito es necesario contemplar el awareness relacionado con los procesos colaborativos. Por ejemplo un awareness interesante podría ser informar el grado de avance que un grupo de usuarios tiene en un proyecto o qué tareas están en curso y cuáles están por iniciarse. Para ello es necesario que el modelo permita especificar este tipo de awareness y relacionarlo con los procesos colaborativos, dando así origen al requisito 4.

**Requisito 4:** Permitir asociar información de awareness a los procesos colaborativos.

Por otro lado, es sabido que las tareas colaborativas (sesiones) en los sistemas colaborativos permiten que los usuarios colaboren utilizando distintas herramientas y

objetos compartidos. Las sesiones no son estáticas, sino que van cambiando de estado cuando los usuarios van realizando sus actividades. Por ejemplo, en una sesión donde un usuario está exponiendo un tema, otro usuario puede querer hacer preguntas. Si el sistema permite que los usuarios pidan autorización para preguntar entonces el expositor podrá dar la palabra a quien la haya pedido o continuar con la exposición. Como podemos ver la sesión está en un estado cuando el expositor tiene la palabra y en otro cuando otro usuario está preguntando. Estos distintos estados permiten ejecutar distintos grupos de acciones. El modelado de estos diferentes estados describe cómo el sistema se va modificando a partir de la interacción de los usuarios. Arribamos así al enunciado del requisito 5.

**Requisito 5:** Permitir modelar los distintos estados por lo que pasa una sesión de interacción (protocolos de interacción).

Vemos a partir de este requisito que es necesario modelar el awareness relacionado a los distintos estados de la sesión. Siguiendo el ejemplo anterior cuando un usuario pide la palabra, el sistema debería informar a los otros participantes que alguien quiere preguntar. Cuando el expositor autoriza a preguntar, los otros usuarios deberán recibir información de awareness que les indique qué usuario está preguntando. Esta dinámica de cambios de estado genera la necesidad de expresar awareness específicos asociados a los estados de la sesión. Como consecuencia de esto arribamos al requisito 6.

**Requisito 6:** Incorporar distintos tipos de awareness asociados a los estados por lo que pasa una sesión.

Finalmente, en el desarrollo de software, es importante el uso de estándares industriales. Estos facilitan tanto la comunicación entre desarrolladores como la interacción entre diferentes aplicaciones y productos basados en ellos, que propician el mejoramiento de la calidad tecnológica. Por ejemplo, si el modelo se basa en el metamodelo de UML, se podrían utilizar distintos plugins [13] basados en dicho metamodelo. En consecuencia para describir nuestro sistema colaborativo nos serían útiles editores relacionados con UML; lo cual da origen al requisito 7.

**Requisito 7:** utilización de estándares en la construcción de los modelos que intervienen

En la siguiente sección se discute la utilidad de los requisitos obtenidos en esta revisión.

## 5. Conclusiones

En este trabajo se analizaron propuestas científicas que presentan recursos de ingeniería de software para modelar awareness en sistemas colaborativos. De los trabajos analizados vemos que no hay una propuesta que cubra todas las necesidades de modelar el awareness. No hay un modelo que se haya destacado y por ello no existe un estándar por defecto aceptado por la comunidad académica. La mayoría de los trabajos muestran ejemplos de uso de aplicaciones que utilizan la información de

awareness en algún dominio en particular (educación, salud, etc.). Hay pocos trabajos que abordan el modelado del awareness a través de modelos conceptuales, abstracciones o metamodelos. De los 126 trabajos obtenidos inicialmente, solamente 10 de ellos presentaban alguna abstracción que permita conceptualizar el awareness en este tipo de sistemas.

Aparecen algunos trabajos que presentan modelos formales que introducen los conceptos principales de los sistemas colaborativos; esto nos permitió enunciar el primer requisito.

Otros trabajos permiten incorporar awareness a los diseños de los sistemas colaborativos y vincularlo a los conceptos principales. Esto permite modelar distintos tipos de awareness y asociarlos a los conceptos del sistema. Así enunciamos el requisito 2, lo que habilitará el diseño de sistemas colaborativos donde pueda especificarse qué tipo de awareness va a tener cada elemento del sistema. Por ejemplo podemos decir en qué espacio se encuentran los usuarios o qué están haciendo en una sesión de trabajo.

Una de las carencias de los trabajos analizados es la falta de elementos que permitan modelar procesos colaborativos. Algunos permiten incorporar tareas aisladas, pero no es posible darle un orden lógico a las mismas. En consecuencia el requisito 3 de este trabajo enuncia la necesidad de incluir el modelado de procesos colaborativos.

Luego al incorporar awareness a los procesos, surge el requisito 4 permitiendo modelar, por ejemplo, qué información recibirán los usuarios en relación al avance de las tareas colaborativas en los procesos.

Por otro lado vemos la necesidad de modelar la interacción de los participantes dentro de las tareas colaborativas. Particularmente se requiere expresar qué acciones pueden realizar los roles en cada momento. Esto nos permitió elaborar el requisito 5 brindando la posibilidad al diseñador de definir diferentes estados por los que pasan las tareas colaborativas. Estos estados y el cambio entre ellos necesitan mantener a los participantes actualizados sobre la dinámica de cada tarea. Esta información de awareness fue cubierta por nuestro requisito 6.

Finalmente vemos la necesidad de desarrollar herramientas de diseño que implementen los requisitos mencionados. Siguiendo la idea de facilitar el intercambio de modelos entre desarrolladores y considerando que el uso de estándares permite este intercambio, es que se incorporó el requisito 7.

Como continuidad de este trabajo se construirá un modelo conceptual que cumpla con los requisitos enunciados. A partir de este modelo se definirá un lenguaje específico de dominio que permita describir sistemas colaborativos con awareness. Basándonos en este lenguaje se desarrollará un conjunto de editores para describir distintos aspectos de los sistemas colaborativos. Particularmente, se podrán especificar procesos colaborativos, protocolos e información de awareness que serán parte del sistema. Finalmente se espera desarrollar herramientas que automaticen la generación de código y que permitan transformar los modelos especificados en prototipos ejecutables.

## Referencias

1. Ellis, C.A., Gibbs, S.J., Rein, G.L., Groupware: some issues and experiences, in: *Communications of the ACM*, 34(1) (1991).
2. Gutwin, C., Greenberg, S., & Roseman, M. (1996). Workspace awareness in real-time distributed groupware: framework, widgets, and evaluation. In *People and computers XI (Proceedings of the HCI'96)*.
3. Gutwin, C., & Greenberg, S. (2002). A descriptive framework of workspace awareness for real-time groupware. *CSCW Journal*, 11, 411–446.
4. Teruel M.A., Navarro E., Lopez-Jaquero V., Montero F., Gonzalez P.: An extension of i; to model CSCW requirements applied to a collaborative conference review system. *CEUR Workshop Proceedings*. (2011).
5. Teruel M.A., Navarro E., Lopez-Jaquero V., Montero F., Gonzalez P.: Analyzing the understandability of Requirements Engineering languages for CSCW systems: A family of experiments. *Information and Software Technology*. (2012).
6. Teruel M.A., Navarro E., Lopez-Jaquero V., Montero F., Gonzalez P.: CSRML Tool: A visual studio extension for modeling CSCW requirements. *CEUR Workshop Proceedings*. (2013).
7. Teruel M.A., Navarro E., Lopez-Jaquero V., Montero F., Gonzalez P.: Comparing goal-oriented approaches to model requirements for CSCW. *Communications in Computer and Information Science* (2013).
8. Belkadi F., Bonjour E., Camargo M., Troussier N., Eynard B.: A situation model to support awareness in collaborative design. *International Journal of Human Computer Studies*. (2013).
9. Kamoun A., Tazi S., Drira K.: FADYRCOS, a semantic interoperability framework for collaborative model-based dynamic reconfiguration of networked services. *Computers in Industry* (2012)
10. Gallardo J., Molina A.I., Bravo C., Redondo M.A., Collazos C.A.: An ontological conceptualization approach for awareness in domain-independent collaborative modeling systems: Application to a model-driven development method. *Expert Systems with Applications*. (2010)
11. Gallego F., Molina A.I., Gallardo J., Bravo C.: A conceptual framework for modeling awareness mechanisms in collaborative Systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. (2011)
12. Vieira V., Tedesco P., Salgado A.C.: Using a metamodel to design structural and behavioral aspects in context-sensitive groupware. *Proceedings of the 2010 14th International Conference on Computer Supported Cooperative Work in Design, CSCWD (2010)*
13. Eclipse Modelling Project. <http://www.eclipse.org/modeling/> (Accedido Abril de 2016).