

Diseño e implementación de una arquitectura consciente del contexto para sistemas groupware

No Author Given

No Institute Given

Abstract. Los Groupware son sistemas que se encargan de apoyar el trabajo colaborativo de un grupo de personas, tradicionalmente operan en un dominio específico que depende de las actividades que realice el grupo y de los objetivos que quieran cumplir, para mejorar este apoyo se les dota con la característica de ser conscientes del contexto. En el presente trabajo se hace una revisión de las arquitecturas conscientes del contexto para diseñar una dirigida a Groupwares en general, la arquitectura parte de un prototipo implementado parcialmente en el que se divide el manejo de la información contextual en 3 fases: la recuperación de la información, gestión de los datos recuperados y el procesamiento y uso del contexto obtenido. Inherente a estas tareas están las de implementar una ontología de contexto, un método para la inferencia de datos contextuales y la distribución de los resultados obtenidos. La arquitectura se implementará con un Groupware.

1 Introducción

En los trabajos colaborativos un grupo de personas unen sus esfuerzos para alcanzar un objetivo realizando tareas que en ocasiones dependen de otras tareas realizadas por otros integrantes del grupo. El trabajo colaborativo asistido por computadora¹ es el área que estudia los sistemas computacionales que ayudan a los grupos de trabajo a mejorar la coordinación, cooperación y comunicación que hay entre los integrantes del grupo de trabajo. Entre los sistemas que estudia el CSCW están los Groupware, que son sistemas que funcionan como medio para que los usuarios interactúen entre sí y puedan llevar a cabo tareas en conjunto. Para facilitar la interacción de estos usuarios con el sistema y con otros usuarios necesitan tener un grado aceptable de conciencia de la situación que los rodea a ellos como individuos, de los miembros del grupo del que forman parte y de las tareas que se están llevando a cabo.

Debido a esto surge la necesidad de dotar a los sistemas con el conocimiento necesario para poder proporcionar tales niveles de conciencia al usuario a partir de esta necesidad se desarrollan sistemas conscientes del contexto los cuales tienen conocimiento de la situación del ambiente en el que operan, así, los Sistemas Groupware Conscientes del Contexto² tienen ambas características:

¹ CSCW por sus siglas en inglés (*Computer Supported Collaborative Work*)

² CAGS por sus siglas en inglés (*Context Aware Groupware Systems*)

apoyan el trabajo colaborativo de un grupo de usuarios y tienen conocimiento del contexto que los rodea y opera en consecuencia a esta información, para esto se crean arquitecturas o marcos de trabajo conscientes del contexto que puedan ser integradas a un Groupware y así puedan funcionar como un CAGS.

El problema de estos sistemas es que están basados en escenarios particulares satisfaciendo necesidades específicas del problema; por ejemplo, Meeting Reminder Agent[3], toma el tiempo de distracción de actividades, la ubicación, y el sonido en un campus para avisar al usuario de los lugares donde se están llevando a cabo reuniones, y sugerir de acuerdo con los intereses del usuario, conferencias que se lleven a cabo. Otro caso es Portable Help Desk (PHD) [3] que es una aplicación que toma en cuenta la cercanía de los miembros de un grupo y su disponibilidad para poder brindar apoyo a otros miembros y carecen de generalidad para poder ser aplicados en ambientes con un contexto diferente al que han sido desarrollados. Por último CO2DE [16] que se concentra en la edición colaborativa asíncrona de diagramas y resolución de conflictos que mantiene el contexto individual de los integrantes del proyecto y evita traslapar ediciones manteniendo a cada contexto de los individuos en una rama diferente al proyecto original.

Se observa en los tres ejemplos que cada aplicación usa datos contextuales propios del escenario, y si se intenta integrar el modelo de uno de ellos a otro, haría falta añadir las características que el anterior no tenía, lo que dificulta el modelado de soluciones. Por lo tanto son requeridos servicios fundamentales de contexto genérico para hacer de la conciencia del contexto una tecnología factible que puede ser fácilmente incorporada a una variedad de software (Pascoe, J., Ryan, N., & Morse, D., 1999).

En el presente trabajo se propone una arquitectura consciente del contexto, que parte de otro proyecto de investigación[14], constituida por tres niveles: recuperación de información contextual, manejo y gestión de datos y uso del contexto. Se hace especial énfasis en el uso de contexto y en la forma en que se aplican reglas de inferencia sobre un grupo de datos contextuales relacionados entre sí para obtener un resultado, ya sea un comando ejecutable en el groupware o información importante para la tarea del usuario.

2 Trabajos previos

El cómputo consciente del contexto es un término discutido por primera vez en el trabajo de Schilit y Theimer [15] como software que se adapta de acuerdo al contexto, esto limita la definición a aplicaciones que son informadas sobre el contexto y se adaptan a él, no dejando en claro qué tipo de adaptación es la que realiza. En investigaciones más recientes, Dey [5], define computación consciente del contexto como un sistema que usa el contexto para proporcionar información relevante y/o servicios al usuario, donde la relevancia depende de la tarea del usuario. La definición de Dey se puede ver reflejada en el ejemplo del sistema guía de turistas, donde la información dada por el sistema es de interés para los usuarios y la actividad que realizan, como por ejemplo, notificar de eventos

próximos, o recomendar actividades o lugares a los turistas. En el caso de los videojuegos, el sistema ejecutará instrucciones para ir aumentando la dificultad o el nivel del juego conforme el usuario va incrementando su habilidad, esto cumple la segunda propiedad de la definición de Dey que es ejecutar comandos para adaptarse al contexto.

Una arquitectura consciente del contexto debe de cumplir con las siguientes características[6]:

- Acceso distribuido a la información contextual.
- Soporte multiplataforma y multilinguaje.
- Interpretación contextual.
- Agregación de información contextual.
- Independencia y persistencia de widgets contextuales.
- Almacenamiento histórico de la información contextual.

Además de estas características se deben de cumplir los siguientes requerimientos[8]: una formalización de contexto para delimitar los datos contextuales y facilitar la distinción de parámetros contextuales, una categorización de datos para reducir la complejidad de su manutención; el segundo requerimiento son reglas de adaptación: la adaptación del contexto debe ser vista como un conjunto de reglas que controlan y anticipan el cambio de contexto que puede ocurrir en el ambiente, por lo tanto en la construcción de reglas de adaptación, el número de parámetros contextuales es grande y así, es evidente que no se pueden enumerar todas las posibles situaciones que van a ocurrir, es por esto que se requiere un método para construir reglas de adaptación que pueda manejarla diversidad de posibles situaciones que se construyen en base de esos parámetros contextuales.

Muchas arquitecturas se han propuesto para poder soportar sistemas conscientes del contexto, la siguiente tabla hace una comparación de los elementos y capas de algunas arquitecturas conscientes del contexto, entre las cuales se encuentran la arquitectura base para el presente trabajo que usa un modelo contextual colaborativo clasificado en tres categorías: elementos cohesivos, elementos interactivos y elementos afectivos. La arquitectura de Dey[6] usa widgets para la captura de datos contextuales y servicios de agregación de contexto así como servicios de distribución y razonamiento contextual. En el marco de trabajo de Kamoun [13] se reconfiguran servicios para adaptarlos a situaciones que cambian dinámicamente. Decouchant [4] divide su arquitectura en tres capas: la capa de espacio de trabajo, la de adaptación y la de detección de información contextual. Guerman [11] que propone una arquitectura orientada a sistemas de aprendizaje electrónico. En la figura 1 se comparan algunos elementos que poseen dichas arquitecturas, entre ellos se encuentran la presencia de capas como adquisición, manejo y distribución de datos contextuales, persistencia de datos y su reuso, apoyo con vías de comunicación, el uso de widgets como comunicadores entre el sistema y la arquitectura, manejo de sesiones, esquemas conceptuales de colaboración, agregación de datos y la representación de espacios de trabajo como parte de la arquitectura.

Autor	Adquisición de datos	Manejo de contexto	Representación contextual	Razonamiento contextual	Distribución de información	Persistencia de datos	Reuso de contexto	Comunicación	Widgets contextuales	Manejo de Sesión	Esquema de Colaboración	Agregación de datos	Representación de estado de trabajo
Montané E-Jiménez, L.G <i>et. al.</i> (2013) [13]	x	x	x	x	x	x							
Dey, A.K. <i>et. al.</i> (1999) [6]	x		x	x		x			x				
Kamoun, E. <i>et. al.</i> (2012) [12]				x		x		x		x	x		
Decouchant, D. <i>et. al.</i> (2013) [4]	x	x		x	x	x			x			x	x
Guermah, H., <i>et. al.</i> (2013) [10]	x	x	x	x		x	x						

Fig. 1. Comparación de arquitecturas que soportan consciencia del contexto[14][6][13][4][11]

3 Modelo de Contexto

En la literatura la mayoría de modelos contextuales comparten elementos que comunmente describen cuatro factores típicos de contexto [1]: ubicación, identidad, estado de las personas, grupos y objetos físicos y virtuales. Algunos de estos modelos difieren en la forma de ser representados, o en el ámbito en el que se aplican, desde representaciones de espacios de trabajo [4], meta modelos que describen el comportamiento de un grupo de usuarios [14] [2] [12], hasta aquellos modelos enfocados a las actividades de un usuario y su comportamiento [13][10][11][7]. Para el presente trabajo se hace uso de un meta modelo contextual para el modelado de sistemas groupware basado un modelo propuesto anteriormente[14], en el cual se pueden encontrar dos categorías de elementos, interactivos y cohesivos. Entre los interactivos se encuentran *actores*, que son los usuarios del sistema, *objetos* que se usan en *tareas* o que son producto de ellas, *categorías* que clasifican a los actores, objetos y tareas según sus atributos, por último están los roles del objeto y del actor los cuales son asignadas a una tarea para establecer el rol que va a tener el actor u objeto involucrado en dicha tarea. Entre los datos cohesivos se encuentran las *comunidades* que son el conjunto de actores con una *actividad* en común, estas actividades pueden tener varias *metas* las cuales se cumplen realizando tareas. Por último se encuentran las reglas que son sentencias en un lenguaje definido para poder inferir las interacciones que suceden en el groupware. En la figura 2 se muestra un diagrama de los elementos de este modelo y sus relaciones.

Con este meta modelo contextual se pueden describir groupwares definiendo cada uno de estos elementos a partir de interacciones, una vez dado de alta un caso junto con sus reglas se instancia un modelo que representará al sistema colaborativo y almacenará las variables contextuales que este transmita a la arquitectura. Cabe mencionar que en este metamodelo los elementos cuentan con 3 atributos principales, un identificador del objeto, un nombre descriptivo, y una lista de atributos almacenados en formato JSON, lo que vuelve flexible la forma de registrar casos de estudio.

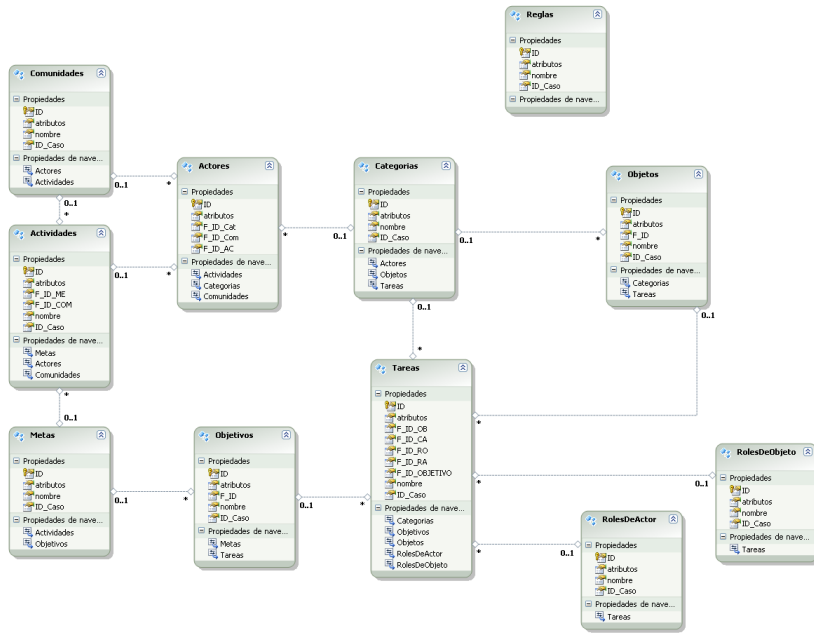


Fig. 2. Meta modelo contextual

4 Caso de estudio

Para el actual proyecto se necesita un groupware al cuál se le pueda acoplar la arquitectura para poder analizar sus datos, en este caso el sistema seleccionado es un videojuego colaborativo de disparos en primera persona: *AssaultCube*. Este groupware en particular tiene las características de ser distribuido y síncrono según la clasificación de Ellis[9], contiene varios tipos de elementos y los modos multijugador son entre equipos en los cuales se requiere de una buena colaboración para cumplir los objetivos de la actividad.

En la Tabla 1 se muestran las interacciones identificadas en el juego, en ellas se encuentran algunos elementos del modelo como pueden ser Actores, Tareas y Objetos, esto nos da la pauta para empezar a diseñar nuestro modelo.

Table 1: Tabla de interacciones detectadas en *Assault Cube*

Interacción	Elementos identificados
Jugador se Mueve	Actor: Jugador; Tarea: Movearse
Jugador salta	Actor: Jugador; Tarea: Saltar
Jugador dispara arma	Actor: Jugador; Tarea: Saltar; Objeto: Arma
Jugador recarga arma	Actor: Jugador; Tarea: Recargar; Objeto: Arma

Continúa en siguiente página...

Table 1 – ... *Continúa de página anterior*

Interacción	Elementos identificados
Jugador dispara arma	Actor: Jugador; Tarea: Saltar; Objeto: Arma
Jugador cambia arma	Actor: Jugador; Tarea: Cambiar; Objeto: Arma
Jugador obtiene mejora de salud	Actor: Jugador; Tarea: Obtener; Objeto: Mejora de salud
Jugador obtiene protección	Actor: Jugador; Tarea: Obtener; Objeto: Protección
Jugador obtiene munición	Actor: Jugador; Tarea: Obtener; Objeto: munición
Jugador envía mensaje de texto	Actor: Jugador; Tarea: enviar; Objeto: Mensaje de texto
Jugador envía mensaje de voz predefinido	Actor: Jugador; Tarea: enviar; Objeto: Mensaje de voz
Jugador elige arma inicial	Actor: Jugador; Tarea: Elegir; Objeto: Arma predeterminada
Jugador cambia rol	Actor: Jugador; Tarea: Cambiar; Objeto: Rol
Jugador se agacha	Actor: Jugador; Tarea: Agacharse
Jugador se suicida	Actor: Jugador; Tarea: Suicidarse
Jugador es eliminado	Actor: Jugador; Tarea: Ser eliminado
Jugador elimina oponente	Actores: JugadorA, JugadorB; Tarea: Eliminar
Jugador reaparece	Actor: Jugador; Tarea: Reaparecer
Jugador captura bandera	Actor: Jugador; Tarea: Capturar; Objeto: Bandera
Jugador regresa bandera a su base	Actor: Jugador; Tarea: Recuperar; Objeto: Bandera
Jugador ve mapa	Actor: Jugador; Tarea: Ver; Objeto: Mapa
Jugador ve puntuaciones	Actor: Jugador; Tarea: Ver; Puntuaciones

En la lista anterior de interacciones se pueden identificar ya algunos elementos del modelo del groupware, por ejemplo, jugador como actor, arma, munición, mapa como tipos de objetos, y el conjunto de ellos como tareas. También a partir de estas interacciones pueden empezar a definirse algunas reglas.

5 Propuesta

La arquitectura que se propone en este trabajo, como ya se mencionó antes cuenta con tres capas: recuperación de datos, gestión de contexto, y uso de contexto. Para poder acoplar la arquitectura primero se tiene que dar de alta un modelo del groupware. Para esto se creó una plataforma para registrar casos de estudio en la que se establece el nombre del caso de estudio y todos sus



Fig. 3. Assault Cube

elementos, una vez creado el meta modelo del groupware se instancia el modelo y se administran las interacciones para poder relacionar sus elementos, esto se hace en una segunda plataforma en la que se toman los elementos del meta modelo y se instancia uno más apegado al groupware, ya con este modelo se pueden capturar los datos contextuales que el sistema va a enviar a la arquitectura.

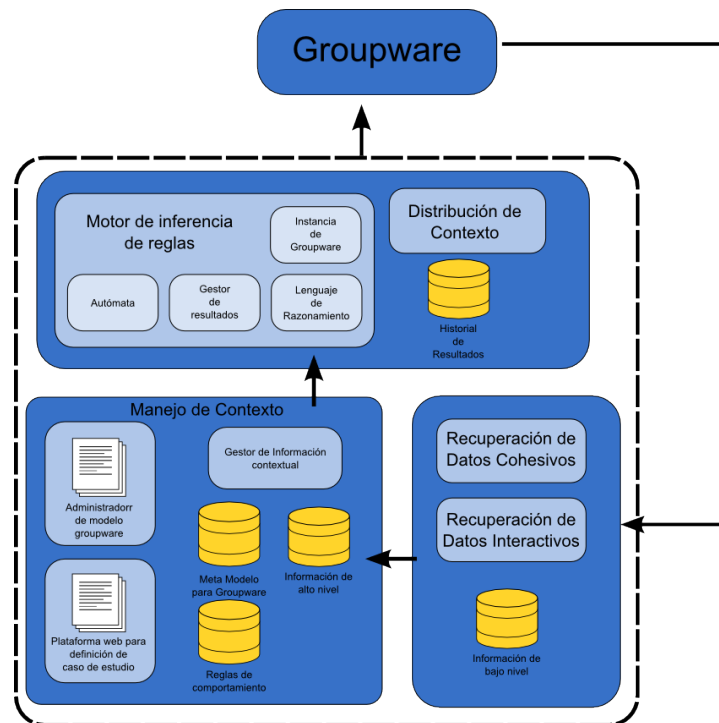


Fig. 4. Arquitectura propuesta

La arquitectura, similar a la arquitectura en que se basó, se divide en tres capas; la capa de recuperación de datos, la capa de gestión de datos y la capa de uso de contexto. En la primera capa, la de recuperación de datos, se captura información contextual enviada por el groupware por medio de servicios web publicados para la comunicación entre la arquitectura y el sistema, estos datos son enviados en un formato específico para que capas superiores puedan procesarla. Seguido de este módulo está el de gestión contextual, este gestor se encarga de registrar, actualizar y recuperar la información contextual en bases de datos, es a este nivel donde el meta modelo y el modelo del sistema se elaboran y donde los datos enviados desde el groupware se registran y son recuperados para inferir resultados en niveles superiores. En la última sección de la arquitectura, uso de contexto, los datos contextuales recuperados son procesados por un motor de inferencia que da como resultado información de interés para los usuarios o comandos de ejecución para la adaptación del sistema a la situación actual. El núcleo de este motor es un autómata que especifica un lenguaje para definir reglas, mismas que van a ser definidas en las plataformas antes mencionadas, dentro del motor se hará una instancia del modelo del groupware con los datos que envíe el sistema, así se puede comparar su estado actual con las reglas de inferencia con el objetivo de ofrecer resultados coherentes al tiempo en el que el sistema se ejecuta. Los resultados obtenidos son gestionados por un administrador de resultados que almacena los datos en una base de datos para mantener registro histórico del comportamiento del groupware. Una vez obtenidos y almacenados los resultados un módulo de distribución de datos se encarga de enviar la información al groupware con la información de entrega necesaria. Este proceso es iterativo, ya que vive por el tiempo en el que el groupware opera.

6 Conclusiones

Esta arquitectura está diseñada para apoyar el trabajo colaborativo, pero su modelo se puede utilizar para brindar consciencia contextual a otro tipo de sistemas, se implementará en el groupware Assault Cube y se probarán sus resultados, con esto se pretende evaluar el desempeño de la arquitectura. Haría falta una interfaz acoplable al groupware para que este se pudiera comunicar con la arquitectura con facilidad. La eficiencia de los resultados depende de las reglas definidas, si las reglas están establecidas para apoyar al grupo de trabajo colaborativo entonces la arquitectura también.

References

1. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: Gellersen, H.W. (ed.) *Handheld and Ubiquitous Computing, Lecture Notes in Computer Science*, vol. 1707, pp. 304–307. Springer Berlin Heidelberg (1999)
2. Alves, P., Ferreira, P.: Radiator: Context propagation based on delayed aggregation. In: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*. pp. 249–260. CSCW '13, ACM, New York, NY, USA (2013)

3. Anhalt, J., Smailagic, A., Siewiorek, D., Gemperle, F., Salber, D., Weber, S., Beck, J., Jennings, J.: Toward context-aware computing: experiences and lessons. *Intelligent Systems, IEEE* 16(3), 38–46 (May 2001)
4. Decouchant, D., Mendoza, S., Sánchez, G., Rodríguez, J.: Adapting groupware systems to changes in the collaborator's context of use. *Expert Systems with Applications* 40(11), 4446 – 4462 (2013)
5. Dey, A.K., Abowd, G.D., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.* 16(2), 97–166 (Dec 2001)
6. Dey, A.K., Salber, D., Futakawa, M., Abowd, G.D.: An architecture to support context-aware applications (1999)
7. Döweling, S., Schmidt, B., Göb, A.: A model for the design of interactive systems based on activity theory. In: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*. pp. 539–548. CSCW '12, ACM, New York, NY, USA (2012), <http://doi.acm.org/10.1145/2145204.2145287>
8. El Ghayam, Y., Erradi, M.: Distributed context management in collaborative environment. In: *New Technologies of Distributed Systems (NOTERE)*, 2011 11th Annual International Conference on. pp. 1–8 (May 2011)
9. Ellis, C.A., Gibbs, S.J., Rein, G.: Groupware: Some issues and experiences. *Commun. ACM* 34(1), 39–58 (Jan 1991)
10. Gallardo, J., Molina, A.I., Bravo, C.: A framework for the design of awareness support in collaborative situations of implicit interaction. In: *Proceedings of the 13th International Conference on Interacción Persona Ordenador*. pp. 7:1–7:2. INTERACCION '12, ACM, New York, NY, USA (2012)
11. Guermah, H., Fissaa, T., Hafiddi, H., Nassar, M., Kriouile, A.: Ontology based context aware e-learning system. In: *ISKO-Maghreb, 2013 3rd International Symposium*. pp. 1–7 (Nov 2013)
12. Hoyos, J.R., Molina, J.G., Botía, J.A.: A domain-specific language for context modeling in context-aware systems. *Journal of Systems and Software* 86(11), 2890 – 2905 (2013)
13. Kamoun, A., Tazi, S., Drira, K.: Fadyrcos, a semantic interoperability framework for collaborative model-based dynamic reconfiguration of networked services. *Computers in Industry* 63(8), 756 – 765 (2012), special Issue on Sustainable Interoperability: The Future of Internet Based Industrial Enterprises
14. Montane-Jimenez, L., Benitez-Guerrero, E., Mezura-Godoy, C.: A context-aware architecture for improving collaboration of users in groupware systems. In: *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom)*, 2013 9th International Conference Conference on. pp. 70–76 (Oct 2013)
15. Schilit, B., Theimer, M.: Disseminating active map information to mobile hosts. *Network, IEEE* 8(5), 22–32 (Sept 1994)
16. Schmidt, K., Bannon, L.: Taking cscw seriously. *Computer Supported Cooperative Work (CSCW)* 1(1-2), 7–40 (1992)