UNIVERSIDAD VERACRUZANA

FACULTAD DE ESTADÍSTICA E INFORMÁTICA

PROTOCOLO DEL TRABAJO:

"DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA CONSCIENTE DEL CONTEXTO PARA SISTEMAS GROUPWARE"

MODALIDAD:

TRABAJO PRÁCTICO TÉCNICO

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:

LICENCIADO EN INFORMÁTICA

PRESENTA:

SANTIAGO DE JESÚS GONZÁLEZ MEDELLÍN

DIRECTORES:

MCA. LUIS GERARDO MONTANÉ JIMÉNEZ

DR. EDGARD IVÁN BENÍTEZ GUERRERO

XALAPA, VER. JUNIO 2014

Resumen

El trabajo colaborativo es una actividad en la que un grupo de personas une sus esfuerzos para alcanzar un objetivo realizando tareas en las que otras personas pueden estar involucradas. El Trabajo Colaborativo Asistido por Computadora (CSCW) [1] es el área que estudia los sistemas computacionales que ayudan a estos 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 se encuentran los Groupware, sistemas que fungen 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, los groupware les proporcionan comunicación e información para que tengan conocimiento de la situación actual con el fin de facilitar la toma de decisiones. Así surge la necesidad de dotar a los sistemas con el conocimiento necesario para poder proporcionar tales niveles de conscienca al usuario, a partir de esta necesidad se desarrollan sistemas conscientes del contexto. Los Sistemas Groupware Conscientes del Contexto (CAGS por sus siglas en inglés)[2] tienen ambas características antes descritas: apoyan el trabajo colaborativo de un grupo de usuarios y soportan el uso de contexto y opera en consecuencia a esta información, para eso se diseãn arquitecturas conscientes del contexto que se integran a un groupware y puedan, entre ambos sistemas, funcionar como un CAGS.

En el presente trabajo se hace una revisión de las arquitecturas conscientes del contexto para analizar sus características y poder diseñar una dirigida a sistemas groupware independiente del escenario en el que se desempeñen, la arquitectura parte de un trabajo antes propuesto[1] en el que se divide el manejo de la información contextual en 3 etapas: la recuperación de la información, gestión de los datos recuperados y el uso y distribución de resultados. Para la generación de CAGS es necesario un mecanismo de razonamiento contextual, el cual pueda generar resultados a partir de información del contexto del sistema, para esto se propone un lenguaje de definición de comportamiento del que se infieren dichos resultados, además se necesita implementar una ontología o clasificación contextual que pueda ser reutilizada en cualquier caso de estudio. Una vez terminado el diseño se implementará la arquitectura con un groupware y se harán pruebas.

Actualmente la mayoría de las arquitecturas elaboradas están orientadas a groupwares con ambientes específicos, es decir, usan el contexto de un escenario en particular, por ejemplo un sistema para procesos empresariales, tomaría en cuenta datos sobre el giro de la empresa y las actividades que se realizan internamente inherentes a las políticas de la organización; estos datos no serían de utilidad para otro tipo de situaciones como en un groupware para desarrollo de software que se centra en los proyectos de desarrollo y que utiliza otro tipo de variables contextuales.

[1] CSCW por sus siglas en inglés(*Comupter Supported Collaborative Work*)

[2] *Context Aware Groupware Systems*

Índice general

Capítulo 1

Introducción

En la actualidad existen actividades o procedimientos que se llevan a cabo en la vida cotidiana que requieren ser desarrolladas de manera conjunta, un grupo de personas que unen sus esfuerzos para realizar diversas tareas. En ocasiones, algunas de estas tareas son soportadas por los sistemas de información siendo un medio por el cual los usuarios interactuan para realizar su cometido. El área que estudia este tipo de sistemas es el Trabajo Colaborativo Asistido por Computadora o CSCW por sus siglas en inglés (*Computer Supported Collaborative Work*), entre los sistemas que estudia el CSCW están los Groupware, sistemas de magnitud organizacional que apoyan a un grupo de trabajo permitiendo la colaboración, comunicación y coordinación de un grupo para alcanzar una meta, Ellis[2] define Groupware como sistemas computacionales que apoyan a grupos de personas ocupadas en una actividad(u objetivo) en común y que proporcionan una interfaz en un ambiente compartido.

Para hacer que la interacción humano-computadora se lleve a cabo de manera natural y transparente, se dota al Groupware con la característica de percibir y trabajar con datos que describan la situación que rodea al grupo, a esto se le conoce como conciencia contextual, una característica que trae consigo el surgimiento de los Sistemas Groupware Conscientes del Contexto (CAGS).

Se entiende por contexto la situación actual que tiene lugar en una actividad realizada por un sujeto o un grupo de entidades. Estas situaciones están definidas por diferentes elementos que responden a las preguntas ¿Quién?, ¿Dónde?, ¿Cuándo? y ¿Qué?. Schilit y Theimer [3] se refieren al contexto como la ubicación, identidades de personas y objetos cercanos, y los cambios en esos objetos. Mientras que la definición de Dey [4] menciona que el contexto es cualquier información relevante sobre las entidades en la interacción entre el usuario y una computadora, incluyéndolos a ambos, una entidad puede ser una persona, lugar u objeto considerado relevante para la interacción entre un usuario y una aplicación.

El cómputo consciente del contexto es un término discutido por primera vez en el trabajo de Schilit y Theimer 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[3]. En investigaciones más recientes se 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[4]. La definición anterior se puede ver reflejada un 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 usuarios. En el caso de un juego de video colaborativo, el sistema puede elevar la dificultad dependiendo del nivel del usuario, así como notificar a los usuarios de las actividades de sus compañeros.

Para que los Groupware Conscientes del Contexto(CAGS por sus siglas en inglés) puedan trabajar con datos contextuales es necesario tener una descripción del ambiente en el que va a trabajar, hace falta especificar los elementos que se deben de tomar en cuenta, por ejemplo, para un sistema de edición simultanea de textos se debe trabajar información como las modificaciones que se han hecho, la fecha de las modificaciones, los permisos de los usuarios tienen para acceder al documento, etc. mientras que para un sistema de guía de turistas se toman elementos como la ubicación del usuario, el lugar(por ejemplo un edificio, o en un espacio abierto), fechas de eventos próximos, etc. Como se puede observar en ambos casos, se usan descripciones diferentes del ambiente del sistema y esto implica tener que desarrollar dos sistemas completamente diferentes, uno para cada caso en específico. Esto se vuelve un problema al momento de desarrollar varios sistemas que requieren el procesamiento de información contextual, ya que hay que cambiar la especificación del contexto cada vez y, en el peor caso, desarrollar desde cero el sistema.

De modo que los usuarios puedan coordinarse, comunicarse y colaborar de forma eficiente, es necesario que los miembros del equipo tengan un alto nivel de conciencia del espacio de trabajo en el que participan, esto se cumple cuando los usuarios se encuentran en el mismo lugar grográfico, ya que pueden interactuar en tiempo real y así mantenerse informados de la actividad de los demás, por ejemplo, en un juego colaborativo [1], cuando los usuarios juegan en una misma habitación pueden comunicarse directamente con sus compañeros, ver sus expresiones; al jugarlo en habitaciones distintas les es más difícil poder captar este tipo de señales. Además de eso, para que el sistema pueda proveer medios de colaboración que permitan a los usuarios comunicarse eficientemente, ellos tienen que interactuar explícitamente con los medios de comunicación que ofrece el sistema, provocando distracciones que afectan la concentración de los usuarios para lograr el objetivo principal. Para dar solución a este tipo de problemas, se proporciona al sistema información contextual de los usuarios.

1.1. Motivación

La comunicación, coordinación y cooperación entre grupos colaborativos de trabajo es muy importante, con su eficiencia aumenta el rendimiento de los usuarios en el trabajo que realizan, fomentando la productividad. Con los groupware se mejora la calidad de operación de estos grupos, y añadiendo la conciencia del contexto a este tipo de sistema, la interacción que los usuarios tienen con los dispositivos o aplicaciones se hace de forma más natural y fluida, permitiéndoles concentrarse en la tarea que están haciendo en lugar de detalles de comunicación y evitando la carga de proce-

samiento de información que el sistema hará por ellos. Con la arquitectura que se va a diseñar, el desarrollo se vuelve menos complejo, evitando que los desarrolladores tengan que volver a programar otras características.

En un ambiente colaborativo los integrantes necesitan estar comunicándose constantemente para poder coordinar sus esfuerzos. En un groupware esto implica que el sistema se vuelva intermediario entre los usuarios, para poder hacer estas interacciones más fluidas el sistema necesita saber cómo ayudar a los usuarios y esto se logra dándole datos del contexto en general y logrando que actúe en beneficio del grupo. Para que una aplicación pueda ser consciente del contexto, debe de ser capaz de adquirir información contextual, gestionarla, y usarla para obtener resultados que permitan ejecutar un comando o mostrar información para el usuario.

Anteriormente en el trabajo de Montané[1] se propuso una arquitectura para apoyar el trabajo colaborativo en los groupware que además evalúa la presencia social de sus usuarios para medir el nivel de colaboración, en cuyos módulos se pueden encontrar los elementos anteriores. La primera capa es de adquisición de datos contextuales en la que cada módulo trabaja con tipos diferentes de datos contextuales; en la capa de gestión de contexto se almacena la información obtenida para acceder a ella más tarde, aquí se guardarán, actualizarán y recuperarán los datos históricos de la aplicación; en la última capa, que es la de uso de contexto, se encuentran dos módulos, uno que procesa y razona la información contextual, y otro que entrega los resultados al sistema groupware.

Se han propuesto modelos y arquitecturas para el desarrollo de sistemas consientes del contexto, sin embargo estos sistemas están basados en escenarios particulares, es decir, satisfaciendo necesidades específicas de un dominio; por ejemplo, Meeting Reminder Agent[5], 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) [5] 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 otra parte CO2DE [6] 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.

Las aplicaciones mencionadas anteriormente usan elementos diferentes del contexto según las necesidades que cubren, y es complicado re-usar los modelos propuestos en cada uno de ellos con otros sistemas. Por ello falta integrar características que parcialmente son incluidos en los trabajos explorados, lo que supone una mayor carga en el modelado de soluciones. Por lo tanto, servicios fundamentales de contexto genérico son requeridos para hacer de la conciencia del contexto una tecnología factible que puede ser fácilmente incorporada a una variedad de software [7]. Para evitar este problema, se propone modelar e implementar una arquitectura orientada a servicios para sistemas groupware conscientes del contexto, cuyo diseño tome en cuenta los aspectos contextuales más generales, que pueda utilizarse en otros ámbitos, reduciendo así tiempo de desarrollo. La arquitec-

tura será probada con un juego colaborativo (un video juego de disparos en primera persona) y se documentarán los resultados para futuras investigaciones.

Actualmente se encuentra desarrollada la capa de adquisición de datos y un módulo para percibir la presencia social de los usuarios, pero aun carece de un mecanismo de razonamiento contextual.

1.2. Contribuciones

El fin principal de este trabajo es diseñar e implementar una arquitectura funcional orientada a servicios que apoyen la creación de sistemas Groupware consientes del contexto, particularmente en el caso de estudio de los video juegos colaborativos, partiendo de un modelo propuesto anteriormente [1]. Para poder llevar a cabo lo anterior se proponen los siguientes objetivos

- Diseñar una arquitectura orientada a servicios para la adquisisión, gestión y uso de información contextual.
- Desarrollar un módulo de razonamiento contextual.
- Integrar un método de razonamiento contextual para ser aplicado en la arquitectura propuesta.
- Validar la arquitectura con el caso de estudio.

1.3. Alcances y Limitaciones

Se pretende proponer una arquitectura genérica orientada a servicios e implementarla en un sistema groupware, se extraerá información contextual y se procesará con un mecanismo también propuesto, en la arquitectura se hará una distribución física y lógica de cada uno de los módulos de la arquitectura y se establecerán formas de comunicación entre ellos. También se implementará una clasificación contextual que abarque los elementos necesarios para definir la situación de un ambiente. A pesar de que la arquitectura y el método de razonamiento está planteada para funcionar en cualquier caso de estudio, se probará por ahora en un videojuego colaborativo de disparos en primera persona, ambiente que nos proporciona equipos que compiten entre si para cumplir objetivos según el modo de juego.

Capítulo 2

Sistemas Groupware y Consciencia Contextual

El trabajo colaborativo tiene un alto impacto hoy en día debido a que muchas actividades que una persona realiza son en grupo [2], así una actividad puede dividirse en tareas que diferentes miembros pueden hacer, esto se logra manteniendo comunicación, cooperación y colaboración entre los miembros del equipo de trabajo. La comunicación se entiende como el proceso de interacción entre personas que incluye el intercambio implícito o explícito de información[8]; la coordinación es el manejo de interdependencias entre actividades realizadas por multiples actores basadas en objetos mutuos que son intercambiados entre actividades[9]; y la cooperación ocurre cuando un grupo trabaja para lograr una misma meta con alto grado de tareas interdependientes, compartiendo la información disponible a través de alguna clase de espacio compartido[9].

El hecho de que múltiples individuos situados en diferentes escenarios de trabajo y situaciones, con diferentes responsabilidades, perspectivas y propensiones interactúen sean mutuamente dependientes en el conducto de su trabajo tienen implicaciones importantes en el diseño de sistemas computacionales dirigidos a apoyar estos esfuerzos [6].

Shmidt et.al. menciona un conjunto de características que se deben de tomar en cuenta para que los sistemas computacionales que estudia el CSCW sean aceptables para los usuarios [6]:

- Las unidades cooperativas son grandes o son parte de conjuntos más grandes.
- Las unidades cooperativas son, por lo general, formaciones de poca duración, emergen para manejar una situación particular y luego se disuelven.
- La afiliación de las unidades cooperativas no son estables y normalmente no determinable. Las unidades cooperativas a veces se intersectan.

- El patrón de interacción en trabajos cooperativos cambia dinámicamente con los requerimientos y restricciones de la situación.
- El trabajo cooperativo es distribuido físicamente en tiempo y espacio.
- El trabajo cooperativo es distribuido lógicamente, en términos de control, es decir, los agentes son semi-autónomos en su trabajo parcial.
- El trabajo cooperativo involucra perspectivas inconmensurables (e.g. profesiones, especialidades, funciones de trabajo, responsabilidades, etc.) así como estrategias incongruentes y motivos discordantes.
- No hay agentes omniscientes en el trabajo cooperativo en escenarios naturales.

2.1. Groupware

Existen diferentes tipos de sistemas groupware, [2] define dos clasificaciones: por espacio-tiempo y por tipo de aplicación. Entre el rubro espacio-tiempo se encuentran aquellos que su funcionalidad permite interacciones cara a cara o interacciones distribuidas, así mismo podemos encontrar aquellos que permiten una interacción en tiempo real o una interacción asíncrona. Además de groupwares síncronos y asíncronos agrega un tercer rubro que combina los dos anteriores, y añade el tipo de comunicación como característica, para la interacción de los usuarios usa el estado de conexión de los usuarios, las interacciones distribuidas son incluidas en la categoría de disponibilidad del usuario[10]. Una clasificación similar es descrita por Johansen unos años atrás tomando en cuenta solamente las dimensiones espacio tiempo mostradas en la Figura 2.1.

	Mismo tiempo	Tiempos diferentes
Mismo Lugar	Interacción cara a cara	Interacción asíncrona
Diferentes Lugares	Interacción síncrona distribuida	Interacción distribuida asíncrona

Figura 2.1: Dimensiones de los groupware por tiempo y espacio [11]

Por otra parte cuando se toma en cuenta el tipo de aplicación para clasificar los sistemas groupware, podemos encontrar los siguientes:

1. Sistemas de mensajes que soportan el intercambio asíncrono de mensajes entre miembros de un grupo.

2. Editores multiusuario que permite a miembros de un grupo crear y editar un mismo documento al mismo tiempo, pueden ser síncronos o asíncronos.

3. Sistemas para soporte de decisiones de grupo y cuartos electrónicos de reuniones, proveen infraestructura para la exploración de problemas no estructurados en un grupo.

4. Cómputo para conferencias: proporciona servicios como medios de comunicación en muchas formas, hay 3 tipos: de tiempo real, teleconferencia, y de escritorio.

5. Agentes inteligentes que son programas computacionales *inteligentes* encargados de tareas específicas.

Además de la clasificación por tipo de aplicacioón, Ellis[2] propone elementos que tienen que ser tomados en cuenta para poder comparar y describir a los sistemas groupware, entre ellos se encuentran el contexto compartido que es el ambiente en el que va a correr el sistema; un grupo de ventanas que son las que se van a mostrar en diferentes pantallas del sistema y se van a compartir; un tele apuntador que es la capacidad de más de un usuario de mover el apuntador al mismo tiempo; vista, es la porción de contexto o ambiente que se va a mostrar en diferentes ventanas del sistema; sesión que son los tipos de interacción del usuario con el sistema y que ya se mencionaron antes (mismo tiempo, mismo lugar; diferente tiempo, diferente lugar, etc.) y los roles que son los tipos de usuarios que hay y los derechos y permisos que tienen con el sistema.

Shmidt [6] menciona varias propiedades que deben de tener los sistemas groupware que a continuación se listan:

- Es social; objeto, sujeto, medios, fines, motivos y necesidades, competencias e implementaciones, están mediadas socialmente.
- Integrantes mutuamente dependientes, necesitan cooperar para terminar el trabajo. Diferente a solo compartir el mismo recurso, sujeto A confía positivamente en la calidad y líneas de tiempo del trabajo de sujeto B y vice versa
- Distribución de tareas qué va a hacer cada individuo, cuándo y dónde.
- Distribución física en tiempo y espacio
- Distribución lógica, en términos de control, en el sentido de que los sistemas son semi autónomos en su trabajo parcial.
- Deben aumentar las capacidades mecánicas y de procesamiento de información del individuo.
- Deben combinar las actividades especializadas de múltiples trabajadores dedicados a las diferentes herramientas especializadas.
- Deben facilitar las aplicaciones de múltiples problemas resolviendo estrategias y heurísticas de un problema dado.

- Deben facilitar la aplicación de múltiples perspectivas y concepciones de un problema dado para adaptarse a naturaleza múltiple del ambiente de trabajo.
- Deben apoyar la auto organización de conjuntos cooperativos, no interrumpir trabajo cooperativo computarizando procedimientos formales.

En un estudio más reciente [8] se encuentra una recopilación de los atributos pertenecientes a un groupware, entre ellas están las dinámicas de trabajo en escenarios de colaboración (comunicación, cooperación, y coordinación), dimensiones temporales y espaciales, características grupales (tipos de tareas grupales, características y tamaño), categorías técnicas de aplicaciones groupware (escalabilidad, software y hardware), y categorías complementarias (e.g. usabilidad). En la Figura 2.2 Cruz et. al. [8] define una taxonomía para groupwares organizando las categorías mencionadas en la literatura general del tema orientada a entidades sociales tales como organizaciones, academias e industrias para obtener ideas sobre dinámicas de colaboración y mejorar la calidad de interacción de estas organizaciones, es por eso que esta taxonomía es llamada sociotécnica.

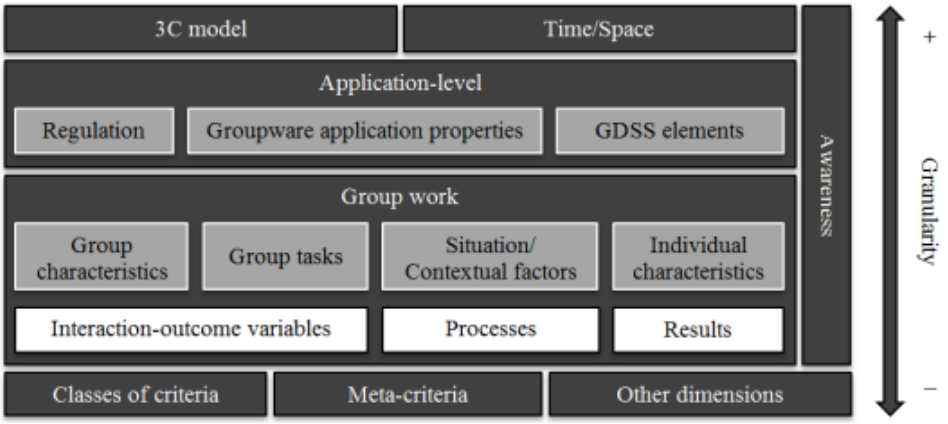

Figura 2.2: Modelo de clasificación de groupware según Cruz [8]

En la Figura 2.2 se observan diferentes capas que muestran las propiedades de los groupware; la primera categoría se encuentra el modelo 3C, que se enfoca en la comunicación, la coordinación y la cooperación. Algunas categorías de relacionadas a la coordinación en la literatura revista por Cruz [8] son: planeación, modelos de control, relaciones entre tareas y subtareas, y manejo de la información, ajuste mutuo, estandarización, protocolos de coordinación, manejo temporal, recursos, o artefactos compartidos producidos durante la sucesión de actividades. Entre las categorías de cooperación se pueden encontrar desde producción (co-autoría), almacenamiento o manipulación de artefactos, hasta concurrencia, control de acceso.

Otro aspecto que toma Cruz en su modelo (Figura 2.2) es tiempo y espacio de donde se observan las siguientes subcategorías: persistencia de sesión, retardo en transmisión de audio/video, reciprocidad y homogeneidad de canales, retardo de envío de mensajes, y espontaneidad de colaboración. En la categoría de nivel de aplicación se identifican tipologías que clasifican a los groupware de acuerdo a

su enfoque en el nivel del grupo. Como subcategoría se encuentran la regulación, que son los mecanismos que permiten a los participantes organizarse en un ambiente compartido; en la literatura algunas de las dimensiones de regulación que se encuentran son: arenas(ubicación); actores(roles, lugares, y posiciones); herramientas(reguladas o no); roles(temáticos o casuales); reglas(restricciones, normas o reglas de trabajo); tipos de interacción; escenarios interactivos; y objetos(medios de comunicación y productos de colaboración). Otra subcategoría del nivel de aplicación son las propiedades de la aplicación groupware entre las que se pueden encontrar arquitectura, cualidades funcionales y de calidad, soporte para procesos grupales, interfaz de colaboración(portales, dispositivos o espacio de trabajo físico), relaciones(colecciones, listas, árboles y gráficas), funciones de núcleo, contenido(texto, ligas, gráficas o flujo de datos), acciones soportadas(recibir, agregar, asociar, editar, mover, borrar, o juzgar), identificación, controles de acceso, mecanismos de alerta, componentes inteligentes, indicadores de consciencia, y plataforma, los elementos GDSS incluyen hardware, software y soporte de personas.

La siguiente capa del modelo se enfoca a los grupos, los cuales son definidos como una "agregación social de individuosçon consciencia de su presencia, conducida por sus propias normas, y soportado por interdependencias de tareas hacia una meta en común en un contexto compartido [12]. Dentro de esta categoría se encuentran las subcategorías: tareas grupales, características del grupo, factores de situación, y características individuales.

Un groupware se puede caracterizar por 3 aspectos complementarios importantes desde la vista del usuario: una descripción de los objetos y las operaciones que se pueden hacer sobre ellos y que están disponibles para los usuarios, una descripción de los aspectos dinámicos del sistema (flujo de control y de datos) y una descripción de la interfaz entre el sistema y los usuarios y entre los usuarios [13]. Este modelo es llamado de coordinación y describe la organización de las actividades realizadas por los usuarios y no por el sistema. Los componentes principales de este modelo ontológico son los objetos y las operaciones. Las operaciones sobre objetos son aspectos importantes que determinan el nivel de contribución de un usuario al trabajo[13], por ej., en un sistema editor de código colaborativo, los usuarios que realicen más operaciones sobre un archivo(actualizar el archivo, crear archivo, eliminar archivo) son los que más aportan al proyecto, sin embargo la desventaja de esto es que no se sabe si la operación que se hizo produjo resultados positivos, así un usuario que realice varios cambios sobre un archivo puede ser señal de que se equivoca mucho en su redacción, otros aspectos importantes en la colaboración son la interacción con miembros del grupo y el ambiente para resolver problemas, tomar decisiones, mantener la afiliación del grupo, etc.

Antunes[14] presenta un framework detallado para evaluar los sistemas colaborativos bajo desarrollo de acuerdo a variables dadas y niveles de desempeño. Considera dos dimensiones: una define el conjunto de variables de evaluación relevantes y el otro se ocupa de los niveles de desempeño que los usuarios evaluados. El tiempo de evaluación es inherentemente asociado con el proceso de desarrollo. Menciona una lista de métodos de evaluación de groupwares son:

- Evaluación heurística de Groupware:(Backer et al 2002) es basado en ocho heurísticas groupware como una lista de características que los groupware deberían de tener.

- Groupware walkthrough: (Pinelle and Gutwin 2002) un escenario es una descripción de una actividad o conjunto de actividades, que incluye a los usuarios, su conocimiento, el resultado esperado y las circunstancias que los rodean.
- Collaboration Usability Analisys (Pinelle et al. 2003): los evaluadores mapean acciones colaborativas a un conjunto de mecanismos de colaboración, o representaciones granulares de acciones colaborativas básicas, que pueden estar relacionadas con elementos en la interfaz de usuario, los diagramas resultantes capturan detalles sobre componentes de tareas, una noción del flujo a través de ellos y la distribución de tareas.
- Groupware Observational User Testing (Gutwin and Greenberg 2000) los evaluadores observan como los usuarios realizan actividades particulares soportadas por un sistema en un ambiente controlado. Los evaluadores monitorean a los usuarios que tienen problemas con una tarea, o piden a los usuarios pensar en voz alta sobre lo que están realizando
- Human-Performance Model (Antunes et al. 2006): los evaluadores descomponen las interfaces en espacios de trabajo compartidos, definen escenarios críticos enfocados a acciones colaborativas para los espacios compartidos y comparan el desempeño del grupo en escenarios críticos para predecir tiempos de ejecución.
- Enfoque de Manejo de conocimiento (Vizcaíno et al. 2005): se mide si el sistema ayuda a los usuarios a detectar flujos de conocimiento y diseminar, reusar y almacenar conocimiento. El proceso de circulación de movimiento se comprende de 6 fases (creación de conocimiento, acumulación, división, utilización, internalización), la evaluación se realiza contestando preguntas relacionadas a cada área.

El trabajo colaborativo es siempre social, en el sentido en el que el objeto y el sujeto, los fines y los medios, los motivos y necesidades están mediados socialmente[6], cada elemento del grupo depende, en parte, del trabajo de algún otro miembro y viceversa, y cuando esto sucede, es importante que ambos elementos del grupo tengan el conocimiento de los avances del otro para poder hacer una estimación de sus propias acciones, es decir, que cada miembro esté conciente de lo que pasa a su alrededor para tomar decisiones adecuadas en cuanto a la actividad que tiene asignada en el grupo.

2.2. Consciencia

El ciclo de colaboración está limitado por la consciencia, que es la percepción del grupo acerca de lo que cada miembro desarrolla, y el conocimiento contextual que tienen sobre qué está pasando entre el grupo [15]. La consciencia en el CSCW se refiere al conocimiento que tienen los individuos sobre sí mismos y sobre el ambiente que los rodea, y en el caso de trabajo colaborativo, el rol que desempeñan en su grupo y el estado de los demás integrantes, es importante que los individuos sepan lo que están haciendo los demás ya que pueden usar ese conocimiento para anticipar las acciones de los otros, y ayudarlos con sus tareas[16], el término awarness(o conciencia en español) fue introducido

por primera vez por Dourish y Belloti[17] definiéndolo como el entendimiento de las actividades de otros, que proporciona un contexto para tu propia actividad, este contexto es usado para asegurar que las contribuciones individuales sean relevantes a las actividades del grupo como un todo, y para evaluar las acciones individuales con respecto a los objetivos del grupo y progresos.

En un estudio realizado por Belkadi et. al. [18] se listan las características que debe de tener la consciencia o el conocimiento, entre ellas se encuentran el tener muchas facetas, es decir, que existen diferentes tipos de conocimiento, el conocimiento está fuertemente enlazado a situaciones colaborativas ya que los colaboradores necesitan información para llevar a cabo algunas tareas o para tomar decisiones; el conocimiento en una situación colaborativa puede incrementar los niveles de confianza entre actores lo que los alienta a compartir información. Así mismo cita algunos requerimientos para permitir la consciencia, para crear un framework que apoye a la conciencia de los usuarios debe de cubrir los siguientes puntos: estar basado en frameworks teóricos sólidos, debe de modelar los elementos de una situación de manera apropiada y sus interacciones para facilitar la conciencia, debe estar basado en conceptos generales.

Además de estas características proponen tres tipos consciencia: social, de las tareas y del espacio de trabajo; estas tres categorías se pueden identificar con las preguntas de la Tabla 2.1:

Cuadro 2.1: Preguntas para entender los tipos de consciencia.

Tipo de consciencia	**Preguntas**
Consciencia Social	¿Qué debo de esperar de otros miembros del grupo? ¿Cómo voy a interactuar con el grupo? ¿Qué rol voy a tomar en este grupo? ¿Qué roles van a tomar los demás miembros del grupo?
Consciencia de las Tareas	¿Qué sé de este tema y la estructura de la tarea? ¿Qué saben los demás? ¿Qué se necesita para completar la tarea? ¿Cómo serán evaluados los resultados? ¿Qué herramientas u objetos se necesitan para completar esta tarea? ¿Cuánto tiempo se necesita y cuánto tiempo hay disponible?
Consciencia del espacio de trabajo	¿Qué hacen los demás miembros del grupo para completar la tarea? ¿Dónde están? ¿Están activos en el espacio de trabajo? ¿Qué harán? ¿Qué hacen actualmente? ¿Qué han hecho? ¿Qué harán después? ¿Cómo los puedo ayudar?

Cruz et. al. también hace su propia interpretación de conciencia [8] y la clasifica en los siguientes rubros:

- Consciencia espacial y atmosférica
- Consciencia de la actividad
- Consciencia de los objetos
- Consciencia humana
- Consciencia presencial
- Consciencia influencial
- Consciencia de habilidades
- Consciencia contextual

Antunes et. al. en su estudio [10] identifica 6 tipos de conciencia, como se puede ver en la Figura entre los que se encuentran conciencia de colaboración, conciencia del contexto, conciencia social, conciencia de ubicación, conciencia de espacio de trabajo, y conciencia de la situación. Con estos tipos plantea una lista de verificación en la que se encuentran criterios de diseño para sistemas colaborativos.

La **Consciencia colaborativa** ha sido generalmente aceptado como la percepción de la disponibilidad del grupo que tiene cada uno de los integrantes. Disponibilidad del grupo quiere decir el conocimiento de si las personas están en el mismo lugar físico, o si están conectados o desconectados y los medios de comunicación que tienen disponibles para colaborar entre sí.

Conciencia contextual es el entendimiento de las actividades de los demás usuarios, que proporciona un contexto para las actividades propias [17], aumenta la experiencia que un individuo tiene en cualquier escenario de asistencia, el manejo de acoplamiento de actividades, la coordinación de acciones cooperativas, anticipación de actividades humanas o ambientales tanto como futuras intenciones, y la búsqueda de ayuda son aspectos básicos de una interacción que puede ser mejorada mediante la consciencia contextual [19]

La **conciencia social** apunta la importancia de entender las prácticas sociales, como los roles de otros y sus actividades, o cómo están otros miembros del grupo contribuyendo a una tarea[10].

La **conciencia de espacio de trabajo** se divide en 2 aspectos: uno se enfoca en el lugar y el otro se enfoca en el espacio, otra cosa importante a considerar es la interacción del grupo con los lugares de trabajo, finalmente la noción del espacio de trabajo trae consigo el nivel de interdependencia de una tarea realizada por el grupo, considerando soporte de actividades paralelas, actividades coordinadas y actividades mutuamente ajustadas.

La **conciencia situacional** está caracterizada por tres niveles cognitivos: en el primero una percepción global del ambiente construido por eventos, acciones, recursos y otros elementos, en el

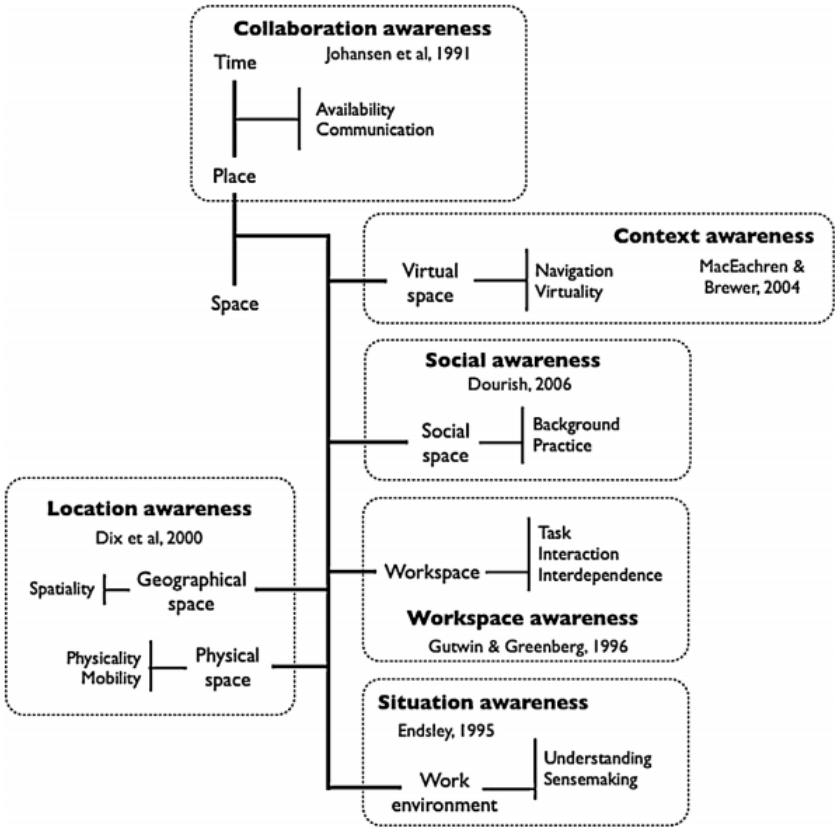

Figura 2.3: Taxonomía de conciencia [10]

segundo nivel se le da un sentido a lo que está pasando actualmente y en el tercero se construyen escenarios a futuro.

La **conciencia del lugar** puede referirse al conocimiento de los elementos de una ubicación geográfica como pueden ser coordenadas, orientación, distancia, etc. O también el conocimiento de los elementos de un espacio físico, incluyendo clima, topología física del lugar, y atributos físicos.

Ya que en un sistema colaborativo se necesita estar enterado de este tipo de información, es importante tenerla en cuenta al momento de diseñarlos así los usuarios pueden coordinarse de una mejor manera y llegar a alcanzar los objetivos que se tienen propuestos.

Por la naturaleza de los sistemas groupware las áreas de más interés son la conciencia social y la conciencia del espacio de trabajo. En otra clasificación (Gutwin, C., Greenberg, S., & Roseman, M., 1996) podemos encontrar 2 tipos de conciencia al contexto: primero conciencia general de las personas en una comunidad de trabajo, y segundo conciencia de las interacciones de otros en un espacio compartido.

Gutwin (Gutwin, Carl, Greenberg, Saul 1996) propone un marco de trabajo que considera elementos que incluyen mecanismos para recopilar información útil para la conciencia de la situación por parte de las personas, y que integran el conocimiento consciente del espacio de trabajo de un grupo.

Cuadro 2.2: Elementos de conciencia de la situación propuestos por Gutwin[16]

Elementos	**Cuestiones que responden**
Presencia	¿Quiénes están participando en la actividad?
Ubicación	¿Dónde están trabajando?
Nivel de actividad	¿Qué tan activos son en el espacio de trabajo?
Acciones	¿Cuáles son sus actividades y tareas actuales?
Intenciones	¿Qué harán después?, ¿Dónde van a estar?
Cambios	¿Qué cambios están realizando y en dónde?
Objetos	¿Qué objetos están usando?
Extensiones	¿Qué pueden ver?, ¿Cuáles son sus alcances?
Habilidades	¿Qué pueden hacer?
Esfera de influencia	¿Dónde pueden hacer cambios?
Expectativas	¿Quśe necesita que haga ahora?

Los elementos de la Tabla 2.2 se pueden clasificar en dos grupos: aquellos que se encargan de saber que está pasando con otras personas, y aquellos que dan a conocer dónde está pasando. Esta clasificación detalla el perfil individual de cada integrante y sigue las actividades de cada uno. Para facilitar la obtención de este conocimiento a los usuarios es necesario que los groupware cuenten con un mecanismo que reconozca el flujo de las actividades que se están realizando. A esto se le llama conciencia contextual, que a diferencia de la conciencia que es el conocimiento que tienen los usuarios, se encarga de dotar de este conocimiento al sistema computacional para que pueda adaptarse a las actividades del equipo de trabajo.

2.3. Consciencia del contexto

Los elementos mencionados hacen referencia a la conciencia de parte del usuario del contexto que lo rodea, pero ¿es posible hacer que la conciencia del ambiente pueda ser aprendida por el propio sistema groupware? Existen groupwares capaces de reconocer aspectos contextuales que rodean a un grupo de usuarios, estos sistemas son llamados conscientes del contexto, la información contextual más común en los sistemas groupware es la ubicación física del usuario. Al hablar de sistemas

groupware conscientes del contexto surgen algunos asuntos importantes a tomar en cuenta, haciendo falta una categorización formal del contexto con el que los sistemas van a modelar la situación de los usuarios, sin embargo los sistemas son desarrollados para que funcionen en ámbitos específicos al dominio de aplicación o situación a la que se le quiere dar solución.

El conocimiento contextual describe una situación, la forma en la que se usan los elementos en un grupo de trabajo, incluyendo los eventos que son manejados por el grupo [20]. Varios autores tienen un concepto de contexto, algunos de ellos traslapan en definición con otros, y diferentes elementos son tomados en cuenta para la descripción de contexto. Dey [4] toma varias definiciones de contexto que otros autores habían hecho antes y hace su definición con un enfoque computacional, se refiere a contexto como cualquier tipo de información que se puede usar para caracterizar la situación de entidades (se entiende por entidad una persona, lugar u objeto) que es considerada relevante para la interacción entre un usuario y una aplicación incluyendo al usuario y la aplicación.

Malik [21] hace referencia a varios problemas que hay en la actualidad inherentes a consciencia contextual, entre ellos están la definición del contexto, ya que contexto es un concepto que abarca todos los posibles parámetros que identifican una situación, las aplicaciones y marcos de trabajo están limitados a definir los parámetros del contexto de su propio ámbito. Otro problema importante es que las arquitecturas no están demasiado desarrolladas aun, están desarrolladas para tareas específicas, hace falta estándares para definir una arquitectura y herramientas, por último otro problema que es de interés para este trabajo es la interpretación del contexto y las adaptaciones del comportamiento del servicio.

Brézillon [20] estudia tres casos de sistemas groupware conscientes al contexto y describe la forma en que apoya a la conciencia del contexto con los usuarios. SisPro es un sistema que tiene por objetivo facilitar las actividades colaborativas y procesos de aprendizaje y el desarrollo de competencias de trabajo colaborativo. SISCO tiene como tarea la preparación de reuniones, da a conocer a los usuarios los temas de los que se está hablando basados en su agenda individual. CO2DE es un software que permite unir los contextos individuales en un solo diagrama proporcionando una infraestructura de edición colaborativa.

Un caso de sistema groupware muy diferente a los anteriores es *Assault Cube* [1], un juego FPS (Fisrt Shoter Person) de código abierto que soporta las actividades colaborativas, en este varios jugadores se conectan a un servidor para llevar a cabo actividades específicas de la modalidad del juego que hayan escojido, el juego ofrece mecanismos de conciencia a los jugadores tales como un mapa de la ubicación del enemigo, un medidor de vitalidad, una pantalla de mensajes para comunicar al equipo, etc., en este sistema es crucial para los jugadores estar al tanto de los movimientos de sus compañeros para formmular estrategias, y tener mayor probabilidad de cumplir con los objetivos del juego.

En Figura 2.4 se comparan los sistemas mencionados anteriormente, para realizar las comparaciones se listan algunos elementos propuestos por Gutwin [16], Montané [1] y Dey [4] estos elementos pertenecen a tres categorías que son ambiente de usuario, ambiente físico, y actividad; en la parte inferior de la Figura2.4 comparan cuatro características de los groupware, el tipo de interacción que

apunta el tiempo en el que interactúan los usuarios ya sea síncrona, asíncrona o mixta; la vista que ofrece el sistema, es decir, si las pantallas que muestra son compartidas por algunos usuarios o son publicas; la sesión se refiere a la distribución geográfica de los usuarios y el contexto que describe en que ambiente se desempeñan los sistemas comparados.

	Elementos	SisPro Project	SISCO	CO2DE	AssaultCube
Ambiente de usuario	Usuarios	☑	☑	☑	☑
	Roles				
	Influencia		☑		
	Relaciones				☑
	Cambios de estado	☑		☑	☑
	Alianzas				☑
	Grupos	☑	☑		☑
	Intenciones				
	Habilidades		☑		☑
	Movimiento				☑
	Expectativa				
	Sentimientos				
	Conciencia de otros	☑	☑		☑
	Identidad				☑
Ambiente Físico	Ubicación				☑
	Lugar	☑	☑		
	Tiempo	☑	☑		
	Clima				
	Cambios		☑	☑	
	Movimiento				☑
	Objetos de Interacción	☑		☑	☑
Actividad	Objetivos				☑
	Reglas				
	Estado de tareas	☑	☑		☑
	Eventos	☑	☑		☑
	Expectativas				
Interacción		síncrona	Mixta	Síncrona	síncrona
Vista		Privada, compartida	Pública, compartida	Pública, compartida	Compartida
Sesión		Distribuida	Distribuida	Distribuida	Mixta
Contexto		Actividades colaborativas varias, procesos de aprendizaje	Preparación de reuniones	Edición en tiempo real de diagramas	Videojuego de disparos en primera persona colaborativo

Figura 2.4: Comparación de sistemas groupware

Como se observa en la Figura 2.4 no todos los sistemas abarcan los mismos aspectos contextuales, cada uno tiene una arquitectura singular creada para satisfacer las necesidades del entorno para el que fueron creadas, esto implica que no se pueda reutilizar el modelo de uno de ellos en los demás.

2.4. Modelos Contextuales

Contexto es tradicionalmente la localización, identidad, y estado de las personas, grupos y objetos virtuales y físicos, según Pereira [22] el contexto puede ser visto como un conjunto de condiciones e influencias en una situación relevante y que la hacen única y comprensible, esta situación puede referirse a una persona, grupo de personas, objeto físico, entidad computacional, etc. El concepto de modelos mentales tiene una relación muy cercana a la consciencia contextual y situacional [19], al momento de modelar contexto, es necesario distinguir entre los diferentes tipos de información contextual[23], el contexto de las actividades colaborativas pueden ir desde un editor de documentos

distribuido, hasta un videojuego colaborativo; por lo que los elementos particulares de dichas actividades cambia muy radicalmente de uno a otro, y con esto surge la necesidad de usar una taxonomía con un alto nivel de abstracción que soporte la diversidad de contexto con los que se trabaja.

Belkadi et al [18] proponen un modelo situacional para mejorar la conciencia centrada en los conceptos de situación, interacción y rol, en el que hacen un recuento de los conceptos clave a tomar en cuenta para un modelo, entre ellos se encuentran:

- Elemento contextual
- Tarea y actividad: describen lo que se espera y lo que se tiene que hacer.
- Recursos: describe un elemento del contexto, en dónde es usado y cómo contribuye a la tarea.
- Interacción: interacciones entre un sujeto y un objeto mediante el uso de herramientas, e interacciones sociales mediante la definición de reglas.
- Role: definen las responsabilidades de los sujetos.

En su modelo para descripción de situaciones definen entidades básicas como recursos humanos y objetos, entidades de interacción que relaciona a las entidades básicas, las cuales se clasifican en cuatro tipos: operacional, que se refiere a las diferentes metas para ser cumplidas(tarea, actividad, proyecto); comunidad, que se refiere a la afiliación de unas entidades con otras; colaborativa que denota el intercambio de información durante una actividad colectiva, y restricciones que indican requerimientos, reglas y límites para la realización de una tarea. En cuanto a los roles se definen cinco tipos: actor(¿Quién hace qué ?), customer(¿Para quién ?), manager(¿Cómo ?), support(¿Con qué ?), object(¿Sobre qué ?).

Así mismo Pereira et. al. [22] maneja un modelo llamado SeCoM(*Semantic Context Model*) dividida en tres capas: ontología de alto nivel que representan el contexto o procesos de software, la ontología de integración es parte de una técnica para conectar los datos recojidos por los sensores con la arquitectura y la ontología de sensores que implementan ontologías para los datos que son recolectados. El modelo SeCoM consiste de un conjunto modular de ontologías basadas en dimensiones semánticas de identidad(Actor), ubicación(Espacio), temporal(tiempo), de actividad(Actividad) y métodos de captura y acceso(Dispositivo).

En otro estudio Alves [24] menciona una definición para los elementos de su modelo definida formalmente como sigue: Un atributo A_i es una tupla (N, V), donde N es un nombre representando el atributo(por ejemplo, velocidad) y V es el valor del atributo(e.g., 100); P es un conjunto finito de personas $P_1, P_2...P_n$; t es el rango de tiempo entre dos marcas de tiempo; y A un conjunto finito de atributos $A_1, A_2...A_n$. Con estos elementos representa situaciones definiendo un contexto C como una 3-tupla (P, t, A) que representa los atributos que caracteriza la situación de un grupo de personas P durante un intervalo de tiempo t. Por ejemplo suponiendo que Alice está en Nueva York entre Julio 1 y Julio 3. Podemos definir su contexto C como:

$C = ((Alice), 01/07., 03/07, (Ubicacion, NuevaYork))$

Este modelo esta orientado a la propagación después de que la información ha sido agregada tomando en cuenta los atributos en común de las personas que comparten un mismo contexto.

En el trabajo de Ardissono [25] mencionan activity frames para simplificar el contexto de actividades cooperativas, un frame era definido por una 5-tupla (fn, U, O, Oi, T), donde fn era el nombre del frame, U el conjunto de usuarios involucrados en la actividad, O los objetos asociados al frame, Oi es el conjunto de objetos del frame por medio de inferencias, y T las tareas asociadas a la actividad. Y a su vez las tareas están formadas por $(tn, U, O, Oi, g, P, T, s, d)$, donde tn es el nombre de la tarea, U, O y Oi como dicho antes son los Usuarios, Objetos, y objetos inferidos asociados a la tarea, g es el objetivo, P es el grupo de tareas que deben de estar cerradas antes de iniciar tn, T es el conjunto de tareas hijas, s es el estado(habilitada, deshabilitada, cerrada) y d es la fecha límite para terminar la tarea, nula por defecto.

Anallely Olivares [26] menciona que es importante tomar en cuenta el contexto de una actvidad colaborativa en lugar de hacerlo para cada uno de los usuarios, es por eso que en su modelo enfatiza el uso de variables típicas de un grupo de trabajo, por ejemplo estado de los proyectos, políticas de la organización, ubicación física de los colaboradores, recursos disponibles, etc.

Bo Hu [27] divide su modelo en dos partes una meta ontología y una ontología de dominio, los elementos del meta modelo se encuentran organizados en cinco categorías: cómputo, que define elementos como dispositivos, servicios, redes, y sistemas; persona, con elementos como usuarios, diseñadores y desarrolladores; ubicación, que se divide en la ubicación física y distancia; actividad, con dos tipos de actividades, planeadas y no planeadas; y misión, donde se identifican misiones, metas y requerimientos. La ontología específica de dominio define los conceptos y relaciones dentro del dominio dado, y es restringido por el meta modelo.

En el año 2013 se propone MARS que es un modelo contextual de regulación que ayuda a modelar la actividad soportada por herramientas groupware. En este modelo las interacciones toman lugar en un espacio llamado arena, cada interacción es representada por un escenario que describe la forma en que se lleva a cabo dicha interacción, quiénes participan y los objetos involucrados, además se definen condiciones y precondiciones para que este escenario se lleve a cabo; en esta arena están presentes actores que realizan acciones, durante esta actividad se manejan o producen objetos; actores y objetos pertenecen a diferentes familias y desempeñan diferentes papeles o roles. Dado que los miembros de un grupo colaborativo pueden serlo a su vez de otro, se definen vistas que determinan los actores, objetos e interacciones que una arena puede compartir con otra.

En otro caso Montané et. al. proponen un modelo de contexto social[1], donde hay categorías similares, tomando en cuenta las relaciones entre los sujetos además de las actividades y estado actual de cada uno de ellos; los elementos que lista en su trabajo están divididos en tres categorías. La interactiva describe a los usuarios de forma individual y sus interacciones con las cosas que los rodean como objetos, tareas, eventos, usuarios o ubicaciones; la cohesiva integra elementos que tienen que ver con la actividad grupal y se pueden encontrar los grupos, roles, metas, alianzas,

actividades y reglas. Y las afectivas que describen cómo se sienten los miembros del grupo al realizar las actividades entre ellas están los sentimientos y los gestos.

Skillen [28] habla de un mecanismo de personalización basado en reglas donde identifica conceptos clave para modelar a los usuarios en un ambiente pervasivo de los que lista los siguientes: Usuario, Perfil de usuario, ubicación, preferencias del usuario, Objeto de asistencia en una interacción, actividad que realiza el usuario, contenido multimedia que será enviado al usuario(audio, video, imagen o texto), Condición de salud que pueden afectar el tipo de medio que se le envía al usuario, escala de calidad de la información entregada, y formato de interfaz de usuario donde se desplegará la información enviada.

En una revisión de la literatura sobre modelos contextuales se revisaron ontologías contextuales, entre las que se podían encontrar los siguientes elementos:

Cuadro 2.3: Unidades contextuales encontradas en los frameworks revisados.

Elementos	**Descripción**
Objetos o artefactos	Entidades sobre las que los usuarios pueden realizar alguna acción.
Tareas	Acciones asociadas con usuarios, objetos, objetivos y subtareas que se deben de llevar a cabo para alcanzar un objetivo.
Eventos	Eventos ocurridos en interacciones[1]
Usuarios	Entidades que pertenecen a una comunidad y que realizan tareas[1].
Ubicaciones	Posición virtual o física en un grupo [1].
Grupos	Colección de usuarios que realizan una actividad[1].
Objetivos	Los objetivos de la comunidad[1].
Alianzas	Subconjunto de usuarios en un grupo [1].
Actividades	Actividades realizadas por un grupo[1].
Reglas	Comportamientos definidos por el grupo [1].
Foco de visión	Dónde están mirando los usuarios[29].
Vistas de sistema, espacios de trabajo	Qué pueden ver los usuarios [29].
Alcance	Alcance de los usuarios[29].
Presencia	Presencia de usuarios en el espacio de trabajo[29].
Intención	De qué objetivo es parte una tarea[29].
Habilidades	Capacidad para llevar a cabo un conjunto de actividades con cierto nivel de destreza [30].
Contexto físico	Incluye todas las magnitudes físicas (e.g. tiempo, espacio, temperatura, nivel de luz, nivel de ruido)[23].

Continúa en siguiente página

Cuadro 2.3 – *Continúa de página anterior*

Elementos	**Descripción**
Contexto computacional	Información relacionada con el software y hardware de sistemas, e.g. trafico de red, condiciones, estatus de hardware, información pedida por el usuario, requerimientos de memoria[23].
Ambiente	Descripción de la distribución física de los objetos y usuarios en un espacio de trabajo[23].

Todos estos elementos se extrajeron de diferentes modelos entre los que se hizo la siguiente tabla comparativa:

Autor	Tipo	Físico	Objetos	Tareas	Eventos	Usuarios	Ubicaciones	Grupos	Roles	Objetivos	Alianzas	Actividades	Reglas	Foco de visión	Vistas	Alcance	Presencia	Intención	Habilidades	Notificaciones	Contexto físico	Ambiente	Contexto computacional	Curso
Ellis, Clarence, and Jacques Wainer 1994	general		☑			☑		☑	☑		☑		☑											
Montané-Jimenez, L. G., Benitez-Guerrero, E., & Mezura-Godoy, C.(2013, Octubre)	Social	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑					☑						
Gallardo, J., Molina, A.I., & Bravo, C. (2012, Octubre)	Centrado en el usuario		☑	☑		☑	☑		☑					☑	☑	☑	☑	☑		☑				
Decouchant, D., Mendoza, S., Sánchez, G., & Rodríguez, J. 2013	Colaborativo			☑		☑		☑	☑	☑					☑			☑	☑					
Hoyos, J. R., et. al., 2013	General	☑	☑	☑		☑	☑		☑	☑	☑	☑					☑		☑		☑	☑	☑	
Guemah, H., et. Al. 2013	e-learning					☑																☑	☑	☑

Figura 2.5: Comparación de modelos de contexto[13][1][29]

2.5. Arquitecturas y Marcos de Trabajo Conscientes del Contexto

Gutheim [31] menciona que hay dos tipos de arquitecturas, las que siguen un modelo distribuido haciendo uso de servicios(*broker*) usado tradicionalmente para decoplar proveedores y consumidores de datos contextuales, este modelo es usualmente utilizado para modelar contexto y para mecanismos de inferencia; el otro tipo de arquitecturas son aquellas que siguen un modelo punto a punto en el que generalmente los proveedores y consumidores de información contextual se conocen mutuamente y se envían información directamente [32].

Para que una arquitectura sea consciente del contexto debe de cumplir con los siguientes requerimientos[33]:

- Permitir a las aplicaciones acceder a información contextual desde máquinas distribuidas en el de la misma forma en que acceden a la entrada del usuario en una máquina local.
- Soportar la ejecución de diferentes plataformas y el uso de diferentes lenguajes de programación.
- Soportar la interpretación de información contextual.
- Soportar la agregación de información contextual
- Soportar independencia y persistencia de widgets contextuales
- Soportar el almacenamiento del historial de contexto

También para el soporte de consciencia contextual, Elg Ghayam [34] menciona 2 requerimientos que se deben de cumplir: la formalización de contexto, para delimitar los datos contextuales y facilitar la distinción de parámetros de contexto, una categorización de datos contextuales es requerida, más aun, para reducir la complejidad de su manutención, una representación formal también es necesaria. El otro punto son reglas de adaptación: la adaptación de contexto debe de 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, no es evidente que no se pueden enumerar todas las posibles situaciones que van a ocurrir. Por lo anterior, un método para construir reglas de adaptación es requerido, que pueda manejar la diversidad de posibles situaciones que se construyen en base de esos parámetros contextuales.

Para poder mejorar la colaboración de los usuarios, la arquitectura debe de permitir que el groupware pueda adaptarse al contexto, y de acuerdo a Abowd[35] hay 3 tipos de adaptación:

- Presentación de información y servicios al usuario. Se refiere a la técnica de interacción que muestra una lista de objetos o lugares cuyos elementos más importantes son resaltados de acuerdo al contexto actual del usuario.
- Ejecución automática de un servicio. En este caso un servicio es automáticamente lanzado si la combinación correcta de condiciones es dada.
- Aumento de la información. Información contextual puede servir para entender mejor el ambiente colaborativo.

Chihani [36] propone un framework que desacopla el manejo de contexto de la lógica de negocio de las aplicaciones siguiendo un modelo *broker*, esta arquitectura consta de 3 capas, los proveedores de información contextual, el mecanismo para el manejo de contexto y a los consumidores. En la capa de proveedores se encuentra una función “provide"que genera la información contextual prima, después está la capa de manejo de contexto en la que se implementan cuatro funciones para el uso de contexto; la función de filtrado “filter"que procesa señales para eliminar el ruido en la información

contextual, la función “abstract"que transforma la información contextual prima a un nivel más alto de abstracción en la que se usa un autómata de estado finito compuesto por los estados de los datos contextuales; la función “select"que permite la selección de la “mejorïnformación contextual basada en criterios programados; y la función “aggregate"que realiza una agregación de información contextual basada en un conjunto de reglas que apuntan condiciones que la información tiene que cumplir antes de ser consumida. Con respecto de la capa de consumidores, se hace uso de una función “consume"la cual accede a la información contexual de cualquier nivel(prima, abstracta o compuesta). Una característica que diferencía esta arquitectura de otras es la falta de un modelo contextual con el cual describa la situación de los usuarios.

Skillen et. al. [28] proponen una arquitectura distribuida orientada a servicios para ambientes pervasivos, consta de cuatro componentes que interactúan unos con otros para lograr la personalización de servicios de asistencia a la orden, el objetivo de esta arquitectura es permitir un flujo de información entre una aplicacion y los servicios de ayuda, entre los componentes se encuentran el ambiente pervasivo que es la que genera y consume la información contextual, capa de servicios de perfil de usuarios donde se modela al usuario y sus preferencias, la capa de personalización de servicios que consiste de un mecanismo basado en reglas y de inferencia y los servicios de asistencia a la orden que ayudan a los usuarios a manejar sus actividades diarias.

La propuesta de Anallely olivares et. al. [26] es una arquitectura para soportar el uso de contexto en herramientas groupware basada en escenarios y situaciones que sirven de medios para validar su funcionalidad. Esta arquitectura consiste en cuatro fases: percepción de eventos contextuales divididos en dos tipos: físicos y lógicos, los físicos son recuperados por sensores y los lógicos reflejan cambios en las dimensiones internas del ambiente colaborativo, como el estado de los proyectos; detección de la situación donde se le da forma a la información recuperada en la fase anterior, para la definición de situaciones se tomaron en cuenta elementos como nombre, entidad afectada por la situación, oyentes que activan o desactivan situaciones, eventos contextuales que pueden provocar la activación o desactivación de situaciones y condiciones que son verificadas antes de activar o desactivar una situación; la siguiente fase es de comunicación de la situación que sigue un modelo suscriptor publicador donde un agente publica la información y el suscriptor toma la que es de su interés; la última fase es adaptación del sistema en el que usan reglas estáticas de la forma **if** situación detectada **then** adaptación planeada.

Pereira et.al. [22] muestra una arquitectura que sigue un diseño multiagentes, dirigida a grupos de trabajo de desarrollo de software, en la que integra herramientas web, una de manejo de proyectos como aplicación como ambiente de soporte para las actividades de los desarrolladores y otra para control de versiones, ambas herramientas son reguladas por un agente, en esta plataforma se definen dos tipos de agentes: agentes de servicio y asistentes personales; cada usuario cuenta con un asistente que cumple con varias funciones: entender sus necesidades, actuar de manera proactiva para anticiparlas, ejecutar comandos dados por los usuarios, presentar la información de forma inteligente, mediar la comunicación con otros miembros del equipo, y capturar y representar las operaciones de los miembros del equipo ayudándolos en el proceso de crear conocimiento. Por otro lado, los agentes de servicio se encargan de encapsular las aplicaciones con las que interactuan los

usuarios(la plataforma web de desarrollo y el controlador de versiones) checando las modificaciones en los contenidos de las aplicaciones o actualizando documentos.

El trabajo actual se basa en un trabajo previo de Montané [1], en el que, a partir de variables contextuales observadas en experimentos realizados, se propone un modelo conceptual de una arquitectura capaz de trabajar con información contextual. La arquitectura se divide en 3 capas: la de adquisición de datos, que es la que recibe los datos por separado dependiendo de la categoría a la que pertenezcan; la de manejo de contexto, que es la capa encargada de administrar el almacenamiento, recuperación y actualización de los datos contextuales que se guardarán en una base de datos; y la capa de uso de contexto, que tiene 2 tareas principales: razonar los datos contextuales recuperados, y a partir del resultado obtenido de este procesamiento, entregar información relevante a los usuarios de un groupware o enviar instrucciones de ejecución al sistema para poder adaptarse al contexto de los usuarios.

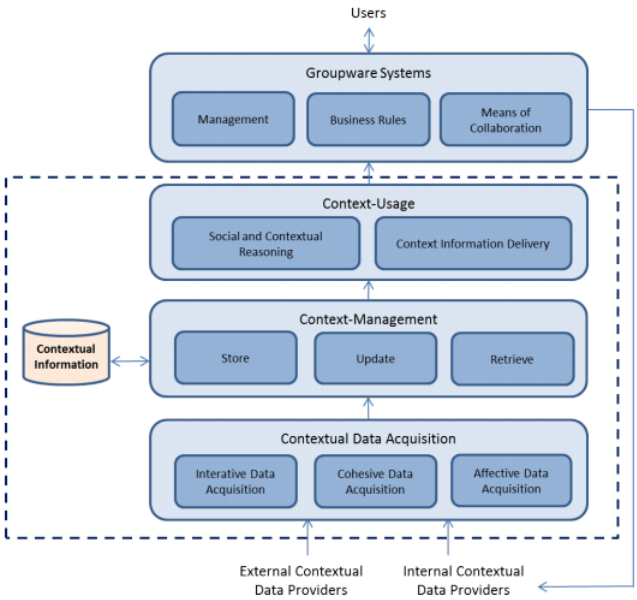

Figura 2.6: Arquitectura para soportar colaboración en groupware conscientes del contexto [1]

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[33] 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 [37] se reconfiguran servicios para adaptarlos a situaciones que cambian dinámicamente. Decouchant [30] divide su arquitectura en tres capas: la capa de espacio de trabajo, la de adaptación y la de detección de información contextual. Guerman [38] que propone una arquitectura orientada a sistemas de aprendizaje electrónico. En la figura 2.7 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	Manejos de contexto	Representación contextual	Razonamiento contextual	Distribución de información	Persistencia de datos	Reusó de contexto	Comunicación	Widgets contextuales	Manejo de Sesión	Esquema de Colaboración	Agregación de datos	Representación de estado de trabajo
Montané E-Jiménez, L.G *et. al.* (2013) [13]	x	x	x	x	x	x							
Dey, A.K. *et. al.* (1999) [6]	x		x	x		x			x				
Kamoun, E. *et. al.* (2012) [12]				x		x		x		x	x		
Decouchant, D. *et. al.* (2013) [4]	x	x		x	x	x			x			x	x
Guermah, H., *et. al.* (2013) [10]	x	x	x	x		x	x						

Figura 2.7: Comparación de arquitecturas que soportan consciencia del contexto[1][33][37][30][38]

Dentro las arquitecturas vistas, se identifica un proceso donde la información contextual es tratada desde su adquisición hasta la entrega de resultados a las aplicaciones consumidoras.

2.5.1. Proceso

Para obtener resultados de un conjunto de variables contextuales se establecen 3 pasos ya descritos antes [1]: recuperar las variables contextuales en un formato legible para la computadora, almacenar, recuperar y actualizar los datos contextuales de una base de datos, e inferir resultados de un conjunto particular de variables asociadas unas con otras para después distribuirlos.

Adquisición

El problema de mantener la consciencia del espacio de trabajo en los groupware gira en torno a la obtención de información útil más que el cómo la utilizan los usuarios [25]. Antunes propone un marco de trabajo para evaluar groupwares [14], un enfoque entrada-procesamiento-salida para conceptualizar las relaciones entre el soporte tecnológico y factores relacionados al comportamiento del grupo y contexto de trabajo. Las variables contextuales son factores importantes para describir el comportamiento del grupo, están clasificadas en 5 categorías generales [14]: personales, situacionales, estructura del grupo, características de la actividad o tarea y características tecnológicas. Los procesos grupales están definidas como las características de las interacciones del grupo, incluyendo las decisionales, comunicacionales, e interpersonales. Por último este framework evalúa los resultados de los procesos grupales afectados por el soporte tecnológico, incluyendo los relacionados con las actividades y con el grupo en sí.

Manejo de contexto

Alves [24] propone un mecanismo de agregación de información contextual para mejorar la eficiencia de consumo de recursos de red para aplicaciones conscientes del contexto, para esto se define contexto como un conjunto (P_n, t_n, A_n), conformado por un grupo de personas $(P_1...P_n)$ que en un lapso de tiempo t cumplen con un cierto atributo A compuesto por un nombre y un valor, con esto propone una función de agregación $Aggr(A_c, (P_1, t_1, A_1), ..., (P_n, t_n, A_n))$ en la que se mezclan un conjunto de contextos acontecidos en un lapso de tiempo con un atributo Ac en común tal que para todo atributo A_i perteneciente a $A_1...A_n$, Ac pertenece a A_i.

AYPUY es un manejador de recursos creado para ambientes de colaboración distribuidos. Almacena diferentes tipos de recursos, como contratos, historias clínicas, objetos de aprendizaje, etc., y garantiza su acceso cumpliendo con los objetos de confidencialidad, seguridad y escalabilidad de los sistemas que lo utilizan. En este framework un recurso está compuesto de un conjunto de atributos $A = L, D, F$, donde L describe las características lógicas generales del recurso (fecha de creación, autor, tipo de recurso), D establece las características relacionadas con el dominio (e.g., salud, educación) al que pertenece el recurso, y F describe las características físicas de almacenamiento del recurso (e.g., tipo de replicación, cadena de conexión), así AYPUY establece la estrategia de almacenamiento y acceso, adicionalmente los recursos tienen un conjunto de operadores O, que determinan las acciones que se pueden hacer con ellos, y son extensibles para soportar un dominio específico.

AYPUY administra los recursos controlando su acceso con espacios de trabajo (ET), crea un ET general al que pertenecen todos los usuarios del sistema siguiendo un rol específico, si se requiere la especialización o modificación de los roles de acceso a los recursos para todos o un subconjunto de usuarios, se crea otro ET. En un ambiente empresarial, por ejemplo, el ET general representa a la empresa, y un ET1 correspondería a un departamento y ET1.1 a un proyecto en específico.

Distribución

La información que se recopila se ocupa de quién está trabajando en un contexto compartido, qué están haciendo, dónde están trabajando, cuándo ocurren varios eventos, y como suceden esos eventos [25]. La presentación de información consciente es una parte importante de los groupware, Gross[39] menciona los siguientes puntos importantes que se deben de tomar en cuenta al momento de presentar información al usuario:

- La identificación y el señalamiento de retos sobre la desorganización de contenedores de información consciente son requerimientos centrales para el apoyar la consciencia.
- Los sistemas deben de proporcionar sugerencias que los usuarios puedan sobrescribir, ya sea en un momento específico o como regla general.
- La presentación de información de consciencia debe de ser explorada con la participación del

usuario, el tipo de visualización de consciencia debe de ajustarse a la necesidad de información del usuario y a su contexto.

- Modelos de consciencia son importantes para estructurar información de consciencia

Alves [24] propone un modelo genérico para propagación de contexto (Radiator) y necesidades de privacidad de aplicaciones distribuidas conscientes del contexto, enfocado a mejorar la escalabilidad de las aplicaciones y brindar privacidad de la propagación de la información, además agrega que la escalabilidad y la privacidad se pueden asegurar retrasando la propagación de contexto hasta que ciertas condiciones son alcanzadas y entonces agregar los mensajes en niveles sintácticos y semánticos, antes de propagar la información los datos tienen que tener un nivel de agregabilidad determinado de donde CP es el contexto actual de una persona P y C_x el contexto de alguien más que el sistema quiere propagar a P, $G(CP, C_x)$ representa que tan agregado debe de estar C_x antes de ser transmitido a P , tomando en consideración el contexto actual de CP. Por lo tanto un conjunto de contextos $C_1..C_n$ es solo propagado a P cuando para todo i perteneciente a $1..n$, $G(CP, C_i) = n$, n siendo un entero. Informalmente la agregabilidad representa el número de mensajes de contexto que deben de ser retenidos antes de su propagación, si se define una función G que siempre regrese 4, el sistema siempre va a agregar cuatro mensajes contextuales antes de propagarlos

En el trabajo de Ardissono [25] se mencionan 2 políticas dependientes del contexto para el manejo de notificaciones que apoya la selección de notificaciones para ser entregadas en base a las actividades actuales del usuario en diferentes niveles de granularidad: colaboración general de la tarea actual del usuario contra tarea llevada a cabo. Estas políticas son ofrecidas por el framework CONRAD (COntext depeNdent awaReness informAtion Delivery, por sus siglas en inglés). Las 2 políticas son las siguientes:

1. el filtro de contexto informa al usuario sobre eventos referentes a los contextos de colaboración en los que está trabajando e ignora los demás
2. el filtro de tareas es más selectivo y filtra las notificaciones basado en la tarea actual del usuario

Con este framework se reduce el nivel de distracción y la carga de trabajo presentada al usuario al momento de obtener información relacionada con su actividad. El objetivo de su trabajo es proveer a usuarios apoyo automatizado para especificar el tipo de información consciente más apropiado basado en la actividad del usuario y ajustar la entrega de las notificaciones y la aplicación de filtros para preferencias individuales de notificación.

Para adaptar servicios cooperativos al contexto del usuario, AYLLU [40] usa el framework AES que adapta la información en diversos contextos. AES funciona de la siguiente manera: una aplicación envía una consulta inicial a AES para que esta la enriquezca, AES obtiene los perfiles o características de la aplicación externa (usuario, contexto, dispositivo, perfiles), e invoca funciones de filtro de acuerdo a los perfiles. El resultado es un conjunto de datos de alta abstracción que es usado para generar una consulta enriquecida que será devuelta a la aplicación externa.

Cuadro 2.4: Lenguaje CoRaL

No terminal	**Expresión**
⟨*sentencia*⟩ ::=	⟨*expresión*⟩⟨*operador*⟩⟨*expresión*⟩";"
⟨*expresión*⟩ ::=	⟨*palabraReservada*⟩ : {⟨*elementos*⟩}
⟨*expresión*⟩ ::=	!⟨*palabraReservada*⟩ : {⟨*elementos*⟩}
⟨*operador*⟩ ::=	:: | →
⟨*palabraReservada*⟩ ::=	"Arena"| "Interacción"| "Actor"| "Familia de actor"| "Familia de Objeto"| "Objeto"| "Rol"
⟨*elemento*⟩ ::=	*cualquiercadenaenE*

El framework AYLLU [40] usa un protocolo de comunicación donde los mensajes son enriquecidos semánticamente, como en un sistema multiagentes. Se basa en la creación de una serie de agentes que ejecutan protocolos de comunicación, que ayudan al usuario a seguir una serie de protocolos de interacción que determinan un servicio cooperativo, cuando un servicio cooperativo se instancia se crea un Agente Manejador de Comunidad (CMA) por medio de un agente de fábrica (FA), el usuario cuenta con un agente asistente que muestra las peticiones al usuario para generar respuestas.

Mecanismos de razonamiento

Para adaptar los sistemas es necesario reconocer la situación de los usuarios y saber cómo brindarles apoyo, en consecuencia se requiere de una forma de interpretar las interacciones que se están llevando a cabo en la aplicación con lo que se han propuesto diferentes mecanismos de razonamiento para estos casos.

En el modelo contextual MARS se usa un lenguaje de regulación para describir escenarios llamado CoRaL[1] que toma en cuenta tres aspectos de los escenarios definidos en MARS especificados en las pre y pos condiciones de la interacción: quién puede participar en la interacción, qué objetos pueden ser manipulados, qué rol tiene un actor u objeto durante la interacción y en casos similares, referencias a otros escenarios. Con esto se define el lenguaje CoRaL con la siguiente sintaxis(Cuadro 2.5.1) descrita en noteación BNF para gramáticas libres de contexto. Siendo E el conjunto de elementos de cadenas de nombres de arenas, actores, familias de actores, familias de objetos, objetos y roles.

Skillen [28] en su capa de uso de contexto define cuatro funciones para el procesamiento de contexto, estas funciones se definen con el lenguaje de reglas de web semantico SWRL con el que describe condiciones que la información contextual debe de cumplir para personalizar un conjunto de servicios. Haciendo uso de la sintaxis de este lenguaje definen reglas como la siguiente.

UserProfile(?up), hasHealthCondition(?up, Blind) → HelpDelivery(PlayAudio), hasMediaType(PlayAudio, Audio), hasMediaVolumeLevel(PlayAudio, VolLevel_5).

[1] Collaborative Regulation Language

En la definición se puede observar un caso en el que se especifica la preferencia de audio con un volúmen específico para la entrega de información para una persona con discapacidad visual.

Así mismo Bo Hu [27] implementa el lenguaje SWRL para definir reglas en la ontología contextual, estas reglas son una clase de restricciones, relaciones y atributos que se deben de cumplir y siguen una estructura de lógica de predicados. Como ejemplo muestran una regla de ejecución entre los conceptos de persona y actvividad definiéndola de la siguiente manera $\forall x.Activity(x) \rightarrow (\sharp\{y|Person(y) \wedge Executing(x,y)\} \geq 1$.

En algunos casos se puede notar en los mecanismos de uso de contexto que se hace uso de lenguajes formales para la descripción de situaciones de los usuarios, algunos de ellos se centran en un usuario y lo que pasa alrededor de él para inferir resultados, en otros se hace más importante el concepto de trabajo colaborativo y los lenguajes que proponen se centran en describir la actividad de todos los usuarios y mezclar aquellos que tienen contextos similares.

Capítulo 3

Arquitectura Consciente del Contexto para Sistemas Colaborativos

Las arquitecturas o marcos de trabajo que soporten consciencia contextual buscan cumplir ciertos requerimientos por ejemplo abstraer la información contextual, ser distribuido e independientes de la aplicación que las use [33], para cubrir estas necesidades se necesita que el marco de trabajo sea accesible por el groupware desde cualquier ubicación. Así la publicación de servicios web consumibles desde el sistema colaborativo se vuelve una solucióna este problema, con esto se cubre la distribución de la arquitectura, además con estos servicios aumenta la compatibilidad con otro tipo de plataformas al enviar sus mensajes serializados o envueltos en una solicitud, haciendo posible la recepción de información contextual, compuesta de el estado actual de los elementos de la aplicación, y la emisión de los resultados hacia el groupware. Con la información recibida se mantiene un registro de la actividad del groupware que puede ser útil para análisis futuros como minería de datos o reconocimiento de patrones.

Se busca que la arquitectura sea independiente del escenario o dominio de una aplicación en particular. Por ello, la definición del modelo se divide en dos partes, al igual que [27] este modelo es presentado con un modelo conceptual y un modelo de dominio de aplicación; el modelo conceptual está compuesto por entidades que describen una actividad colaborativa mediante relaciones de elementos tales como *actores*, *objetos*, *tareas*, *categorías*, *roles*, *comunidades* y *objetivos*. A partir de este modelo, se pueden identificar elementos con los que se elaborará e instanciará el modelo de dominio de aplicación, es en esta instanciación donde se guardarán los datos del groupware.

Los datos recibidos deben de ser gestionados en bases de datos y procesados por un motor de razonamiento explicado más adelante, el cuál tiene que recibir información contextual y tiene que dar como resultado información para el usuario o un comando de ejecución para el groupware. Una vez que se obtienen los resultados se distribuyen a los diferentes dispositivos cliente. Para esto se debe de considerar que al ser las actividades colaborativas los usuarios pueden compartir el contexto

en algunas ocasiones, de lo que se desprende la necesidad de agregar la información contextual de usuarios que compartan ciertos atributos establecidos para una situación en particular, es decir, mezclar la información de tales usuarios para un manejo de la información más eficiente.

3.1. Modelo Conceptual

El presente trabajo hace uso de un modelo conceptual para la representación de actividades colaborativas dentro de los sistemas groupware, basado un modelo propuesto anteriormente [1]. Entre los interactivos se encuentran *actores*, que son los usuarios del sistema, los *objetos* son aquellas entidades que asisten en la realización una *tarea* o que son producto de ellas, las *categorías* clasifican a los actores, objetos y tareas, por último están los roles, que definen el papel que juega un actor en la actividad y por consecuente sus responsabilidades, los roles pueden ser del objeto o del actor, estos son asignados a una tarea para establecer el rol que va a tener el actor u objeto involucrado en la misma. Los datos cohesivos son de naturaleza abstracta, ya que no se les puede percibir de una forma tangible, entre ellos se encuentran las *equipos* que son asociaciones de actores que colaboran durante una *actividad*, estas actividades representan una unidad de trabajo colaborativo con *metas* que son ideales, en ocasiones inconmensurables, a los que los grupos de trabajo aspiran, por ejemplo ser los mejores realizando una actividad en particular, o ser un equipo de trabajo eficiente, para alcanzar estas *metas* se fijan *objetivos*, que al contrario de las metas, definen medidas que se tienen que satisfacer para decir que estos objetivos han sido cumplidos. En la Figura 3.1 se observa un modelo conceptual en el que se pueden ver los elementos antes descritos, se refleja la pertenencia de actores a comunidades, comunidades a actividades, las metas que son parte de las actividades y a su vez, los objetivos pertenecientes a cada una de las metas establecidas, y las tareas a realizar para cumplir dichos objetivos; para las tareas y los objetivos se toman en cuenta algunas propiedades útiles en el cálculo de presencia social como son la asignación directa a algún actor; la afectación de el cumplimiento de la tarea o el objetivo para los usuarios; el resultado generado, es decir a qué magnitud resultó negativo o positivo; y la efectividad en el caso de las tareas en las que es necesario saber si se llevó a cabo correctamente[**?**].

Con este modelo se puede describir el dominio de aplicaciones groupware definiendo cada uno de estos elementos a partir de interacciones, para hacer esto es necesario analizar el groupware y listar los elementos identificados así como sus atributos, estos se darán de alta mediante una plataforma web para la definición de dominio que instanciará los datos a un modelo que representará al sistema colaborativo y almacenará las variables contextuales que este transmita a la arquitectura. Cabe mencionar que en este modelo conceptual 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 y diseñar casos de estudio.

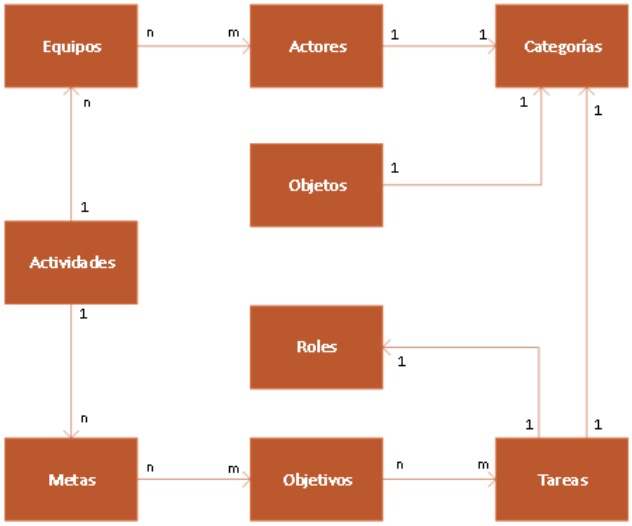

Figura 3.1: Modelo Conceptual [1]

3.2. Arquitectura

Para esta arquitectura se tomó como modelo base el propuesto por [1], la propuesta mostrada en la Figura 3.2, cuenta con tres capas: recuperación de datos, gestión de datos, y uso de contexto. Actualmente, en la arquitectura en la que se basa esta propuesta, se ha desarrollado ya la capa de recuperación de datos y la de gestión, la actual propuesta se concentra principalmente en la tercera capa, la de uso de contexto, teniendo como objetivo implementar un mecanismo de inferencia habilitado para trabajar con el modelo contextual explicado en la sección anterior. Para la primera capa se hace uso de servicios web encargados de recibir la información de parte del groupware que después pasaría a la fase de gestión para ser almacenada; en la segunda capa además del gestor de datos y las bases de datos está la plataforma web para la especificación de elementos de dominio y guiones de comportamiento. En la tercera capa se encuentran un motor de inferencia contextual y un inérprete de reglas de interacción y servicios que regresan los resultados del motor cuando es necesario.

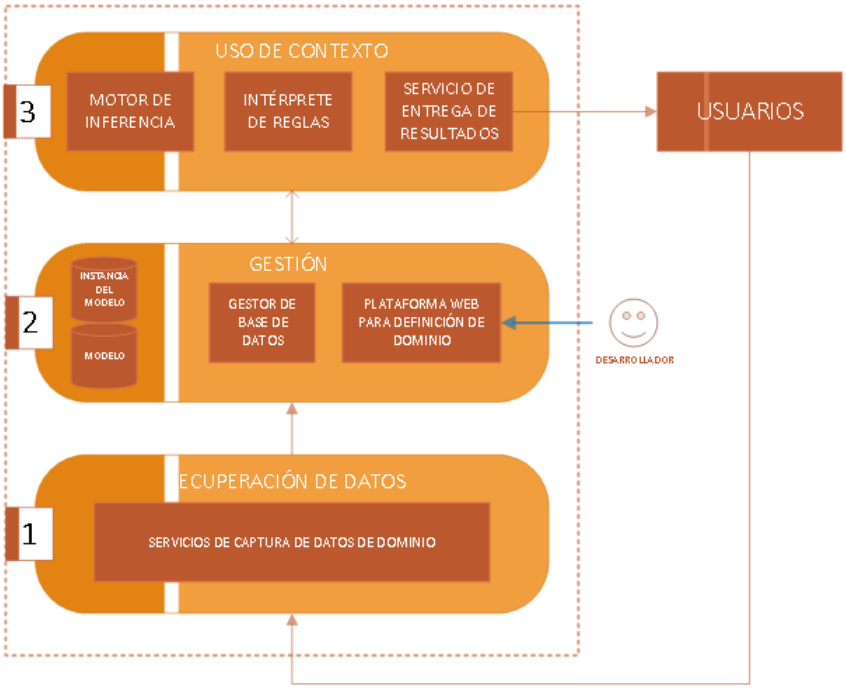

Figura 3.2: Diseõ conceptual de la arquitectura

1. **Recuperacón 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 servicios son generados a partir del modelo especificado en la plataforma web de la segunda capa, así mismo se encuentra un servicio que captura las ejecuciones de tareas durante una actividad en el groupware, los datos son enviados en un formato específico para que capas superiores puedan procesarla.

2. **Gestión**. Aunque es trivial en este caso, es necesario tener métodos de actualización y recuperaión de datos contextuales de las bases de datos, además a este nivel está definido el modelo conceptual y la plataforma web mostrada en la Figura 3.3.

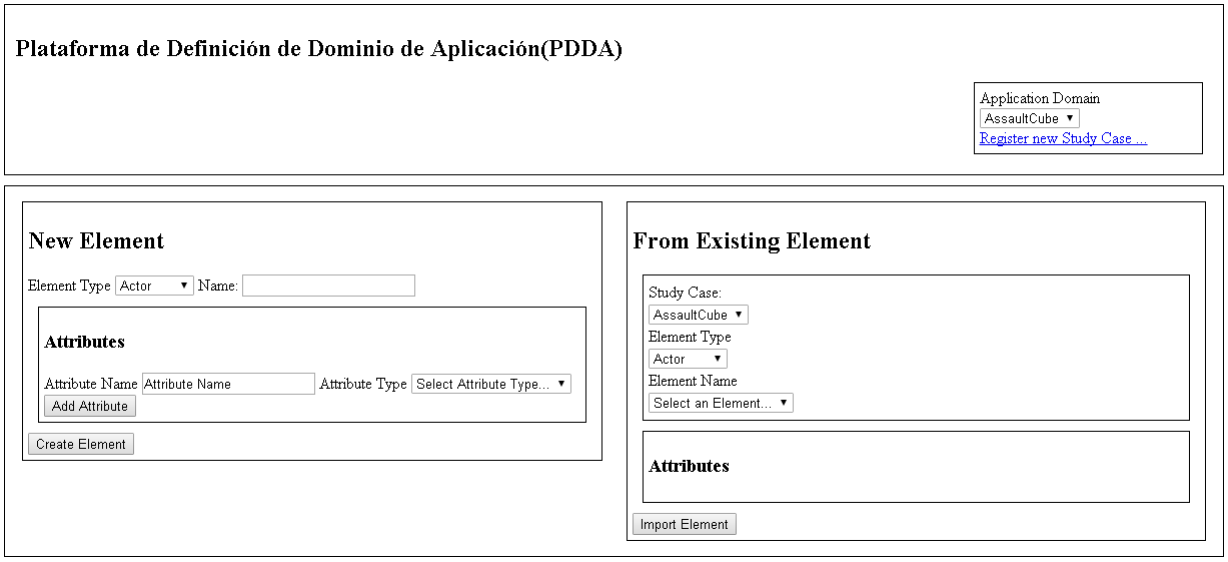

Figura 3.3: Plataforma web de Definición de Dominio de Aplicación

En esta platafaorma se encuentran formularios en los que se define cada elemento con un nombre y con una lista de atributos, de esta forma se define el modelo de dominio de la aplicación, detallando los datos de cada elemento del dominio, después, en la misma plataforma web, se definen el conjunto de reglas para adaptar el groupware a la actividad de los usuarios, estas reglas serán explicadas más adelante. Con la definición del modelo de dominio se dan de alta los servicios mencionados en el punto anterior. Este proceso de definición de modelo de dominio se describe en la Figura 3.4

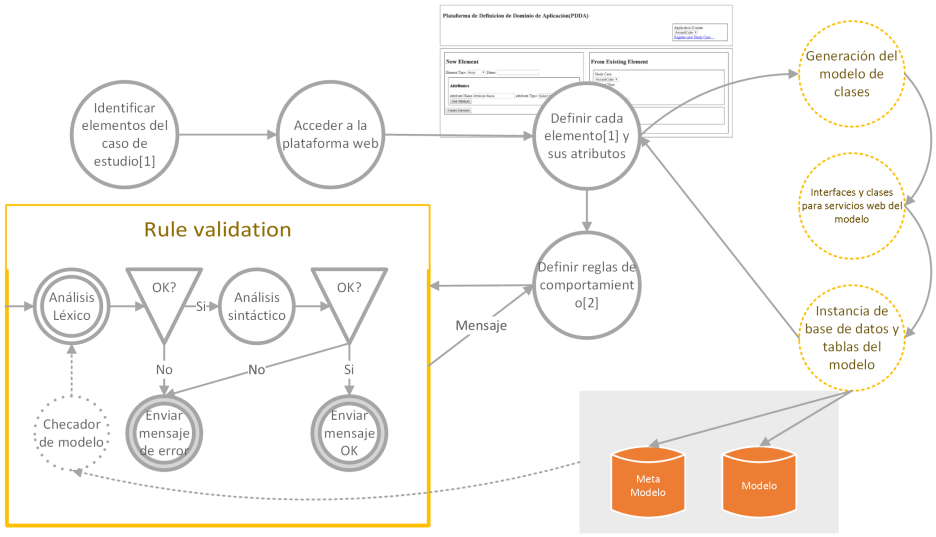

Figura 3.4: Proceso para definir modelo de dominio y reglas de comportamiento.

3. **Uso de contexto**. En esta última capa se encuentra un intérprete de reglas y un motor de inferencia, el intérprete se encarga de verificar la sintaxis, realizar un análisis semántico e identificar los elementos de cada regla, y verificar que estos elementos existan en el modelo de dominio, todo esto con ayuda de métodos de análisis como un autómata y expresiones regulares. El intérprete es usado por la plataforma web para validar las reglas que escriba el desarrollador en los guiones, una vez verificadas, se guardan junto con los guiones para que pueda usarlos el motor de inferencia. El motor de inferencia también hace uso de el intérprete para validar estructuras similares a las reglas, estas son las ejecuciones de las tareas que se llevan a cabo durante una actividad. El motor llevará la cuenta de las frecuencias de estas ejecuciones y las comparará con las establecidas en los guiones, una ves que un guión se cumple, se recuperan los resultados y se reenvían a través de servicios de distribución de información.

Después de definir el modelo en la plataforma, se especifícan los guiones, los cuales son un conjunto de reglas que describen las interacciones que se llevan a cabo dentro de el groupware y las características de un actor de a cuerdo a patrones de comportamiento particulares, con estos guiones se vigila la actividad de los usuarios y dependiendo de las tareas y el estado de los usuarios se tomarán medidas, también incluidas en las reglas, para hacer que el groupware responda a ese comportamiento, un ejemplo sería un guion para un sistema guía de turistas en el que se podría definir el comportamiento para una persona extraviada por la falta de conocimiento del lugar, definiendo reglas que verifiquen el número de consultas sobre ubicaciónes de lugares de interés, la ubicación actual del turista en relación con estos lugares, la frecuencia con la que el turista llega con retardo a los eventos, etc.

3.3. Motor de inferencia contextual

Este trabajo en particular se centra en el diseõ e implementación de la capa de uso de contexto mencionada en la sección anterior, para eso se proponen tres módulos principales: un intérprete para el lenguaje de reglas que se propone para describir el contexto de los usuarios durante una actividad colaborativa, un motor de reglas que apoyándose de las reglas definidas infiere las medidas a tomar para las situaciones descritas por las reglas y un servicio de entrega de resultados que se dispara cuando el motor de inferencia obtiene algún resultado.

El intérprete valida la definición de las reglas con un lenguaje que implementa un autómata de estado finito y expresiones regulares creadas para este propósito, con ellas se describe la pertenencia o relación de alguno de los elementos o instancias al otro en el modelo, comparan valores con los de los atributos de los elementos para condicionar su estado actual, o permiten representar la ejecución de tareas realizadas por actores en las que se puede incluir un elemento de asistencia, además se puede revisar el estado de estas ejecuciones comparando algún atributo de la tarea con un valor dado, por ejemplo el saber si una tarea fue exitosa o fallida. Estos tres tipos de reglas se definen como sigue.

Regla	**Tipo**	**Frecuencia**
$Elem_1(DomEl_a\{*\}) \rightarrow Elem_2(DomEl_b)$	pertenencia	no aplica
& $Elem_1(DomEl_a\{instancia_1\}).Nombre = Ana$	comparación	no aplica
& $Elem_1(DomElem_a\{instancia_1\}) - Task(escribir, Objetos(lapiz)).tiempo = 2segundos$	ejecución	5
:=		
$ME(DE\{i_1, i_2, i_3, ..., i_n\}) - Task(mostrarMensaje)$	tarea resultante	no aplica

- Pertenencia(P): $Elemento \rightarrow Elemento$ para validar que un elemento pertenezca a otro, o $Elemento \rightsquigarrow Elemento$ para validar que no pertenezca.
- Comparación(C): $Elemento.Atributo[<=> \neg]Valor$ para validar que el atributo de un elemento cumpla con la condición impuesta.
- Ejecución(X): indica que una tarea ha sido llevada a cabo, y si es el caso, con qué objeto se realizó, además se puede puntualizar otras características de la tarea como efectividad, o la afectación que tuvo, y si es necesario se puede añadir una regla de pertenencia en la misma declaración del elemento. $[Elemento|P] - Tarea(nombre, Elemento).nombreAtributo[= |\neg| < | >]valor$; un ejemplo práctico de esta regla puede ser la ejecución de un disparo efectivo por parte de un jugador perteneciente al equipo rojo con un francotirador.

En donde *Elemento* es un elemento del modelo descrito por un elemento conceptual, un elemento del dominio, y la instancia o instancias quedando la sintaxis de la siguiente forma: $Concepto(DominioListaDeInstan$ evidentemente los elementos y las instancias tienen que formar parte del modelo.

Para escribir un guión se usan varias reglas concatenadas que sirvan para describir el comportamiento de un usuario y una consecuencia que será el resultado que contiene ejecuciones de tareas que el groupware recibirá, interpretará y adaptará. Un guión tiene la siguiente forma.

En este guión(Figura 3.3) se muestra un ejemplo de cada uno de los tipos de reglas descritos antes, cabe mencionar que el único tipo de regla en el que se especifica una frecuencia es en la ejecución de tareas, ya que no tiene sentido hacerlo para las otras dos porque son comparaciones de estados de o propiedades de elementos.

El motor de inferencia recupera los guiones y los usa como una lsta de verificación para proporcionar información de interés para los usuarios o comandos de ejecución para la adaptación del sistema a la situación actual. Dentro del motor se hará una instancia de los guiones recuperados del modelo de dominio, la arquitectura estará recibiendo ejecuciones de tareas durante la actividad del groupware de manera constante cada cierto tiempo, estas tareas serán comparadas con cada uno de los guiones y se llevará un conteo de la frecuencia de cada una de ellas, cuando las ejecuciones lleguen al número indicado, el guion se cumple y se manda otra tarea, también incluida en el guión, como resultado para que será interpretada y ejecutada en el groupware. Una vez que se cumple un guion, este se reinicia para continuar recibiendo las tareas desde el sistema. Con esto se pueden

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 servicio de distribución de datos se encarga de enviar la información al groupware con los resultados. Este proceso es iterativo, ya que vive por el tiempo en el que el groupware opera. El proceso del motor se puede ver reflejado en la Figura 3.5

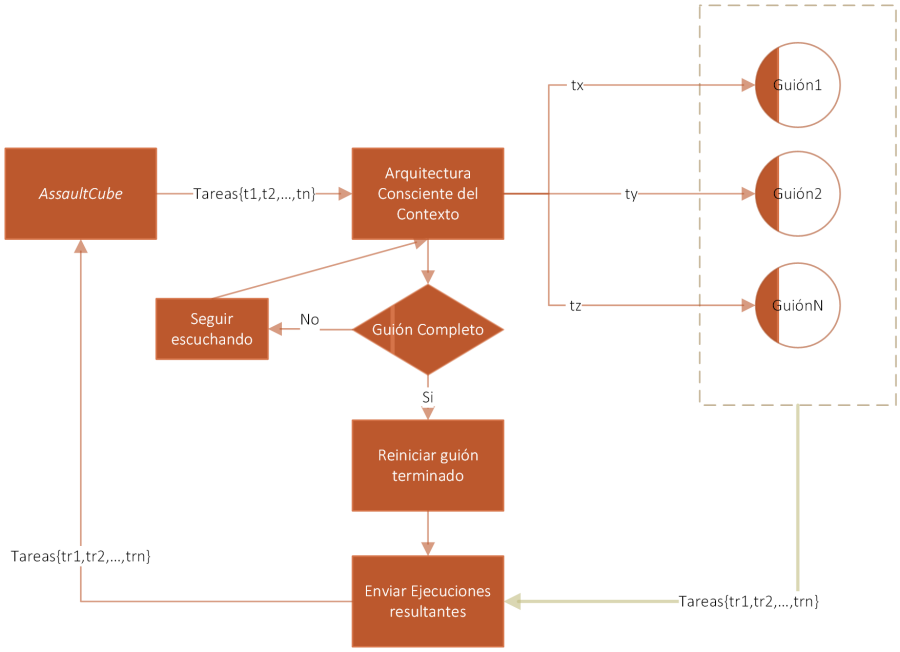

Figura 3.5: Proceso de inferencia

Capítulo 4

Implementación con caso de estudio

4.1. AssaultCube

Para el actual proyecto se necesita un groupware al cuál se le pueda acoplar la arquitectura, en este caso el sistema elegido es un video juego colaborativo de disparos en primera persona, *AssaultCube*. Este groupware en particular tiene las características de ser distribuido y síncrono según la clasificación de Ellis [2], 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, el juego cuenta con un servidor y varios clientes que se conectán a él para jugar, la arquitectura tendrá que estar acoplada al servidor ya que este es el que recibe toda la información de las actividades de los clientes.

El servidor del video juego *AssaultCube*(Figura 4.1) está desarrollado para la plataforma Linux, el cliente para Windows, y la arquitectura a desarrollar será programada en C#. *Assault Cube* tiene diferentes modos de juego, entre ellos capturar la bandera, cuyo objetivo es llegar a la base enemiga y recuperar la bandera que está ahí para regresarla a la propia bas; otro modo de juego es rey de la colina, que tiene por objetivo mantenerse más tiempo en una posición marcada por el juego que el equipo contrario; hay más formas de juego además de estas dos, y en cada una de ellas las actividades son diferentes y poseen objetivos diferentes, sin embargo pueden compartir interacciones como eliminar a un enemigo por ejemplo.

En el Cuadro 4.1 se muestran las interacciones identificadas en el juego para la actividad relacionada al modo de juego de capturar la bandera, estas interacciones están relacionadas directamente con las actividades que se pueden llevar a cabo en el groupware, las interacciones se vuelven irrelevantes cuando no tienen conexión con ninguna actividad, por ejemplo las interacciones del usuario con elementos de configuración en este caso particular, como en la interacción *"Jugador elige mo-*

Figura 4.1: Assault Cube

do de juego". En estas interacciones se encuentran algunos elementos del modelo como pueden ser Actores, Tareas y Objetos, esto nos da la pauta para empezar a diseñar nuestro modelo.

Cuadro 4.1: Interacciones detectadas en *Assault Cube*

Interacción	**Elementos identificados**
Jugador se Mueve	Actor: Jugador; Tarea: Moverse
Jugador salta	Actor: Jugador; Tarea: Saltar
Jugador dispara arma	Actor: Jugador; Tarea: Saltar; Objeto: Arma
Jugador recarga arma	Actor: Jugador; Tarea: Recargar; Objeto: Arma
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 envia mensaje de texto	Actor: Jugador; Tarea: enviar; Objeto: Mensaje de texto
Jugador envia 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

Continúa en siguiente página...

Cuadro 4.1 – ... *Continúa de página anterior*

Interacción	**Elementos identificados**
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 se identifican algunos elementos del modelo del groupware, por ejemplo, jugador como actor; arma, munición y mapa como posibles objetos, y el conjunto de ellos asociados a una tarea; también de ser necesario se pueden tomar en cuenta tareas como *bloquear arma* o *mostrar posición en el mapa* que podrán ser utilizadas como resultado para los guiones de comportamiento explicados más adelante. Además de los elementos identificados por medio de la observación, se exploró el sitio web oficial de la aplicación [41] para extraer más información del juego. Se encontraron dos **Equipos**: team clubers liberation army(TCLA) y rabid viper special forces(RVSF) los que compiten uno en contra del otro en cada partida. Dos tipos de **objetos**, armas y artículos, para las armas se describen atributos como tiempo de recarga, capacidad de munición, rafaga de disparo, etc. Para los artículos sólo se define el atributo de tiempo de reaparición. También se identificaron **objetivos** según los tipos de partida, de los que se infieren algunas **metas**. En el Cuadro 4.1 se muestran los elementos de dominio que se identificaron mediante la revisión de la documentación del sistema.

Cuadro 4.2: Elementos de dominio de *Assault Cube*

Elemento de Meta modelo	**Elemento de dominio de la aplicación**
Actores	Jugador(p.ej. Juan, Ana, Marcos, etc.)
Equipos	Team Clubers Liberation Army (TCLA); Rabid Viper Special Forces (RVSF).
Objetos	Armas , Artículos.
Roles	Explorer, Protector, Fragger, Recoverer.
Objetivos	capturar la bandera, robar la bandera enemiga, eliminar jugadores enemigos, regresar la bandera a la base aliada.
Metas	Tener la mayor cantidad de puntos, eliminar mayor número de enemigos, sobrevivir la mayor cantidad de tiempo.

Continúa en siguiente página...

Cuadro 4.2 – ... *Continúa de página anterior*

Elemento de Meta modelo	**Elemento de dominio de la aplicación**
Actividades	Capturar la bandera en equipo, Juego a muerte.

Con estos elementos se instanció el modelo de dominio en base de datos, el modelo resultante es representado en la Figura 4.2.

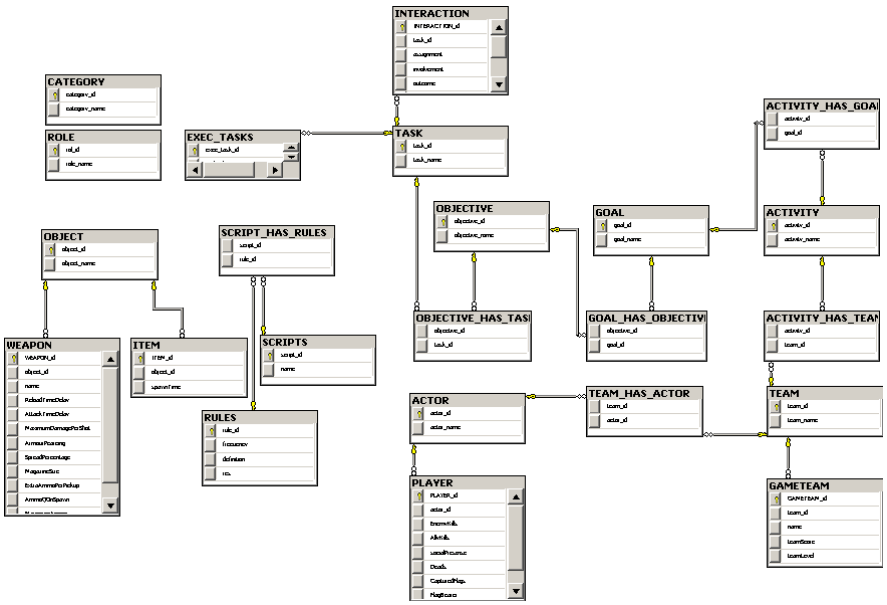

Figura 4.2: Modelo de dominio de *Assault Cube*

Después de instanciar el modelo de aplicación se agregaron datos para fines de pruebas, se tomó como punto inicial la actividad '*TeamKeepTheFlag*' a la cual se le agragó la meta '*scoreTheMost*' y equipos, a su vez a a las metas se le asocian objetivos y a estos tareas; de manera similar a los equipos le son asociados actores, la jerarquía de estas instancias se puede ver en la Figura4.3

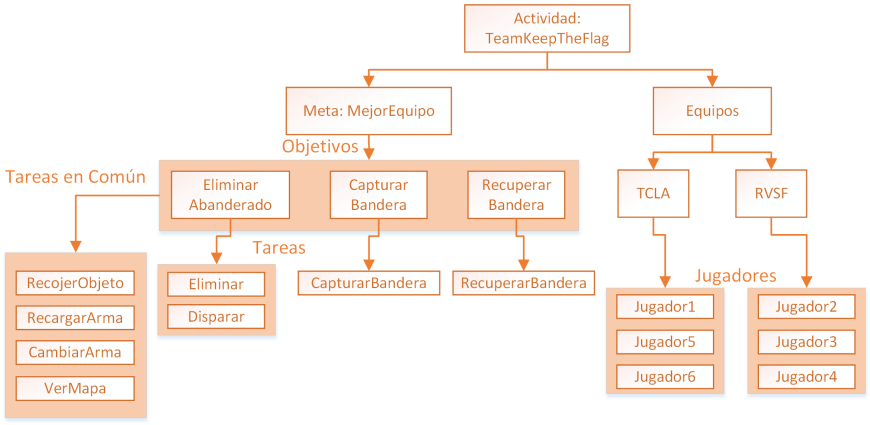

Figura 4.3: Instancias para caso de estudio

A partir de estos elementos pueden empezar a definirse algunas reglas, por ejemplo establecer una regla que diga que si un jugador capturó una bandera y está en peligro inminente de ser atacado, el sistema muestre a sus compañeros de equipo la ubicación del abanderado y envíe una señal de auxilio para que ellos puedan acudir a su ayuda.

Con estas interacciones y el conjunto de reglas definido se propone un conjunto de guiones que

describen diferentes tipos de comportamiento dado el caso de estudio; entre los que se encuentra un guion para jugadores novatos, un guion para jugadores inútiles, uno para jugadores con comportamiento sospechoso que podrían considerarse como traidores y uno para jugadores expertos. Estos guiones servirán al momento de ejecutar, los guiones se definen como sigue.

Cuadro 4.3: Guión para jugador inactivo

Regla	**Frecuencia**
$Actor(Player\{*\}) - Task(Shot)$	3
& $Actor(Player\{*\}) - Task(Move)$	10
& $Actor(Player\{*\}) - Task(PickItem)$	5
& $Actor(Player\{*\}).enemyKills = 0$	no aplica
& $Actor(Player\{*\}).allyKills = 0$	no aplica
& $Actor(Player\{*\}) - Task(captureFlag)$	1
& $Actor(Player\{*\}).socialPresense = bad$	no aplica
& $Actor(Player\{*\}) \rightarrow Team(red)$	no aplica
:=	
$[Actor(Player\{*\}) \rightarrow Team(red)] - Task(sendWarningMessage, Object(UI\{messageConsole\})$	no aplica

El guion de jugador inactivo del Cuadro 4.3 describe el comportamiento de un jugador con poca participación en las partidas, se definen tareas relativas a la actividad del jugador como disparar con frecuencias mínimas que se tienen que cumplir para poder decir que ese jugador es poco activo, también se comparan algunos de sus atributos como el número de enemigos que el jugador ha eliminado, en este caso es cero, como resultado a este comportamiento se manda a ejecutar una tarea que muestra un mensaje de advertencia a los compañeros del jugador indicándoles sobre su comportamiento para que puedan tomar las medidas necesarias.

Cuadro 4.4: Guión para jugador novato

Regla	**Frecuencia**
$Actor(Player\{*\}).enemyKills < 2$	no aplica
& $Actor(Player\{*\}).deads > 2$	no aplica
& $Actor(Player\{*\}) - Task(Shot)$	5
& $Actor(Player\{*\}) - Task(scoreCapturedFlag)$	1
& $Actor(Player\{*\}) - Task(loseFlag)$	1
& $[Actor(Player\{*\}) \rightarrow Team(red)] - Task(Shot).Success = true$	5
& $[Actor(Player\{*\}) \rightarrow Team(red)] - Task(Shot).Dimension = positive$	5
:=	
$[Actor(Player\{*\}) \rightarrow Team(red)] - Task(sendMessage, Object(UI\{messageConsole\})$	no aplica

Para que un jugador sea considerado como novato, se tomaron en cuanta el número de muertes

enemigas que ha causado el jugador en un intervalo de tiempo, que haya capturado la bandera al menos una vez, y que los disparos que haya hecho sean efectivos con cierta frecuencia. Este guión es descrito en el Cuadro 4.4.

Cuadro 4.5: Guion para jugador experto

Regla	**Frecuencia**
$Actor(Player\{*\}).enemyKills > 5$	no aplica
& $[Actor(Player\{*\}) \rightarrow Team(red)] - Task(Shot).Success = true$	5
& $[Actor(Player\{*\}) \rightarrow Team(red)] - Task(Shot).Dimension = positive$	5
& $[Actor(Player\{*\}) \rightarrow Team(red)] - Task(Destroy).Bearer = true$	1
& $Actor(Player\{*\}) - Task(captureFlag)$	3
& $Actor(Player\{*\}).socialPresence = quitegood$	no aplica
& $Actor(Player\{*\}).deads < 2$	no aplica
:=	
$[Actor(Player\{*\}) \rightarrow Team(red)] - Task(showPosition, Object(UI\{map\})$	no aplica
& $[Actor(Player\{*\}) \rightarrow Team(red)] - Task(sendHelpMessage, Object(UI\{messageConsole\})$	no aplica

El guión mostrado en el Cuadro 4.5 especifica el comportamiento de un jugador que tiene experiencia jugando el juego, para esto se mide la cantidad de enemigos eliminados haciéndose efectivo para más de 5 muertes en un lapso de tiempo dado, además se cuentan las veces que el jugador ha capturado una bandera o ha marcado puntos, y al igual que el jugador inactivo se mide su presencia social la cual tiene que ser alta para ser considerado un jugador experto. Como resultado al cumplimiento de este guión se activa la posición de este jugador experto en el mapa, y avisa a los demás que se mantengan cerca de él para que se apoyen mutuamente.

Por último se define un guión(Cuadro 4.6) para un jugador cuyo comportamiento afecta a los miembros del mismo equipo, como disparos a sus compañeros, también se mide el número de disparos que hizo contra enemigos si son menores o igual a uno entonces ese jugador se considera un traidor, y como medidas computacionales se envía la tarea de bloquear armamento y un mensaje a sus aliados donde se avise de su comportamiento para que tomen la estrategía adecuada.

4.2. Prototipo

El prototipo de la arquitectura, mostrado en la Figura 4.4, sigue la misma estructura que el diseño conceptual dividido en tres capas: recuperación de datos, gestion de datos y uso contextual. En la capa de recuperación de datos se encuentran dos tipos de servicios, los que reciben las tareas

Cuadro 4.6: Guion de jugador traidor

$Actor(Player\{*\}) - Task(captureFlag)$	1
$\&\ [Actor(Player\{*\}) \rightarrow Team(RVSF)] - Task(Shot) - Affects([Actor(Player\{*\}) -> Team(red)])$	3
$\&\ Actor(Player\{*\}).socialPresence = bad$	no aplica
$\&\ Actor(Player\{*\}).allyKills > 2$	no aplica
$\&\ [Actor(Player\{*\}) \rightarrow Team(RVSF)] - Task(Shot).outcome > 0$	0
:=	
$[Actor(Player\{*\}) \rightarrow Team(RVSF)] - Task(blockWeaponry, Object(Weapon\{*\})$	no aplica
$\&\ [Actor(Player\{*\}) \rightarrow Team(RVSF)] - Task(sendWarningMessage, Object(UI\{messageConsole\})$	no aplica

ejecutadas en la aplicación colaborativa, en este caso ***AssaultCube***, y los que recibe actualizaciones del estado de los elementos que participan en las actividades llevadas a cabo en el groupware.

En la capa de gestión de datos se implementan dos servidores, SQL Server como motor de base de datos e Internet Information Services 7 (IIS7) como servidor web. Dentro del servidor de base de datos se encuentra el modelo conceptual y los modelos de dominio que se generen; dentro del servidor web se encuentran las aplicaciones necesarias para definir e instanciar los modelos y servicios usados para recibir los datos durante la ejecución de *AssaultCube*, la primera aplicación es la Plataforma web de Definición de Dominio de Aplicación(**PDDA**) programada en el marco de trabajo de ASP.NET. Dentro de esta aplicación se encuentran formularios para definir elementos del dominio y para el registro de nuevos casos de estudio, los elementos definidos se almacenan para que puedan ser reutilizados en otro casos de estudio; también cuenta con un módulo para definir los guiones de comportamiento, en este se escriben las reglas para cada guion y son validadas por un parser en la capa de uso de contexto. La segunda aplicación es un módulo programado en el lenguaje C♯ con la plataforma .NET versión 4.0.0, que genera las instancias del modelo definido en la plataforma implementando técnicas de programación reflexiva, una de las instancias es la del modelo de dominio en el servidor de base de datos, otra es el modelo de clases y el tercero son los servicios propios del dominio en el servidor web IIS7.

Dentro de la capa de uso de contexto se encuentran el intérprete de las reglas usado por la plataforma web y el motor de inferencia que es el objeto principal de este trabajo, este motor accede al modelo de dominio en la base de datos para recuperar los guiones de comportamiento, crear los objetos relativos a los guiones y verificarlos cada vez que se reciben ejecuciones de tareas desde *AssaultCube*, cuando el motor detecta que un guión se cumple recupera las tareas resultantes descritas en dicho guión y un servicio de resultados lo envía al groupware para su interpretación y ejecución.

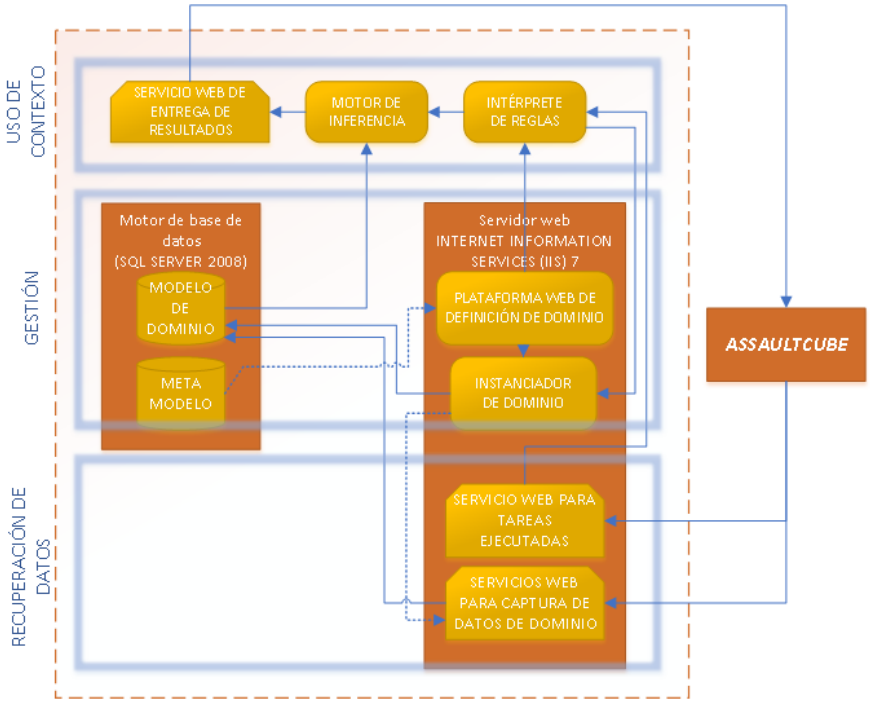

Figura 4.4: Arquitectura de implementación

4.3. Intérprete de reglas y Motor para de inferencia de guiones de comportamiento

Para el presente trabajo se proponen un lenguaje, validado por un intérprete, diseñado para describir las interacciones que se llevan a cabo en una actividad y las condiciones que se deben de cumplir para asumir que un usuario tiene un comportamiento en particular.

Para el intérprete se implementó una gramática libre de contexto asociada a la estructura de los diferentes tipos de reglas propuestas, esta gramática es la siguiente:

```
S->ML:=XJ       //Elemento      //Pertenencia
M->P            E->m(e{iH})     P->EOE
M->C            E->m(*)         O->->
M->X            E->m(e{*})      O->~>
J->&XJ          E->m(iH)
J->$            H->,iH
L->&ML          H->$
L->$
```

```
//Comparacion   //ejecucion de tareas
C->E.aKv        X->F-t(n,F)Q
K->=            X->F-t(n)Q
K->~            Q->.aKv
K->>            Q->$
K->>            F->E
```

A partir de esta gramática se implementa un autómata SLR en el lenguaje C♯ usando la plataforma .NET versión 4.0, éste determina a validez sintáctica de una regla. Antes de este proceso se verifica con expresiones regulares que las reglas cumplan con el patrón de los tipos de reglas definido, este análisis regresa como resultado una serie de carácteres que después son análizados por el autómata.

El motor de inferencia también es implementado en el lenguaje C♯ bajo la plataforma .NET 4.0, básicamente el motor recupera los guiones de la base de datos del dominio de la aplicación y los utiliza como lista de verificación para revisar la frecuencia de las tareas definidas en los guiones, teniendo también la opción de ejecutar los guiones verificando que las tareas descritas sólo se ejecuten una sóla vez. El pseudocódigo de el motor es el siguiente:

```
recuperar scripts
    instanciar scrpits
        recuperar tareas
        recuperar comparaciones
        recuperar pertenencias
    mientras corrida de juego
        recibir tarea
        buscar tarea en cada uno de los scripts
        si tarea esta en script
            decrementar contador
            si contador de todas las tareas == 0
                verificar reglas de comparacion
                verificar pertenencias
                si ambas son verdaderas
                    regresar tareas resultantes
                    reiniciar contadores de tareas
```

Para probar el motor se almacenaron los siguientes guiones con sus reglas en la base de datos(Figura 4.5).

El motor es alimentado por cadenas en formato de tareas ejecutadas por cada una de las cuales decrementa un contador en los guiones como se muestra en la Figura4.6

dummy	Actor(Player{*})-Task(CaptureFlag)
dummy	Actor(Player{*}).socialPresense=bad
dummy	Actor(Player{*})->Team(RVSF)
dummy	[Actor(Player{*})->Team(RVSF)]-Task(sendMessage,...
noob	Actor(Player{*}).enemyKills<2
noob	Actor(Player{*}).deads>2
noob	Actor(Player{*})-Task(shot)
noob	Actor(Player{*})-Task(scoreCapturedFlag)
noob	Actor(Player{*})-Task(loseFlag)
noob	[Actor(Player{*})->Team(RVSF)]-Task(shot).success=tr...
noob	[Actor(Player{*})->Team(RVSF)]-Task(shot).outcome=2
noob	[Actor(Player{*})->Team(RVSF)]-Task(sendMessage,...
expert	Actor(Player{*}).enemyKills>5
expert	[Actor(Player{*})->Team(RVSF)]-Task(shot).success=tr...
expert	[Actor(Player{*})->Team(RVSF)]-Task(shot).outcome>0
expert	[Actor(Player{*})->Team(RVSF)]-Task(destroy).bearer=...
expert	Actor(Player{*})-Task(CaptureFlag)
expert	Actor(Player{*}).socialPresense=quiteGood
expert	[Actor(Player{*}).deads<2
expert	[Actor(Player{*})->Team(RVSF)]-Task(showPossition,...
expert	[Actor(Player{*})->Team(RVSF)]-Task(sendMessage,...
traitor	Actor(Player{*})-Task(CaptureFlag)
traitor	[Actor(Player{*})->Team(RVSF)]-Task(shot).outcome<0
traitor	Actor(Player{*}).socialPresense=bad
traitor	Actor(Player{*}).allyKills>2
traitor	[Actor(Player{*})->Team(RVSF)]-Task(shot).outcome>0

Figura 4.5: Datos de los guiones almacenados

Figura 4.6: Conteo de tareas ejecutadas en los guiones

El motor se puede correr de dos formas distintas, una en la que toma en cuenta las frecuencias definidas, y otra en la que cada tarea sólo necesita ejecutarse una vez para que el guion se cumpla. Cuando los contadores de algún guion llegan a cero en las frecuencias de las tareas este valida las demás condiciones del guion y regresa las tareas a ejecutar en el groupware como respuesta como se muestra en la Figura 4.7

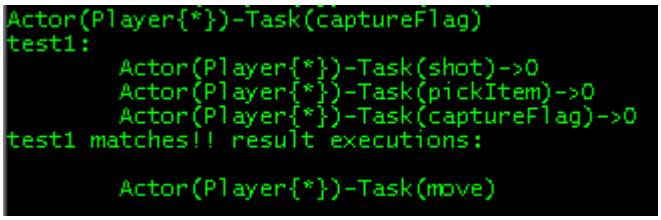

Figura 4.7: Tarea regresada cuando el guion de prueba se cumple

Bibliografía

[1] L. Montane-Jimenez, E. Benitez-Guerrero, and C. Mezura-Godoy, "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*, Oct 2013, pp. 70–76.

[2] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: Some issues and experiences," *Commun. ACM*, vol. 34, no. 1, pp. 39–58, Jan. 1991. [Online]. Available: http://doi.acm.org/10.1145/99977.99987

[3] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," *Network, IEEE*, vol. 8, no. 5, pp. 22–32, Sept 1994.

[4] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Hum.-Comput. Interact.*, vol. 16, no. 2, pp. 97–166, Dec. 2001. [Online]. Available: http://dx.doi.org/10.1207/S15327051HCI16234_02

[5] J. Anhalt, A. Smailagic, D. Siewiorek, F. Gemperle, D. Salber, S. Weber, J. Beck, and J. Jennings, "Toward context-aware computing: experiences and lessons," *Intelligent Systems, IEEE*, vol. 16, no. 3, pp. 38–46, May 2001.

[6] K. Schmidt and L. Bannon, "Taking cscw seriously," *Computer Supported Cooperative Work (CSCW)*, vol. 1, no. 1-2, pp. 7–40, 1992.

[7] J. Pascoe, N. Ryan, and D. Morse, "Issues in developing context-aware computing," in *Handheld and Ubiquitous Computing*, ser. Lecture Notes in Computer Science, H.-W. Gellersen, Ed. Springer Berlin Heidelberg, 1999, vol. 1707, pp. 208–221.

[8] A. Cruz, A. Correia, H. Paredes, B. Fonseca, L. Morgado, and P. Martins, "Towards an overarching classification model of cscw and groupware: A socio-technical perspective," in *Collaboration and Technology*, ser. Lecture Notes in Computer Science, V. Herskovic, H. Hoppe, M. Jansen, and J. Ziegler, Eds. Springer Berlin Heidelberg, 2012, vol. 7493, pp. 41–56. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33284-5_4

[9] T. W. Malone and K. Crowston, "The interdisciplinary study of coordination," *ACM Comput. Surv.*, vol. 26, no. 1, pp. 87–119, Mar. 1994. [Online]. Available: http://doi.acm.org/10.1145/174666.174668

[10] P. Antunes, V. Herskovic, S. F. Ochoa, and J. A. Pino, "Reviewing the quality of awareness support in collaborative applications," *Journal of Systems and Software*, vol. 89, no. 0, pp. 146 – 169, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121213002756

[11] R. Johansen, *GroupWare: Computer Support for Business Teams.* New York, NY, USA: The Free Press, 1988.

[12] D. Pumareja and K. Sikkel, "An evolutionary approach to groupware implementation: the context of requirements engineering in the socio-technical frame," no. TR-CTIT-02-30, pp. 1–27, 2002. [Online]. Available: http://doc.utwente.nl/38248/

[13] C. Ellis and J. Wainer, "A conceptual model of groupware," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '94. New York, NY, USA: ACM, 1994, pp. 79–88. [Online]. Available: http://doi.acm.org/10.1145/192844.192878

[14] P. Antunes, V. Herskovic, S. F. Ochoa, and J. A. Pino, "Structuring dimensions for collaborative systems evaluation," *ACM Comput. Surv.*, vol. 44, no. 2, pp. 8:1–8:28, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/2089125.2089128

[15] D. Mittleman, R. Briggs, J. Murphy, and A. Davis, "Toward a taxonomy of groupware technologies," in *Groupware: Design, Implementation, and Use*, ser. Lecture Notes in Computer Science, R. Briggs, P. Antunes, G.-J. de Vreede, and A. Read, Eds. Springer Berlin Heidelberg, 2008, vol. 5411, pp. 305–317. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-92831-7_25

[16] C. Gutwin, S. Greenberg, and M. Roseman, "Supporting awareness of others in groupware," in *Conference Companion on Human Factors in Computing Systems*, ser. CHI '96. New York, NY, USA: ACM, 1996, pp. 205–. [Online]. Available: http://doi.acm.org/10.1145/257089.257282

[17] P. Dourish and V. Bellotti, "Awareness and coordination in shared workspaces," in *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work*, ser. CSCW '92. New York, NY, USA: ACM, 1992, pp. 107–114. [Online]. Available: http://doi.acm.org/10.1145/143457.143468

[18] F. Belkadi, E. Bonjour, M. Camargo, N. Troussier, and B. Eynard, "A situation model to support awareness in collaborative design," *International Journal of Human-Computer Studies*, vol. 71, no. 1, pp. 110 – 129, 2013, special Issue on supporting shared representations in collaborative activities. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S107158191200050X

[19] M. Aehnelt, C. Peter, and P. Müsebeck, "A discussion of using mental models in assistive environments," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '12. New York, NY, USA: ACM, 2012, pp. 37:1–37:5. [Online]. Available: http://doi.acm.org/10.1145/2413097.2413145

[20] P. Brézillon, M. Borges, J. Pino, and J.-C. Pomerol, "Context-awareness in group work: Three case studies," in *Proc. of*, 2004.

[21] N. Malik, U. Mahmud, and Y. Javed, "Future challenges in context-aware computing," in *proceedings of the IADIS International Conference WWW/Internet*, 2007, pp. 306–310.

[22] J. Pereira de Souza, C. Tacla, F. Beal, E. Cabrera Paraiso, and G. Gimenez-Lugo, "An ontology-based agent for context aware software process development," in *Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on*, June 2013, pp. 287–292.

[23] J. R. Hoyos, J. G. Molina, and J. A. Botía, "A domain-specific language for context modeling in context-aware systems," *Journal of Systems and Software*, vol. 86, no. 11, pp. 2890 – 2905, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121213001696

[24] P. Alves and P. Ferreira, "Radiator: Context propagation based on delayed aggregation," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ser. CSCW '13. New York, NY, USA: ACM, 2013, pp. 249–260. [Online]. Available: http://doi.acm.org/10.1145/2441776.2441806

[25] L. Ardissono and G. Bosio, "Context-dependent awareness support in open collaboration environments," *User Modeling and User-Adapted Interaction*, vol. 22, no. 3, pp. 223–254, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11257-011-9100-1

[26] A. Olivares, S. Mendoza, and A. de Luca, "An architecture to support context of use in groupware systems," in *Electrical Engineering Computing Science and Automatic Control (CCE), 2011 8th International Conference on*, Oct 2011, pp. 1–6.

[27] B. Hu, Z.-X. Wang, and Q.-C. Dong, "A novel context-aware modeling and reasoning method based on owl," *Journal of Computers*, vol. 8, no. 4, 2013.

[28] K.-L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, W. Burns, and I. Solheim, "Ontological user modelling and semantic rule-based reasoning for personalisation of help-on-demand services in pervasive environments," *Future Generation Computer Systems*, vol. 34, no. 0, pp. 97 – 109, 2014, special Section: Distributed Solutions for Ubiquitous Computing and Ambient Intelligence.

[29] J. Gallardo, A. I. Molina, and C. Bravo, "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*, ser. INTERACCION '12. New York, NY, USA: ACM, 2012, pp. 7:1–7:2. [Online]. Available: http://doi.acm.org/10.1145/2379636.2379643

[30] D. Decouchant, S. Mendoza, G. SÃ¡nchez, and J. RodrÃguez, "Adapting groupware systems to changes in the collaborator's context of use," *Expert Systems with Applications*, vol. 40, no. 11, pp. 4446 – 4462, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417413000742

[31] P. Gutheim, "An ontology-based context inference service for mobile applications in next-generation networks," *Communications Magazine, IEEE*, vol. 49, no. 1, pp. 60–66, January 2011.

[32] Y. Oh, J. Han, and W. Woo, "A context management architecture for large-scale smart environments," *Communications Magazine, IEEE*, vol. 48, no. 3, pp. 118–126, March 2010.

[33] A. K. Dey, D. Salber, M. Futakawa, and G. D. Abowd, "An architecture to support context-aware applications," 1999.

[34] Y. El Ghayam and M. Erradi, "Distributed context management in collaborative environment," in *New Technologies of Distributed Systems (NOTERE), 2011 11th Annual International Conference on*, May 2011, pp. 1–8.

[35] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Handheld and Ubiquitous Computing*, ser. Lecture Notes in Computer Science, H.-W. Gellersen, Ed. Springer Berlin Heidelberg, 1999, vol. 1707, pp. 304–307.

[36] B. Chihani, E. Bertin, and N. Crespi, "Programmable context awareness framework," *Journal of Systems and Software*, vol. 92, no. 0, pp. 59 – 70, 2014.

[37] A. Kamoun, S. Tazi, and K. Drira, "Fadyrcos, a semantic interoperability framework for collaborative model-based dynamic reconfiguration of networked services," *Computers in Industry*, vol. 63, no. 8, pp. 756 – 765, 2012, special Issue on Sustainable Interoperability: The Future of Internet Based Industrial Enterprises. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166361512001261

[38] H. Guermah, T. Fissaa, H. Hafiddi, M. Nassar, and A. Kriouile, "Ontology based context aware e-learning system," in *ISKO-Maghreb, 2013 3rd International Symposium*, Nov 2013, pp. 1–7.

[39] T. Gross, "Supporting effortless coordination: 25 years of awareness research," *Computer Supported Cooperative Work (CSCW)*, vol. 22, no. 4-6, pp. 425–474, 2013.

[40] M. Arias-Baez, L. Torres-Ribero, A. Carrillo-Ramos, A. Quimbaya, and E. Gonzalez, "Platform based on agents for support to the collaboration of work teams," in *Computing Congress (CCC), 2012 7th Colombian*, Oct 2012, pp. 1–6.

[41] R. V. Productions. (2014) Documentación de assault cube. [Online]. Available: http://assault.cubers.net/docs/