

Limitaciones del Modelo de Tareas del W3C para aplicaciones Post-WIMP

Miguel A. Teruel, Arturo C. Rodríguez, Francisco Montero,
Elena Navarro, Víctor López-Jaquero and Pascual González

Grupo de Investigación LoUISE

Instituto de Investigación en Informática de Albacete

02071 – Albacete (España)

miguel@dsi.uclm.es, arturo.rodriguez@uclm.es, fmontero@dsi.uclm.es,
elena.navarro@uclm.es, victor@dsi.uclm.es, pgonzalez@dsi.uclm.es

RESUMEN

La evolución tecnológica y la aparición de nuevos dispositivos que ésta conlleva está provocando grandes cambios en el modo en que las personas interactúan con los sistemas software. Cada vez son más los sistemas cuyas técnicas de interacción están basados en realidad virtual, reconocimientos de gestos, dispositivos corporales y muchos otros, que se alejan notablemente de la interacción clásica de los sistemas denominados WIMP (Windows, Icons, Menus, Pointers). Estos nuevos sistemas, a los que se denomina sistemas Post-WIMP, necesitan, por lo tanto, que los lenguajes para la especificación de sistemas evolucionen de igual forma para poder hacer frente al incremento de la complejidad y a los nuevos conceptos inherentes a sus formas de interacción. En este artículo se presenta un lenguaje de modelado de tareas de sistemas Post-WIMP, basado a su vez en CSRML, como alternativa al actual meta-modelo de tareas propuesto por el W3C. Este nuevo lenguaje tiene como finalidad especificar las tareas que serán realizadas por el usuario a través de la interfaz de sistemas Post-WIMP con la expresividad de la que carece el actual meta-modelo de tareas del W3C. Cabe destacar principalmente dos ventajas de esta nueva propuesta respecto a la del W3C. Por un lado, se han incluido varios elementos y relaciones para enriquecer el nuevo meta-modelo con la expresividad necesaria para llevar a cabo la especificación de la colaboración de forma detallada. Por otro lado, nuestra propuesta permite representar la información de awareness que los usuarios Post-WIMP deberían tener durante la interacción con el sistema. Para mostrar la utilización de este nuevo meta-modelo, presentamos un ejemplo de modelado basado en un juego “first person shooter” Post-WIMP con una interfaz de realidad virtual.

Categorías y descriptores de tema

H.5.2 [User Interfaces]: Interaction styles., I.3.6. [Methodology and Techniques]. Interaction techniques.

Términos Generales

Modelado, Interacción, Experimentación, Lenguajes.

Palabras clave

Post-WIMP; CSRML; W3C task model; collaboration; awareness.

1. INTRODUCCIÓN

Durante las últimas dos décadas, investigadores en Interacción Persona-Ordenador (HCI) han desarrollado una amplia gama de nuevas interfaces que se alejan de las tradicionales WIMP (Window, Icon, Menu, Pointing). Estas interfaces WIMP presentan carencias para llevar a cabo tareas complejas como el diseño asistido por ordenador, la manipulación simultánea de una

gran cantidad de datos o los juegos altamente interactivos. También tienen deficiencias en aplicaciones que requieren una monitorización continua de señales de entrada, representación de modelos 3D o simplemente en interacciones para las que no hay widgets estándar. Por esta razón, las interfaces de usuario personalizadas se han convertido en la mejor alternativa a la hora de realizar tareas complejas.

Post-WIMP es un término usado para describir interfaces que van más allá de los paradigmas habituales (ventanas, iconos, menús y punteros) [1]. Esta nueva generación de interfaces ha sido impulsada tanto por los avances en tecnología como por un mejor entendimiento de la psicología humana. Definidas por van Dam [1] como interfaces “*que contienen al menos una técnica de interacción no dependiente de los clásicos widgets 2D, como menús o iconos*”, algunos ejemplos de interacciones Post-WIMP son: (i) realidad virtual, aumentada o mixta, (ii) interacción tangible, (iii) computación ubicua, (iv) computación sensible al contexto, (v) interacción con dispositivos móviles, (vi) computación perceptual, e (vii) interacción ligera, tácita o pasiva. Aunque algunas personas podrían clasificar estos estilos de interacción como innovaciones dispares, futuristas y heterogéneas, comparten importantes características como una considerable demanda de información de contexto que ayude a entender, conectar y analizar el trabajo que se lleva a cabo.

Esta evolución de las interfaces requiere que los lenguajes para el análisis de requisitos y la especificación de sistemas sean adaptados de forma adecuada, pudiendo así recoger las necesidades de información de estos nuevos paradigmas de interacción. El análisis de tareas no es una excepción y, a pesar de que los meta-modelos actuales han supuesto una herramienta eficaz en el desarrollo de sistemas WIMP, las nuevas interfaces Post-WIMP introducen la necesidad de revisarlos. Además, el análisis de tareas está estrechamente relacionado con la interacción, puesto que, como afirman Gea y Gutiérrez [2], *cubre diferentes técnicas orientadas a describir las interacciones entre las personas y los entornos de una manera sistemática* y tiene como objetivo obtener una descripción formal del conjunto de acciones que debe realizar el usuario para lograr un fin; acciones que pueden ser tanto cognitivas como interacciones con la interfaz.

En este artículo, proponemos una alternativa al actual meta-modelo de tareas del W3C [3] considerando dos de los principales aspectos clave de los sistemas Post-WIMP: *awareness* (habilidad del usuario de ser consciente de lo que le rodea) y *colaboración* entre usuarios. La necesidad de incorporar dichos conceptos surge, no sólo del estudio de la literatura relacionada, sino

también de la amplia experiencia acumulada en nuestro grupo de investigación a lo largo de los últimos quince años en el desarrollo de interfaces post-WIMP, que se inició con sistemas de interacción 3D [4].

Como base para esta propuesta, se ha elegido el lenguaje CSRML (Collaborative Systems Requirements Modelling Language, [5][6]) ya que dispone de capacidad expresiva para especificar las necesidades de colaboración y awareness. Además, otra razón para escoger este lenguaje es que ha sido evaluado para tratar con diferentes dominios, como la edición colaborativa de documentos [5] o la inteligencia de negocio [7]. No obstante, el modelo de tareas CSRML ha sido ampliado y modificado para conseguir dos objetivos. En primer lugar para hacerlo compatible con el modelo W3C y en segundo lugar, para enriquecerlo con elementos de awareness ligados a sistemas Post-WIMP, ya que CSRML estaba diseñado para atender básicamente a las necesidades de los sistemas colaborativos.

Este artículo está organizado de la siguiente manera. La Sección 2 presenta el estado de la cuestión relacionada con el actual meta-modelo de tareas propuesto por la W3C. En base a las deficiencias detectadas en la propuesta de la W3C, en la Sección 3 se presenta el meta-modelo de tareas para interfaces Post-WIMP. Para que puedan apreciarse más claramente las diferencias de la propuesta de la W3C y la nuestra, en la Sección 4 se presenta un ejemplo de modelado basado en un juego Post-WIMP que se ha modelado usando ambos meta-modelos de tareas. Finalmente, en la Sección 5 se presentarán algunas conclusiones.

2. META-MODELO DE TAREAS DEL W3C

Gea y Gutiérrez [2] definen el análisis de tareas como *el estudio de lo que un usuario tiene que realizar en términos de acciones y/o procesos cognitivos para conseguir un objetivo*. Este análisis se traduce en modelos de tareas que tradicionalmente, han proporcionado una descripción eficaz de sistemas interactivos WIMP. Representan información estática de las tareas de aplicación y de usuario a dos niveles: *estructural* (descomposición de tareas) y *temporal* (relaciones temporales). Cada tarea especificada en el modelo de tareas es una actividad o proceso cuya realización es necesaria para conseguir los objetivos del usuario. Además, los analistas pueden representar tareas en diferentes niveles de abstracción, por lo que pueden considerar una actividad como una tarea atómica o descomponerla en subtareas, con el objetivo de proporcionar una especificación con mayor detalle, y después, establecer relaciones temporales entre ellas. Sin embargo, esta descripción no tiene en cuenta detalles sobre cómo se llevan a cabo los procesos e interacciones, ambos necesarios para el correcto diseño del sistema.

El modelo de tareas del W3C está basado en la notación ConcurTaskTrees (CTT), ampliamente aceptada [8] y tiene como objetivo definir un estándar para el modelado de tareas. El meta-modelo de tareas del W3C proporciona una estructura jerárquica entre tareas y ofrece varios operadores, usados también en CTT, para definir las relaciones temporales entre tareas. Además, proporciona una taxonomía de tipos de tareas. Los elementos del meta-modelo de tareas (**Error! No se encuentra el origen de la referencia.**) son los siguientes:

- *Task*: acción que se realiza desde la perspectiva del usuario. Desde el punto de vista estructural, hay dos tipos de tareas: Tareas atómicas (acciones indivisibles) y tareas compuestas (divisibles en subtareras). Además,

pueden definirse cuatro tipos de tareas: (i) *User*, o tareas de usuario: actividad cognitiva que no se realiza directamente a través de la interfaz. (ii) *System*, o tarea de sistema: acción realizada por la aplicación. (iii) *Interaction*: acciones realizadas por los usuarios que dan lugar a respuestas instantáneas del sistema y (iv) *Abstract*: tareas que se descomponen en subtareas de distintas categorías.

- *Domain Object*: un objeto del modelo de dominio relacionado con una tarea.
- *Condition Group*: operación lógica que puede ser usada para especificar una precondición y/o una post-condición de una tarea.
- *Logic Operator*: operador que conecta los operandos del Condition Group y especifica el tipo de operación lógica.
- *Condition Literal*: valor u objeto del modelo del dominio usado como operando en una operación lógica.
- *N-ary Operator*: operador que establece una restricción temporal entre subtareas.
- *1-ary Operator*: operador que establece si una tarea es opcional y/o iterativa.

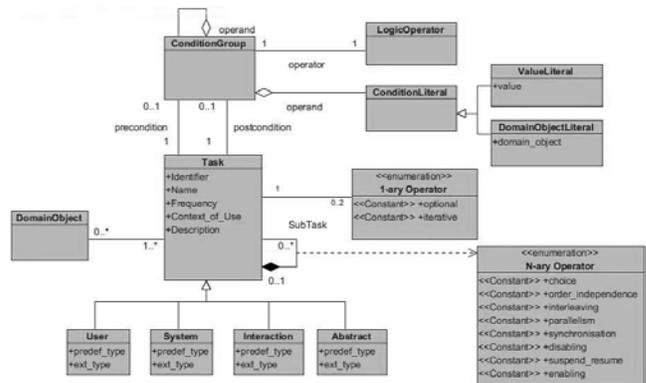


Figura 1. Meta-modelo actual del W3C [3]

El meta-modelo mostrado en la Fig. 1 permite al analista describir modelos de tareas de sistemas WIMP de forma eficaz. Este meta-modelo usa las relaciones de composición para especificar estructuras jerárquicas de tareas y asocia los operadores temporales a estas relaciones. Aunque estos operadores son suficientemente expresivos para describir dichas relaciones, sería razonable incluir una nueva meta-clase específica para establecer relaciones temporales entre tareas, asociando el operador a dicha clase, permitiendo así desligar las relaciones estructurales de las temporales y estableciendo un único operador para cada relación temporal.

Las aplicaciones actuales tienden claramente a ser colaborativas (por ejemplo, juegos, editores de texto, redes sociales, IDEs, etc.) con un considerable número de usuarios interactuando concurrentemente. Además, las interfaces de usuario de dichas aplicaciones han evolucionado enormemente hacia el paradigma Post-WIMP. Así, Jacob et al. [9] identificaron los siguientes cuatro aspectos del mundo real que una aplicación Post-WIMP debería soportar:

- *Naïve Physics*: las personas tienen conocimiento general del mundo físico.
- *Body Awareness & Skills*: las personas tienen conciencia/conocimiento de sus propios cuerpos y poseen habilidades para controlar y coordinar sus cuerpos.
- *Environment Awareness & Skills*: las personas son conscientes de lo que les rodea y poseen habilidades para negociar, manipular y moverse por el entorno.
- *Social Awareness & Skills*: las personas son generalmente conscientes de otras personas en su entorno y tienen habilidades para interactuar con ellos.

Como se puede ver, todos estos aspectos están principalmente relacionados con el concepto de awareness [10]. Por ejemplo, la propiocepción del usuario en el entorno físico juega un papel fundamental en entornos de realidad virtual en los que puede existir una interacción con cualquier parte del cuerpo. Una representación virtual del cuerpo y otras informaciones, como la localización virtual pueden mejorar la experiencia de usuario al interactuar con sistemas Post-WIMP. El conocimiento de los grupos o equipos a los que pertenecen los usuarios es un ejemplo de Social Awareness. Otro ejemplo de awareness lo encontramos en sistemas en los que varios usuarios están editando un documento al mismo tiempo, necesitan saber con quién están colaborando y qué cambios están haciendo en el documento. Esta necesidad ha sido ampliamente analizada en Computer Supported Collaborative Work (CSCW). Ellis et al. [11] identificaron las fuertes relaciones entre awareness y colaboración en el modelo de colaboración 3C. Debido a que awareness y colaboración están muy extendidos en las aplicaciones contemporáneas, especialmente en entornos virtuales colaborativos (Collaborative Virtual Environments CVE) [12], un meta-modelo de tareas debería permitir especificar tales características. Por lo tanto, puesto que la especificación actual del W3C fue originalmente diseñada para especificar entornos WIMP, en este artículo presentamos un meta-modelo de tareas extendido para entornos Post-WIMP basado en CSRML con el fin de especificar estos innovadores sistemas.

3. UN LENGUAJE DE MODELADO DE TAREAS PARA ENTORNOS POST-WIMP

Una vez se han identificado los problemas y limitaciones del meta-modelo de tareas del W3C para especificar sistemas Post-WIMP, presentamos en esta sección el meta-modelo de tareas CSRML [5] adaptado para la especificación de tareas Post-WIMP.

Se ha escogido este lenguaje porque soporta la especificación de requisitos de sistemas colaborativos, uno de los elementos principales de los sistemas Post-WIMP. CSRML es un lenguaje de modelado de requisitos orientado a objetivos para la especificación de sistemas colaborativos respaldado por una herramienta empíricamente validada [13]. Los elementos usados para el modelado de tareas se describen a continuación (ilustrados en Fig. 3 y Fig. 4). Además, la representación gráfica de dichos elementos se puede ver en la Fig. 2.

- *Task*: El concepto de tarea del W3C ha sido extendido para incluir la importancia usando un código de colores (verde, amarillo, naranja y rojo) siendo el color verde el de menor importancia y el rojo el de mayor importancia. Además, se han identificado cuatro tipos de tareas: (i) *Abstract Task*: Abstracción de un conjunto de tareas concretas, (ii) *User Task*: Tarea realizada por un usuario sin necesidad de interactuar con la interfaz e (iii) *Interaction Task*: Tarea que implica una interacción con la interfaz o respuestas del sistema. Hay cuatro tipos de tareas de interacción: *Individual Task* es una tarea que no conlleva colaboración (puede ser realizada por un participante). Estas tareas contemplan también las tareas de sistema, ya que éste es considerado como un actor más dentro del modelo. *Collaboration/Communication/Coordination tasks* son tareas que requieren al menos dos participantes para poder ser realizadas, con el objetivo de llevar a cabo algún tipo de colaboración / comunicación / coordinación entre ellos.
- *Legacy Elements*: Con el fin de hacer este modelo de tareas compatible con el del W3C, varios elementos han sido añadidos a su meta-modelo.
- *Role*: Establece el comportamiento de un actor para un conjunto de tareas relacionadas. Un actor que desempeña un rol puede participar en una tarea individual o colectiva (mediante las relaciones de participación). Cabe señalar que la relación entre un actor y un rol es especificada en el modelo de usuario, no en el modelo de tareas.
- *Goal*: Contesta a la pregunta “¿por qué?”. Describe un estado del mundo que el usuario querría alcanzar. Sin embargo, un *Goal* (objetivo) no describe cómo debería alcanzarse dicho estado.

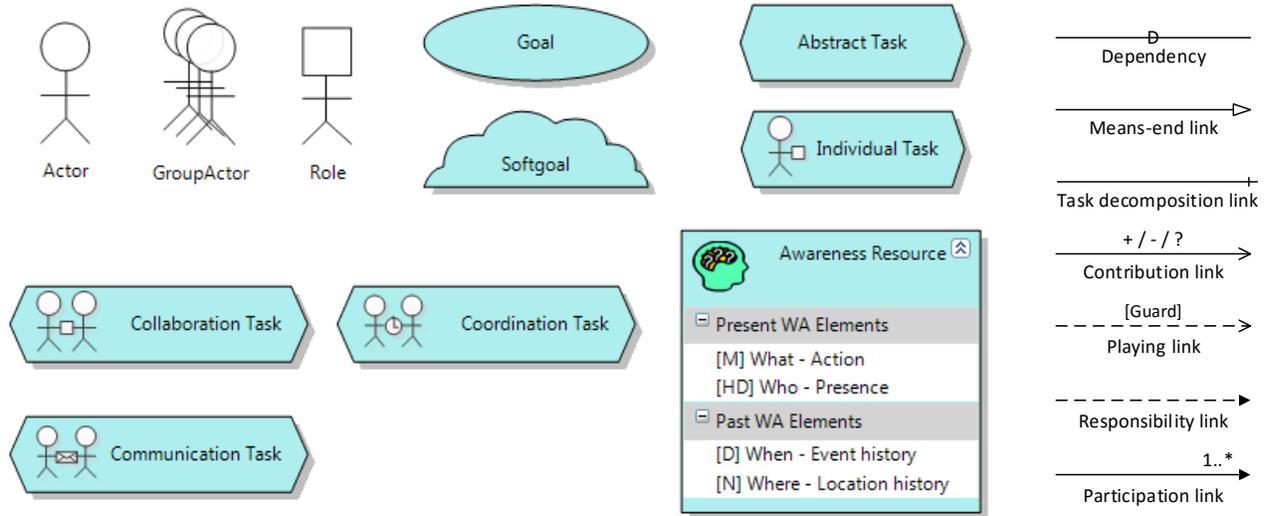


Figura 2. Representación gráfica de los elementos del meta-modelo de tareas para sistemas Post-WIMP

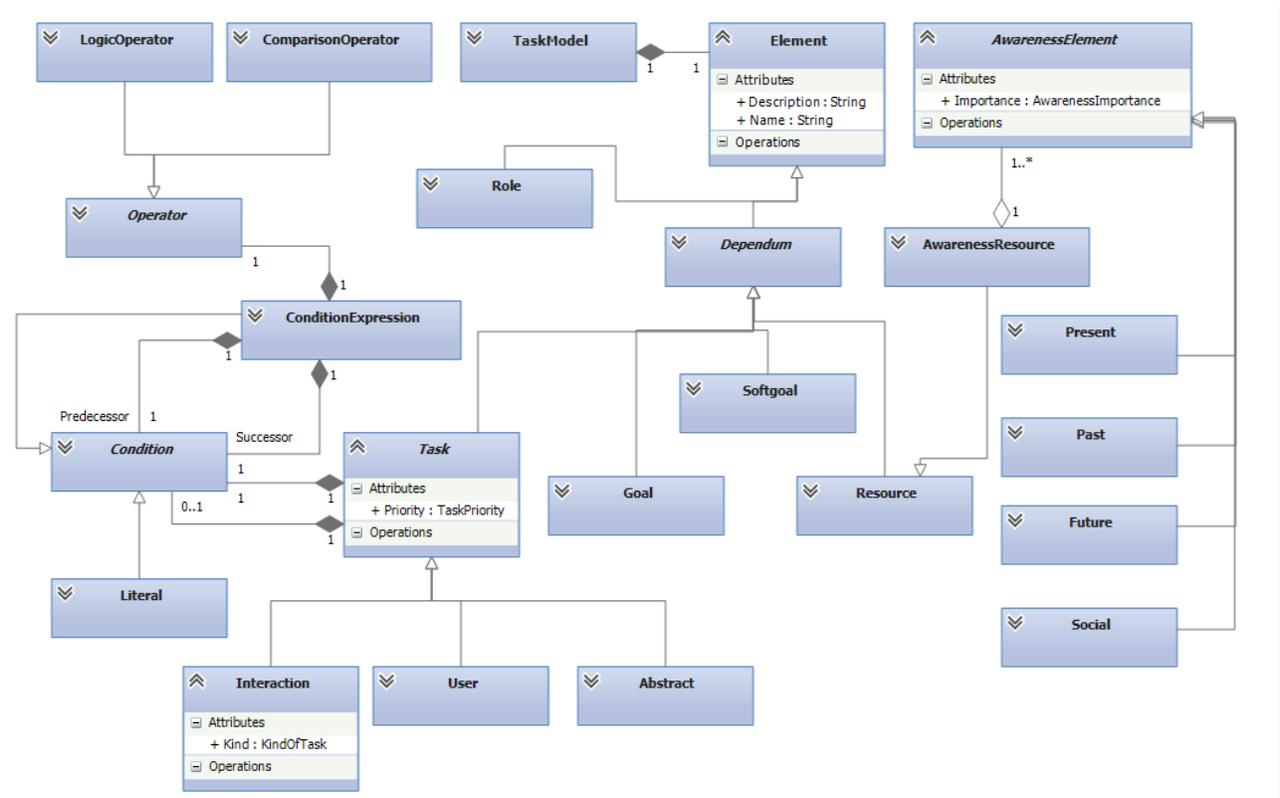


Figura 3. Meta-modelo de tareas para sistemas Post-WIMP (Elementos)

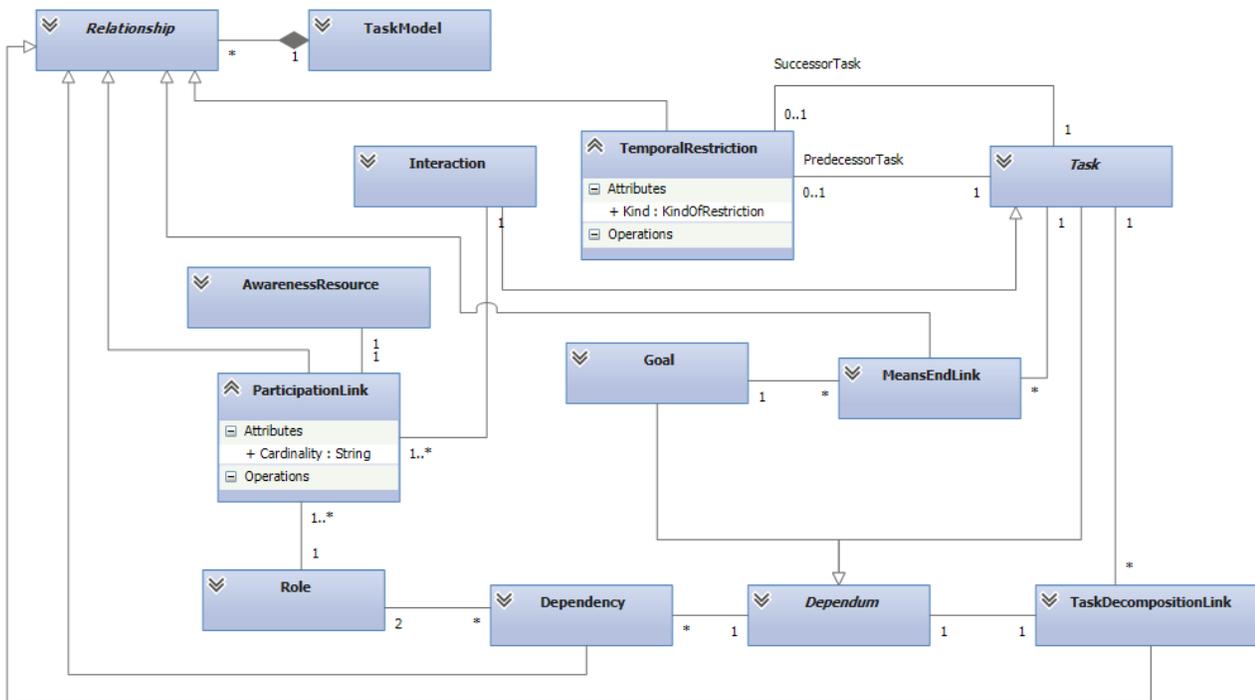


Figura 4. Meta-modelo de tareas para sistemas Post-WIMP (Relaciones)

- *Softgoal*: Es una condición del mundo que el usuario quiere alcanzar, pero a diferencia del concepto *Goal*, esta condición no está claramente definida. Un *Softgoal* es normalmente un atributo cualitativo que restringe otros elementos, como un objetivo, una tarea o un recurso.
- *Resource*: Es una entidad que el usuario necesita para alcanzar un objetivo o realizar una tarea. El principal interés de un recurso es saber si está disponible y de quién.
- *Awareness Resource*: Este elemento representa información de *awareness* que un usuario necesita para completar una tarea. Este elemento es representado como un conjunto de elementos de *Awareness* (*Awareness Elements*) ligados a una relación de participación entre un rol y una tarea.
- *Awareness Element*: Un *Awareness Resource* muestra todos los *Awareness Elements* disponibles (si son necesarios) e indica su importancia en función de la contribución a la realización de la tarea. Estos elementos de *awareness* son una extensión de los identificados por Gutwin y Greenberg [14] para entornos Post-WIMP con el propósito de ser compatibles con los diferentes elementos de *awareness* identificados por Jacob et al. [9]. Por ejemplo, el elemento *Presence – Where – Reach* está relacionado con el elemento de Jacob *Body awareness*; el elemento *Future – Where – Next Position* está relacionado con el *conocimiento del entorno*; y *Social & group dynamics – What – Group goal* es un ejemplo de *Social Awareness*. Además, estos elementos de *awareness* están clasificados en cuatro categorías relacionadas con el presente, el pasado, el futuro y aspectos sociales. Por último, su importancia puede ser establecida como *nice to have* (N), *desirable* (D), *highly desirable* (HD) o *mandatory* (M).
- *Participation Link*: denota quién está involucrado en una tarea. Esta relación tiene un atributo para especificar su cardinalidad, es decir, el número de usuarios que pueden involucrarse en la tarea. Opcionalmente puede tener un *Awareness Resource* ligado que representa que el *Role* tiene una necesidad especial de percepción para participar en la tarea. Sin esta percepción, la realización de la tarea se vería afectada negativamente o incluso el *Role* podría ver impedida su participación en la tarea.
- *Task Decomposition Link*: Describe elementos esenciales de una tarea. Un *Task Decomposition Link* relaciona una tarea con subtareas a realizar, *subgoals* a conseguir, *Resources* necesarios y *softgoals* que normalmente definen la calidad de los objetivos para la tarea.
- *Means-end Link*: documenta qué *softgoal*, *task* o *resource* contribuyen a alcanzar un objetivo.
- *Dependency*: Es una relación ente un *depender* y un *dependee* para un *dependum*. El *depender* y el *dependee* son actores y el *dependum* puede ser un *goal*, una *task* (tarea), un *resource* o un *softgoal*. El *depender* depende del *dependee* para alcanzar un objetivo, realizar una

tarea o usar un recurso. Si el *dependee* no puede proporcionar un *dependum* al *depender* la realización de la tarea, alcanzar el objetivo o usar el recurso se vuelve difícil o imposible.

4. MODELADO DE UN SISTEMA POST-WIMP

Para analizar el meta-modelo propuesto, presentamos un ejemplo de modelado que consiste en un juego *First Person Shooter* (FPS) multi-dispositivo de realidad virtual. Este sistema hará uso de diferentes dispositivos para facilitar la interacción entre el usuario y el sistema. En primer lugar utilizará un dispositivo de visualización inmersivo para proporcionar al usuario los gráficos 3D así como para capturar los movimientos de cabeza del usuario para controlar la dirección de la visión de la representación virtual del usuario (personaje). Para proporcionar al usuario sonido posicional, permitiéndole saber la dirección de los disparos y los pasos de otros personajes, se dispone de auriculares de sonido envolvente, los cuales incorporan un micrófono para comunicarse con otros usuarios. Por otra parte para incorporar sensaciones hápticas se situarán varios actuadores vibrotáctiles sobre el cuerpo del usuario para permitir que éste sienta en qué lugar del cuerpo y cuándo el personaje es alcanzado. Se situarán 7 dispositivos colocados en cabeza, brazos, pecho y espalda. Para controlar la posición de la mano dominante del usuario que simula el arma, se utilizará un tracker que recoja la posición de la mano. Por lo tanto, la dirección y el disparador/gatillo del arma del personaje serán controlados con este dispositivo. Finalmente, en su mano no dominante, el usuario dispondrá de un mando para controlar los movimientos del personaje (adelante, atrás y a los lados).

Por otra parte, las tareas que un usuario realizará durante el juego son: *DefeatEnemy* (tarea principal), *Move*, *Communicate* (dependerá de la distancia entre los jugadores), *Revive Player* (dos jugadores pueden devolver la vida a otro jugador del mismo

equipo), *Shoot*, *Aim*, *Pull The Trigger*, *Get Power-ups*, *Deploy Power-up* (tarea de sistema), *Show End* (tarea de sistema). Además, puesto que el juego a implementarse es tanto colaborativo como Post-WIMP, puede proporcionarse a los usuarios información de *awareness* sobre el entorno del juego (por ejemplo, estado del jugador, posición de otros personajes, terreno de juego, etc.). Concretamente, la información de *awareness* que será proporcionada por el sistema se describe en la Tabla 1.

Una vez explicadas las tareas del juego y las necesidades de *awareness*, todo ello será modelado usando tanto el actual meta-modelo de tareas de la W3C (en adelante W3CTM) como nuestra propuesta PWTM (Post-WIMP Task Meta-model).

Tabla 1. Información de awareness del FPS

Tipo de Awareness	Descripción
Own status	Información sobre el estado de salud del personaje y los <i>power-ups</i> activos
Connected players	Lista que mostrará tanto los jugadores conectados como sus equipos.
Allies location	Localización de los usuarios miembros del mismo equipo es mostrada en un mini-mapa
Power-ups position	Las nuevas mejoras del arma y del personaje que se han desplegado en el juego se muestran en el mini-mapa.
Received Damage	Cuando el personaje de un usuario es alcanzado por una bala, el usuario sentirá una vibración en su propio cuerpo, obteniendo así información de <i>awareness</i> de la dirección del disparo (por ejemplo si el personaje es alcanzado desde atrás, el usuario sentirá una vibración en su espalda).
Caused Damage	Cuando un personaje es alcanzado por un disparo se proporciona información auditiva.
Steps and Shooting sound	Las armas y los pasos de los personajes producirán sonidos que pueden dar información de <i>awareness</i> sobre su localización.

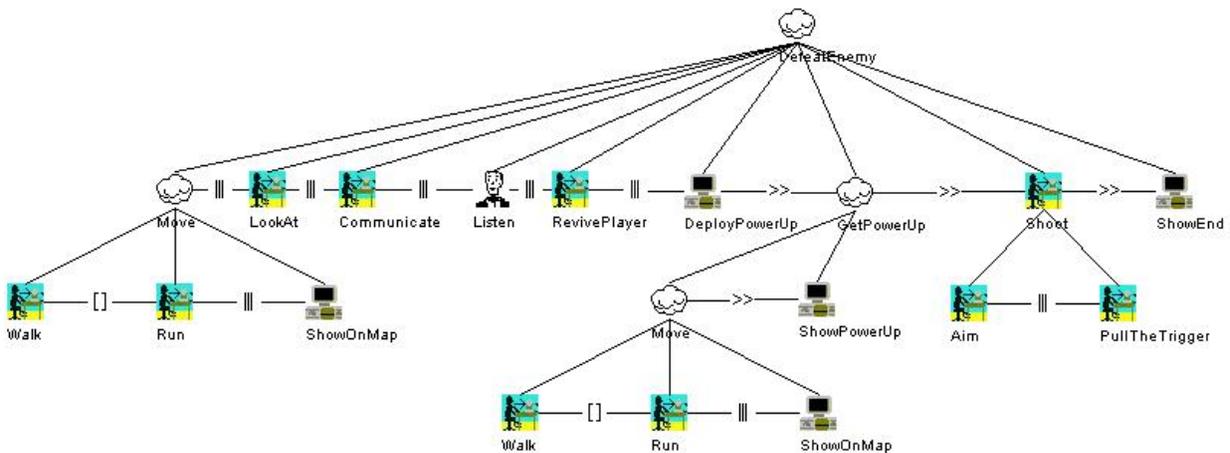


Figura 5. Especificación de tareas usando el meta-modelo del W3C

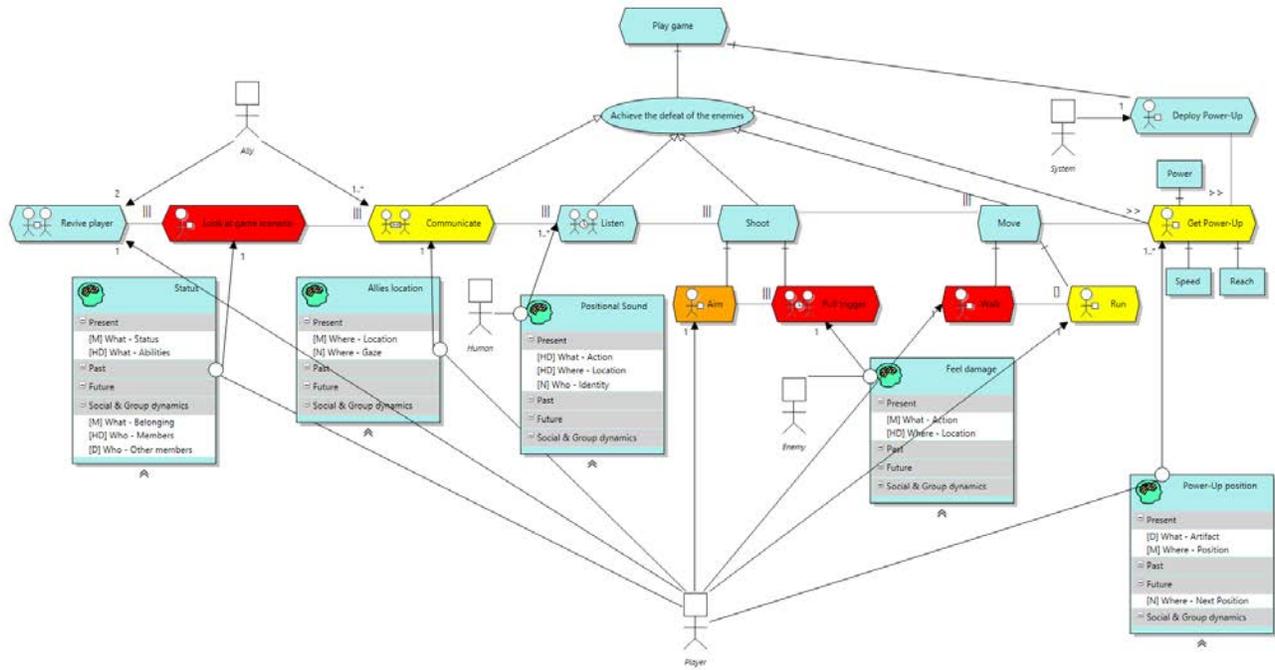


Figura 6. Especificación de tareas usando el meta-modelo Post-WIMP

4.1 USANDO W3CTM PARA ESPECIFICAR LAS TAREAS DEL JUEGO

La Fig. 5 muestra una instancia del modelo obtenido usando el meta-modelo de la W3C para nuestro ejemplo de modelado. Como puede observarse, la tarea *DefeatEnemy* es la raíz del árbol de tareas. Esta tarea se descompone en otras subtareas. *Move*, *LookAt*, *RevivePlayer*, *Communicate*, *Listen* y *DeployPowerUp* que pueden ser realizadas de forma concurrente. *DeployPowerUp* habilita la tarea *GetPowerUp* puesto que ésta no puede ser realizada si no hay *power-ups* en el escenario. *GetPowerUp* habilita la tarea *Shoot* y ésta, *ShowEnd*. La tarea *Move* puede ser realizada de dos formas diferentes, por lo que se descompone *Walk* y *Run* y el operador *N-Ary* entre estas tareas será el operador *Choice*. Además, el sistema muestra el movimiento sobre el mapa de forma concurrente. Para conseguir *power-ups*, el usuario debe moverse a través del escenario virtual. Cuando el personaje pasa a través de un ítem virtual que representa el *power-up*, el sistema informa al usuario mediante la tarea *ShowPowerUp*. La tarea *Shot* se descompone en dos subtareas: *Aim* y *PullTheTrigger*. Ambas tareas pueden ser realizadas de forma concurrente. Cuando un equipo gana un juego, el sistema informa a los usuarios del final. El modelo, mostrado en la Fig. 5, presenta varias deficiencias de cara a posteriores fases del proceso de desarrollo. En primer lugar, no aparecen los participantes de las tareas ni el concepto de rol, lo que impide describir quiénes intervienen en tareas como *Communicate* o *RevivePlayer* y el papel que juegan en las mismas. Por otro lado, este modelo no nos permite definir la información de awareness necesaria para la realización de algunas tareas. Por ejemplo, la tarea *Communicate*, según la propia especificación del ejemplo, sólo puede llevarse a cabo cuando la posición de los participantes hace que estén lo suficientemente cerca, por lo que esta información es relevante para el desarrollo del sistema. Por último, en un entorno virtual de inmersión completa, la tarea de dirigir la mirada hacia distintas zonas del escenario requiere de información de awareness que está

relacionada con la propiocepción del usuario y de las dinámicas sociales y de grupo.

4.2 Post-WIMP Task Meta-model PARA ESPECIFICAR LAS TAREAS DEL JUEGO

Tras especificar las tareas del ejemplo de modelado usando el meta-modelo de la W3C, se procederá al modelado del mismo sistema usando nuestra propuesta. Como puede observarse en la Fig. 6, esta especificación enriquece la mostrada anteriormente (Fig. 5) añadiendo nuevos elementos y relaciones que aportan características de colaboración y *awareness* al modelado. Puesto que este nuevo modelo está basado en un lenguaje de requisitos orientado a objetivos [15], se ha añadido el objetivo *Achieve the defeat of enemies* al modelo de tareas, ya que es el principal objetivo de los usuarios que realizan la tarea *Play Game*. Además, tres recursos llamados *Speed*, *Reach* y *Power*, correspondientes a tres diferentes *power-ups* han sido incluidos en el modelo de tareas. Otra diferencia es que las tareas de interacción han sido detalladas definiendo su tipo y prioridad. Como ejemplo, la tarea *Communicate* es considerada una *Communication Task*, ya que permite a los usuarios de un mismo equipo comunicarse entre ellos. Las tareas *Listen* y *Pull Trigger* son consideradas *Coordination Tasks*. Estas tareas de coordinación son un tipo especial de tareas colaborativas en las que distintas informaciones (ruido y salud en nuestro caso) permiten coordinarse a varios jugadores. Además, la tarea *Revive Player*, es una *Collaborative Task* ya que dos diferentes personajes tienen que aproximarse a un tercero que ha muerto para poder revivirlo. Finalmente, otras tareas como *Look At Game Scenario* o *Walk* son tareas individuales que pueden ser realizadas por un solo usuario. Estas dos tareas han sido asignadas con la prioridad más alta (color rojo) ya que deben ser implementadas en primer lugar para obtener un prototipo de la aplicación lo antes posible. Por esta misma razón, *Aim* también ha sido considerada una tarea de alta prioridad, mientras que *Listen* y *Deploy Power-Up* han sido consideradas con una prioridad normal porque en nuestro primer

prototipo no es absolutamente necesario tener el sonido o el sistema de *power-ups* totalmente desarrollado.

Como se muestra en la Fig. 6, la especificación Post-WIMP no tiene tareas de sistema. Cuando una tarea es realizada por el sistema, se indicará con la participación del Role “System” en dicha tarea (por ejemplo la tarea *Deploy Power-Up* es realizada por el sistema). Además, otros cuatro roles toman parte en esta especificación: *Player*, *Allies*, *Enemies* y *Human*, que es el “super-role” de los tres primeros. Sin embargo, hay que señalar que la asociación del usuario con los roles que puede jugar se especifica en el modelo de usuario, no en el modelo de tareas, que no se presenta aquí por problemas de espacio.

Los *Participation Links* son usados para especificar qué usuarios están involucrados en tareas de interacción. Por ejemplo, dos diferentes *Participation Links* están relacionados con la tarea *Communicate*, porque cuando se realiza esta tarea un jugador (1) habla con uno o más (1..*) de sus aliados.

Finalmente, cinco diferentes recursos de *awareness* se han adjuntado a los *Participation Links* para representar la información de *awareness* requerida para realizar las tareas. Primero, el jugador debe ([M]) tener el recurso de *Awareness Allies location* para proporcionarle la localización de sus aliados a la hora de realizar la tarea *Communicate*, ya que la comunicación sólo puede llevarse a cabo si los miembros del equipo están cerca. Adicionalmente, sería deseable ([N]) tener información sobre dónde miran otros jugadores para coordinar ataques conjuntos. Además, cuando se realiza la tarea *Look At Game Scenario*, los jugadores necesitan información de *Awareness* sobre su propio estatus, algo que es altamente deseable ([HD]), para ser conscientes de sus propias habilidades (*power-ups*). Finalmente, es necesario que cada usuario sepa a qué grupo pertenece así como los miembros que posee ese grupo. Por último, es deseable ([D]) saber quiénes son los miembros de los equipos. Como se ha mostrado, la especificación Post-WIMP permite especificar la colaboración y el *awareness* de forma adecuada, siendo estos dos aspectos primordiales para el desarrollo de sistemas Post-WIMP contemporáneos. Esta especificación de la información de *awareness* solventa los problemas especificados en la sub-sección anterior.

5. CONCLUSIONES Y TRABAJOS FUTUROS

A lo largo de los últimos años, no sólo la apariencia visual de las interfaces de usuario ha cambiado, sino también la forma en la que las personas interactúan con los ordenadores. De hecho, las clásicas interfaces WIMP están dando paso a nuevos sistemas Post-WIMP que van más allá de ventanas, iconos, ratón y punteros. La interacción con estos nuevos sistemas está basada en reconocimiento de gestos, realidad virtual, dispositivos corporales, etc. Debido a la evolución de los sistemas de interacción, los mecanismos necesarios para especificarlos deben hacerlo también. Este hecho supone la principal motivación para la realización de este trabajo, que propone el desarrollo de un modelo de tareas alternativo a la actual revisión del modelo del W3C, para adecuar el modelado de tareas a la especificación de requisitos de sistemas Post-WIMP. La colaboración es uno de los aspectos clave en dichos sistemas, por lo que nuestra propuesta se basa en el modelo de tareas Collaborative Systems Requirements Modelling Language (CSRML) [5], que implementa el ampliamente aceptado modelo de colaboración 3C [10]. Con el objetivo de hacer que este modelo tenga la capacidad de especificar las necesidades de *awareness* de sistemas Post-WIMP

identificadas por Jacob [9], el modelo de *awareness* de CSRML, llamado Gutwin’s Workspace Awareness [14] ha sido enriquecido con características especiales como *Context Awareness* o *Social Awareness*. Además, se ha presentado un ejemplo de modelado en el que se ha modelado un juego Post-WIMP tanto con la actual propuesta del W3C como con el lenguaje presentado en este trabajo. Esto nos ha permitido ilustrar que las características Post-WIMP relacionadas con el *awareness* y la colaboración sólo pueden ser especificadas con éste último.

A pesar de que este modelo ha mostrado ser más adecuado para sistemas Post-WIMP, es necesario profundizar más en este sentido. Con el objetivo de mejorar la especificación de requisitos de sistemas Post-WIMP, nuestros siguientes trabajos se centrarán en un modelo de contexto que complementa el modelo de tareas. Así, mediante el uso de dicho modelo, será posible especificar cómo las aplicaciones Post-WIMP pueden adaptarse al contexto de uso (es decir, hardware disponible, capacidades del usuario, características del entorno físico como luz y ruido, etc.). Al mismo tiempo, la implementación del caso de uso presentado en este trabajo ya ha comenzado. Ya está disponible un prototipo jugable basado en la especificación de tareas creada mediante la propuesta de meta-modelo de tareas post-WIMP recogida en este artículo.

6. AGRADECIMIENTOS

Este trabajo está financiado por el Ministerio de Economía y Competencia de España y por la fundación FEDER de la UE bajo el proyecto insPIre (TIN2012-34003). También ha sido financiado por el Ministerio de Educación, Cultura y Deporte de España gracias a la becas FPU (AP2010-0259).

7. REFERENCIAS

- [1] A. van Dam, “Post-WIMP User Interfaces,” *Commun. ACM*, vol. 40, no. 2, pp. 63–67, 1997.
- [2] M. Gea and F. L. Gutiérrez, *El Diseño*. Capítulo 5 del libro electrónico de la Interacción Persona-Ordenador. Asociación AIPO, 2002.
- [3] “W3C: MBUI - Task Models.” [Online]. Available: <http://www.w3.org/TR/task-models/>.
- [4] P. González, F. Montero, V. López, A. Fernández-Caballero, J. Montañés, and T. Sánchez, “A Virtual Learning Environment for Short Age Children,” in *IEEE International Conference on Advanced Learning Technologies (ICALT’01)*, 2001, pp. 283–284.
- [5] M. A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, and P. González, “CSRML: A goal-oriented approach to model requirements for collaborative systems,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6998 LNCS, pp. 33–46.
- [6] M. A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, J. Jaen, and P. González, “Analyzing the understandability of Requirements Engineering languages for CSCW systems: A family of experiments,” *Inf. Softw. Technol.*, vol. 54, no. 11, pp. 1215–1228, 2012.
- [7] M. A. Teruel, R. Tardío, E. Navarro, A. Maté, P. González, J. Trujillo, and R. Terol Muñoz, “CSRML4BI: A Goal-Oriented Requirements Approach for Collaborative Business Intelligence,” in *33rd International Conference on Conceptual Modeling (ER’14)*, 2014, pp. 423–430.

- [8] F. Paterno, "Model-based design of interactive applications," *Intelligence*, vol. 11, no. 4, pp. 26–38, 2000.
- [9] R. Jacob, R. Jacob, A. Girouard, A. Girouard, L. Hirshfield, L. Hirshfield, MS, MS, 2008, and 2008, "Reality-based interaction: a framework for post-WIMP interfaces," in portal.acm.org, 2008, pp. 201–210.
- [10] I. Steinmacher, A. P. Chaves, and M. A. Gerosa, "Awareness support in global software development: A systematic review based on the 3C collaboration model," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6257 LNCS, pp. 185–201.
- [11] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: some issues and experiences," *Communications of the ACM*, vol. 34, no. 1, pp. 39–58, 1991.
- [12] A. García and J. Molina, "Enhancing collaborative manipulation through the use of feedback and awareness in CVEs," *Proc. 7th ACM SIGGRAPH Int. Conf. Virtual-Reality Contin. Its Appl. Ind. ACM.*, p. 32, 2008.
- [13] M. A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, and P. González, "A CSCW Requirements Engineering CASE Tool: Development and usability evaluation," *Inf. Softw. Technol.*, vol. 56, no. 8, pp. 922–949, 2014.
- [14] C. Gutwin and S. Greenberg, "A descriptive framework of workspace awareness for real-time groupware," *Comput. Support. Coop. Work*, vol. 11, no. 3–4, pp. 411–446, 2002.
- [15] M. A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, and P. González, "Comparing Goal-Oriented Approaches to Model Requirements for CSCW," in *Evaluation of Novel Approaches to Software Engineering*, L. A. Maciaszek and K. Zhang, Eds. Springer-Verlag, 2012, pp. 169–184.