

## TABLA DE CONTENIDO

<b><u>INTRODUCCION .....</u></b>	<b><u>- 1 -</u></b>
<b>1. META-MODELO DE AGENTE.....</b>	<b>- 1 -</b>
1.1 CONTROL DEL AGENTE .....	- 2 -
1.2 EL ESTADO MENTAL .....	- 7 -
1.3 EJEMPLO: ASISTENTE PERSONAL .....	- 11 -
1.4 INTEGRACIÓN CON OTROS META-MODELOS .....	- 14 -
<b>2. META-MODELO DE OBJETIVOS Y TAREAS.....</b>	<b>- 16 -</b>
2.1 EJEMPLO: UN AGENTE PLANIFICADOR DE TAREAS .....	- 21 -
2.2 INTEGRACIÓN CON OTROS META-MODELOS .....	- 25 -
<b>3. META-MODELO DE ORGANIZACIÓN.....</b>	<b>- 27 -</b>
3.1 DESCRIPCIÓN ESTRUCTURAL DE LA ORGANIZACIÓN .....	- 27 -
3.2 DESCRIPCIÓN FUNCIONAL.....	- 29 -
3.3 DESCRIPCIÓN SOCIAL .....	- 31 -
3.4 EJEMPLOS DE MODELO DE ORGANIZACIÓN:.....	- 34 -
3.5 INTEGRACIÓN CON OTROS META-MODELOS .....	- 40 -
<b>4. META-MODELO DE INTERACCIÓN .....</b>	<b>- 41 -</b>
4.1 EJEMPLO: AGENTE DE BOLSA .....	- 48 -
4.2 INTEGRACIÓN CON OTROS META-MODELOS .....	- 54 -
<b>5. META-MODELO DE ENTORNO .....</b>	<b>- 55 -</b>
5.1 EJEMPLOS.....	- 58 -
5.2 INTEGRACIÓN CON OTROS META-MODELOS .....	- 60 -
<b><u>REFERENCIAS.....</u></b>	<b><u>- 62 -</u></b>

**INDICE DE FIGURAS**

Figura 1. Entidades principales en el meta-modelo de agente .....	- 1 -
Figura 2. Conceptos relevantes en el control del agente.....	- 2 -
Figura 3. Representación de agentes en ejecución .....	- 4 -
Figura 4. Posibles expresiones para designar un agente en ejecución .....	- 5 -
Figura 5. Descripción de un agente de bolsa en ejecución.....	- 6 -
Figura 6. Meta-modelo de entidades mentales válidas .....	- 7 -
Figura 7. Asociación de condiciones al estado mental .....	- 7 -
Figura 8. Ciclo de vida de un objetivo .....	- 8 -
Figura 9. Ejemplo de estado mental .....	- 9 -
Figura 10. Ejemplo de utilización de eventos y hechos.....	- 9 -
Figura 11. Atributos genéricos para la gestión de entidades mentales .....	- 10 -
Figura 12. Objetivos del asistente personal (modelo de agente) .....	- 11 -
Figura 13. Percepción del agente utilizando el procesador de texto.....	- 12 -
Figura 14. El agente conectado al procesador de texto recibe un evento.....	- 12 -
Figura 15. Creencias .....	- 13 -
Figura 16. Descripción del agente colaborador (modelo de agente) .....	- 14 -
Figura 17. Dependencias del meta-modelo de agente respecto de otros meta-modelos .....	- 15 -
Figura 18. Relaciones entre objetivos y tareas.....	- 17 -
Figura 19. Formas en que una tarea afecta a un objetivo.....	- 18 -
Figura 20. Relación de herencia entre hechos.....	- 19 -
Figura 21. Meta-modelo de objetivos y tareas. Descomposición de tareas y objetivos.....	- 19 -
Figura 22. Meta-modelo de tareas y objetivos. Descripción de tareas .....	- 20 -
Figura 23. Meta-modelo de tareas y objetivos. Uso de entidades del entorno .....	- 21 -
Figura 24. Meta-modelo de tareas y objetivos. Restricciones sobre elementos consumidos .....	- 21 -
Figura 25. Modelo de agente para representar el agente planificador. ....	- 22 -
Figura 26. Modelo de tareas y objetivos para especificar las tareas A y B.....	- 23 -
Figura 27. Modelo de tareas y objetivos.....	- 23 -
Figura 28. Modelo de agente. ....	- 24 -
Figura 29. Modelo de objetivos .....	- 24 -
Figura 30. Modelo de objetivos y tareas.....	- 25 -
Figura 31. Dependencias del meta-modelo de objetivos y tareas respecto de otros meta-modelos.....	- 26 -
Figura 32. Las tres vistas de una organización .....	- 27 -
Figura 33. Meta-modelo de organización. Visión estructural .....	- 28 -
Figura 34. Meta-modelo de organización. Elementos que componen un flujo de trabajo.....	- 29 -

---

Figura 35. Meta-modelo de organización. Relación entre los elementos del flujo de trabajo .....	30 -
Figura 36. Descomposición de tareas y flujos de trabajo .....	30 -
Figura 37. Meta-modelo de organización. Asociación de tareas con sus ejecutores.....	31 -
Figura 38. Meta-modelo de organización. Relaciones sociales entre los componentes de la organización .-	32 -
Figura 39. Meta-modelo de Organización. Semántica sobre las meta-relaciones sociales.....	34 -
Figura 40. Estructura de una organización.....	35 -
Figura 41. La gestión de la comunidad refinada en flujos.....	36 -
Figura 42. Tareas asociadas al flujo de trabajo para dar de alta en una comunidad.....	37 -
Figura 43. Asociación de tareas en el flujo de trabajo para dar de alta en una comunidad .....	37 -
Figura 44. Responsabilidades asociadas a roles dentro de un flujo de trabajo .....	38 -
Figura 45. Tareas del flujo de trabajo asociadas a la interacción para dar de alta en una comunidad.....	38 -
Figura 46. Descripción de qué agentes desempeñan los roles .....	39 -
Figura 47. Meta-modelo de organización. Relaciones con otros meta-modelos.....	40 -
Figura 48. Relación entre los diferentes aspectos de la interacción .....	42 -
Figura 49. Taxonomía de métodos de coordinación entre agentes .....	43 -
Figura 50. Meta-modelo de interacción .....	44 -
Figura 51. Unidades de interacción y su relación con tareas y roles dentro de una especificación. ....	45 -
Figura 52. Patrones de estado mental asociables con las unidades de interacción.....	45 -
Figura 53. Ordenes de ejecución utilizables sobre las unidades de interacción.....	47 -
Figura 54. Relaciones entre la interacción y la organización .....	48 -
Figura 55. Definición del asistente de bolsa (modelo de agente) .....	49 -
Figura 56. Interacción para la formación de alianzas .....	50 -
Figura 57. Descripción inicial de la interacción para formar alianzas con diagramas de colaboración .....	50 -
Figura 58. Boceto de descripción GRASIA de la interacción para formar alianzas .....	51 -
Figura 59. Interacción para conseguir movimientos en bolsa .....	52 -
Figura 60. Descripción detallada de unidades de interacción.....	52 -
Figura 61. Condición mental contenida en <i>Condición inicio propuesta movimiento</i> .....	53 -
Figura 62. Estado mental requerido por un colaborador al inicio de las negociaciones.....	53 -
Figura 63. Relación del meta-modelo de interacción con otros meta-modelos .....	54 -
Figura 64. Meta-modelo de entorno. Recursos.....	56 -
Figura 65. Meta-modelo de entorno. Aplicaciones y percepción .....	57 -
Figura 66. Meta-modelo de entorno. Tareas, recursos y aplicaciones .....	58 -
Figura 67. Descripción de aplicaciones y recursos para el caso de estudio.....	58 -
Figura 68. Tarea asociada al asistente para deducir la nueva ubicación de los ficheros .....	59 -
Figura 69. Organización de agentes asistentes en la máquina del usuario .....	60 -
Figura 70. Meta-modelo de entorno. Relaciones con otros meta-modelos.....	60 -

---



## **ANEXO B. METODOLOGIA PARA EL DISEÑO DE SISTEMAS DE AGENTES INGENIAS**

### **INTRODUCCION**

La siguiente metodología ha sido tomada del proyecto desarrollado por el grupo de agentes software: ingeniería y aplicaciones (GRASIA), en el cual se hizo mejoras sobre la versión anterior a INGENIAS, la metodología MESSAGE.

En la metodología INGENIAS se propone un método de desarrollo que servirá para describir un sistema multiagente a través de un conjunto de modelos. Cada uno de estos modelos muestra una visión parcial del sistema: los agentes que lo componen, las interacciones que existen entre ellos, cómo se organizan para proporcionar la funcionalidad del sistema, qué información es relevante en el dominio y cómo es el entorno en el que se ubica el sistema a desarrollar.

Para especificar cómo tienen que ser estos modelos se definen meta-modelos. Un meta-modelo es una representación de los tipos de entidades que pueden existir en un modelo, sus relaciones y restricciones de aplicación.

Esta metodología está compuesta por los siguientes cinco meta-modelos:

- Meta modelo de Agente.
- Meta modelo de objetivos y tareas.
- Meta modelo de organización.
- Meta modelo de interacción.
- Meta modelo de entorno.

Además de la metodología, GRASIA proporciona, un entorno de desarrollo para permitir al diseñador especificar gráficamente cada uno de los meta-modelos correspondientes al sistema que pretenda modelar. En las siguientes secciones se describirá cada meta-modelo, acompañado de ejemplos que servirán de guía al diseñador.

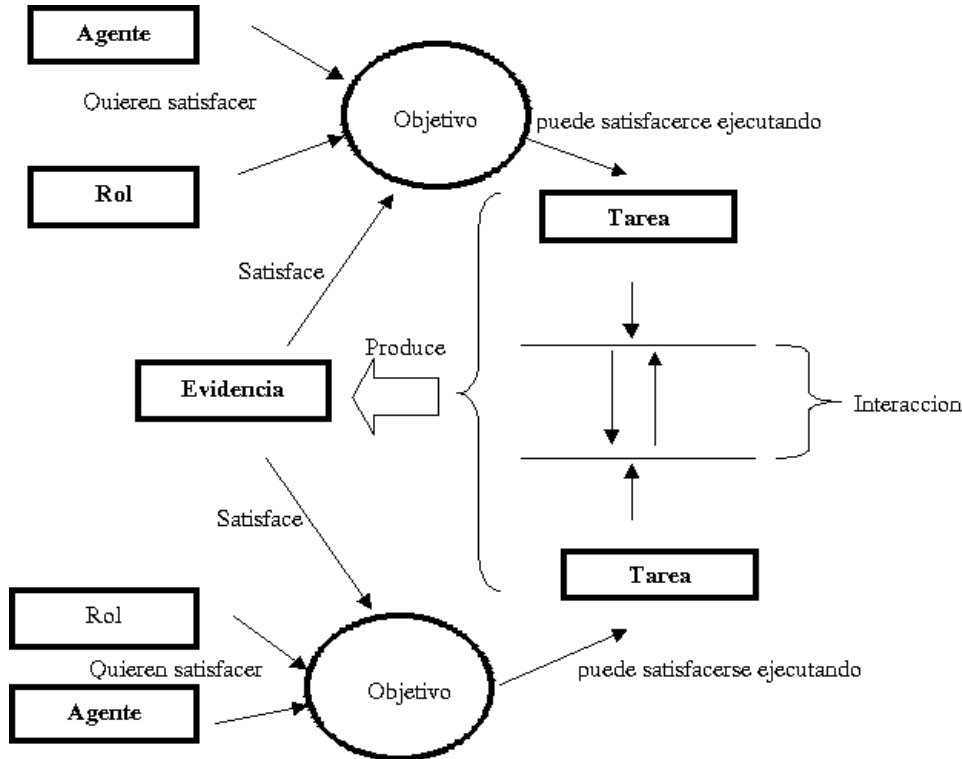




La segunda explica cómo se definen las responsabilidades del agente. Para terminar, la tercera sección se centra en la definición del estado mental. Detalla la forma en que se representa el mundo utilizando entidades mentales y cómo explicitar los estados mentales intermedios que alcanzan agentes en ejecución.

### 1.1 Control del agente

El meta-modelo propuesto conlleva un paradigma de control (Figura 2) basado en objetivos y tareas. El agente juega cierto rol, el cual le lleva a perseguir unos objetivos. Estos objetivos se alcanzan cuando el mundo, tal como lo concibe el agente, se ha modificado. Las tareas son las encargadas de producir esta modificación. La modificación se expresa mediante la producción de evidencias, que pueden ser nuevas entidades mentales o eventos disparados por el entorno. Este esquema se incluye en el meta-modelo a través del estado mental, el gestor de estado mental y el procesador de estado mental.



**Figura 2. Conceptos relevantes en el control del agente**

El estado mental puede verse como toda aquella información que permite al agente tomar decisiones. Esta información es gestionada y procesada para producir las decisiones del agente. En esta metodología se han incluido a nivel conceptual dos entidades, el *Gestor Del Estado Mental* y el *Procesador del Estado Mental*. El propósito del primero es desarrollar la evolución del estado mental mediante las operaciones de creación, destrucción, modificación y monitorización del conocimiento del agente. Es responsable de mantener la



coherencia del conocimiento almacenado y de hacerlo evolucionar. El propósito del segundo es la toma de decisiones y el control del agente.

La ventaja de esta separación entre gestor y procesador del estado mental es que se desacoplan los mecanismos que implementan la autonomía e inteligencia del agente de su conceptualización. El diseñador puede decidir cualitativamente cómo va a ser el control, establecer qué debe incluir antes de implementarlo, expresarlo en función de manipulaciones del estado mental y ver cómo afecta a su desarrollo.

Para definir los requisitos que deben satisfacer el gestor y el procesador se utilizan tres herramientas: relaciones entre las tareas y entidades mentales, relaciones entre el agente y objetivos y especificación de estados intermedios por los que pasa un agente en ejecución.

Las relaciones que asocian tareas y entidades mentales son instancias de la meta-relación *GTAfecta*. Las meta-relaciones *GTAfecta* tienen como propósito poner de manifiesto que la ejecución de una tarea afecta al estado mental del agente, satisfaciendo un objetivo o haciéndolo fallar.

Las relaciones entre los objetivos y el agente, aparte de las heredadas de *Entidad Autónoma* (meta-relación *GTPersigue*) y de las obtenidas a través del Estado Mental, se vuelven a recalcar con la relación *WFPersigue*. Esta meta-relación coloca al objetivo y al agente en el contexto de la organización. La relación *WFPersigue* es parte de la definición del flujo de trabajo y se usa para justificar por qué se ejecutan las tareas en el seno de una organización.

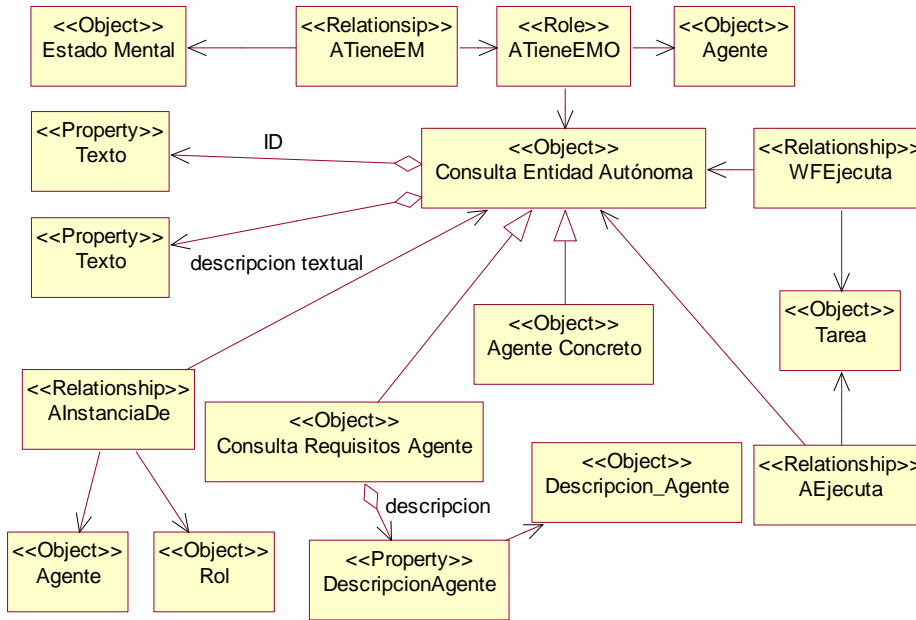
Por último, en la asociación de estados intermedios a agentes en ejecución, que se verá con detalle en la presentación del estado mental, hay que determinar cómo se referencian instancias de los agentes a nivel de meta-modelo. Se habla de referenciar instancias de entidades del modelo en lugar de utilizar directamente las instancias porque de otra forma se cometería una violación de los niveles de meta-modelado. Este tipo de solución también se aplica en los diagramas de secuencia y colaboración de UML donde se asigna un tipo y un identificador de variable para representar un objeto.

### **1.1.1 Agentes en ejecución**

En el meta-modelo de agente es necesario, para describir la evolución del estado mental hacer referencia a propiedades del agente en ejecución. Con este propósito se introduce en la entidad *Consulta Entidad Autónoma*. Esta entidad sirve también para describir quién participa en las interacciones, jugando los roles correspondientes.

*Consulta Entidad Autónoma* se especializa en consultas acerca de las propiedades de los agentes y en agentes concretos. Las primeras contienen expresiones de restricciones que un agente debe satisfacer, como “que desempeñe el papel de administrador de fondos de pensiones”. Las segundas contienen expresiones que

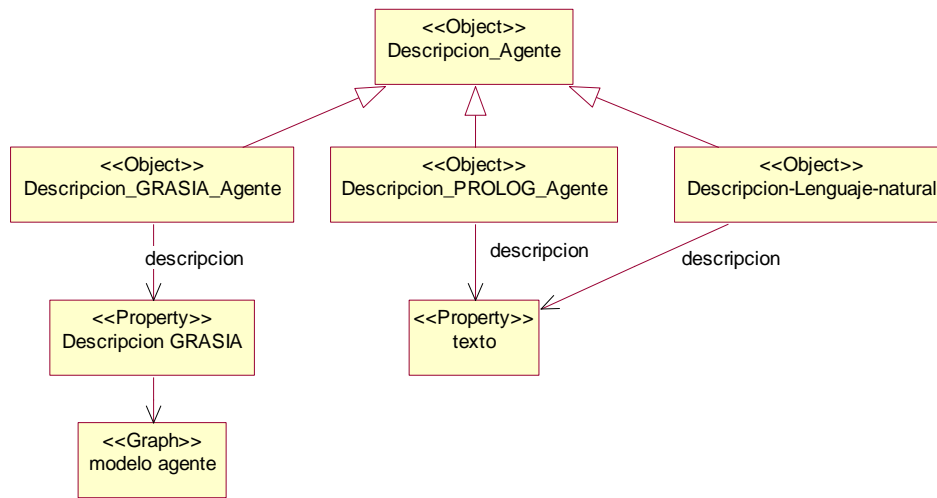
denotan agentes concretos, como `librero@iiop://librería.com:1099/acc1` que denota un agente único en el sistema.



**Figura 3. Representación de agentes en ejecución**

Para poder representar completamente un agente en ejecución, por cada asociación en que participe la entidad agente, debe poder participar *Consulta Entidad Autónoma* (ver Figura 3). Se permite indicar si el agente en ejecución ha de ser de un tipo concreto (meta-relación *AInstanciaDe*). Esta restricción puede ser útil para hacer referencia a cualquier agente o rol de un tipo concreto. Para hacer referencia a otras cualidades del agente, se utiliza la propiedad *DescripciónAgente* (ver Figura 4).

<sup>1</sup> Localizador universal según el estándar FIPA.



**Figura 4. Posibles expresiones para designar un agente en ejecución**

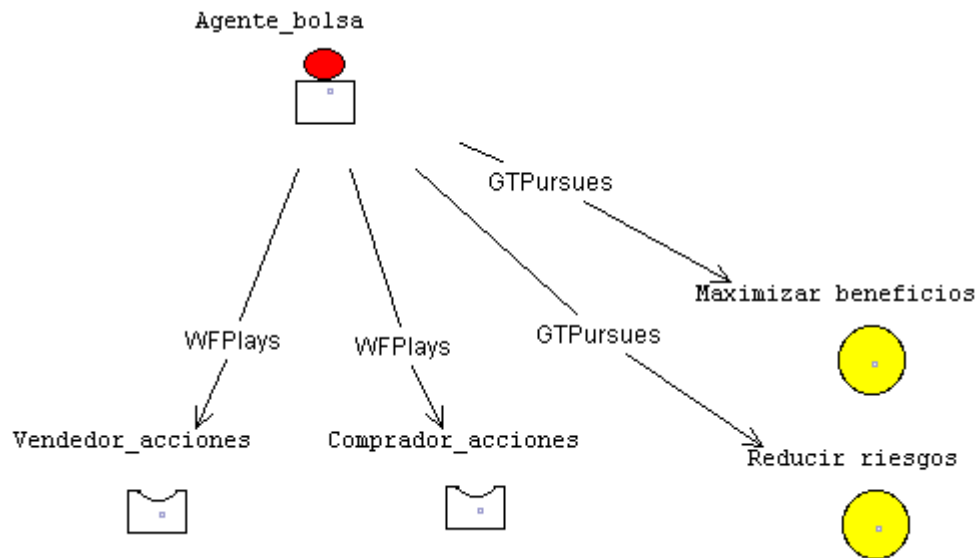
La descripción del agente se puede hacer utilizando sentencias PROLOG, GRASIA o lenguaje natural.

Los términos *Consulta Entidad Autónoma* se representan con el símbolo de un agente encerrado entre paréntesis. Este símbolo se etiqueta de diferente manera dependiendo de a qué se haga referencia:

- **Un agente concreto (Agente Concreto).** En este caso la etiqueta es el nombre del agente o bien una expresión que denota un único agente. Como caso especial, se tiene la etiqueta “(Ejecutor)”. Aparece al referenciar un agente que va a ejecutar una tarea, al definir un estado mental intermedio de un agente en una interacción, o al estudiar la satisfactibilidad de un objetivo. El ejecutor está unido a la tarea ejecutada con una instancia de la meta-relación *WFEjecuta* o *AEjecuta*.

**NOTA:** En las figuras de este documento, los nombres de estas relaciones pueden aparecer en Inglés ya que para realizarlas se utilizó la herramienta para el modelamiento de sistemas multiagente INGENIAS, proporcionada por GRASIA (Grupo de Agentes Software Ingeniería y Aplicaciones). Por ejemplo, *WFJuega* y *WFPlays* tienen el mismo significado.

- **Un conjunto de agentes (Consulta Requisitos Agente).** Otra vez la etiqueta define un conjunto de agentes. Se utiliza una expresión que denota las propiedades que se requieren de los agentes. Estas expresiones utilizan como términos, predicados equivalentes a las relaciones representadas en los meta-modelos. Como indica la Figura 4, se contemplan expresiones en PROLOG (para incluir expresiones en lógica proposicional), lenguaje natural y descripciones con modelos de agente. Estas últimas tienen la ventaja de poder hacer referencia directa a elementos de otros modelos. Un ejemplo de descripción de agente es el de la Figura 5.



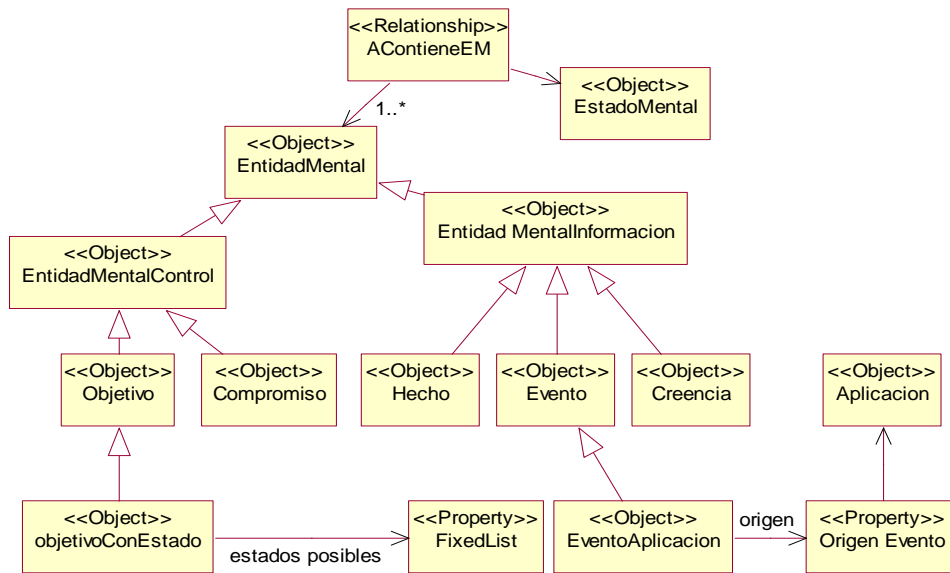
**Figura 5. Descripción de un agente de bolsa en ejecución.**

En el ejemplo de la figura 5 describe un agente que es capaz de vender y comprar acciones y que persigue maximizar beneficios y reducir riesgos. Para representar esta situación, se asocia un agente en ejecución con los objetivos pertinentes y con los roles identificados en el problema. Los objetivos se representan con un círculo, los roles con un cuadrado con una concavidad en la parte superior y el agente es un cuadrado con un círculo en la parte superior.

### 1.1.2 Asociación de responsabilidades

El meta-modelo contempla el uso de roles para asignar responsabilidades. El papel del concepto de rol es separar la definición del agente de lo que se requiere de él. El rol del que aquí se habla es distinto del de UML. En general, el rol es una abstracción de un conjunto de funciones, que puede tener estado y que para existir necesita de otra entidad no abstracta que lo desempeñe, en este caso el agente. De esta forma es posible aplicar los roles para describir conjuntos de responsabilidades, protocolos, o permisos de actuación (cambiando los roles de un agente, se cambia su capacidad de actuación).

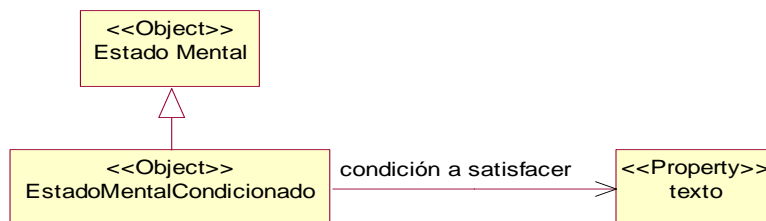
Los roles aparecen en las interacciones y en las organizaciones. Los roles carecen de gestores y procesadores de estado mental, y, por lo tanto, de capacidad de toma de decisiones. Por su simpleza, es utilizado con frecuencia en lugar de los agentes, ya que hace más tratable el diseño de algunos aspectos del sistema, como las interacciones. Los roles también fomentan la reusabilidad, ya que existen para otorgar al que los juega sus cualidades. La relación entre roles y agentes se hace mediante la meta-relación WFJuega. Esta meta-relación indica al diseñador que el agente adquiere todos los objetivos asociados al rol, todas sus responsabilidades y sus capacidades.



**Figura 6. Meta-modelo de entidades mentales válidas**

### 1.2 El estado mental

El estado mental se define como agregación de entidades mentales. Se admiten múltiples instancias del estado mental relacionadas con un agente. Al asociarlo directamente, se indica que el agente, al crearse, parte de ese estado. Cuando se asocia indirectamente, como en la Figura 9, se está definiendo un estado mental intermedio en la vida del agente. Para expresar estas situaciones, se admite que existen instancias del meta-modelo de agente que representan momentos concretos en la vida del agente. Para estos estados intermedios, se emplean entidades *Consulta de Entidad Autónoma*, ya introducidas en las secciones anteriores.

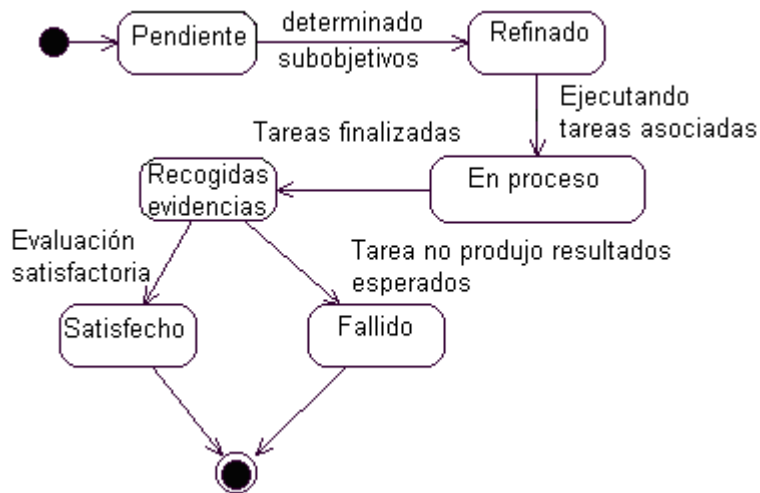


**Figura 7. Asociación de condiciones al estado mental**

En estas situaciones intermedias, a veces es importante añadir información acerca de qué se espera de las entidades mentales. Por ejemplo, si sus propiedades deben contener algún valor concreto o si se debe satisfacer cierta expresión. Para ello, se define un estado mental condicionado (Figura 7) que se caracteriza por contener expresiones que referencian las entidades mentales que existen en un momento concreto. Las entidades mentales identifican los elementos en los que se basa el agente a la hora de tomar decisiones.

En el estado mental (ver Figura 6), la información se representa a alto nivel. Siguiendo las tendencias existentes, el estado mental se define a partir de entidades como objetivos (fundamental en este modelo), creencias y compromisos. Adicionalmente, se plantean hechos (información cierta para el agente) y eventos (los cambios ocurridos, en el mundo que el agente capta). Los eventos deberían transformarse en hechos automáticamente en cuanto formaran parte del estado mental del agente. Sin embargo, en este modelo conviene más el mantenerlos así y dejar que sean los procesos internos del agente los que decidan si debe realizarse tal transformación o simplemente deshacerse del evento.

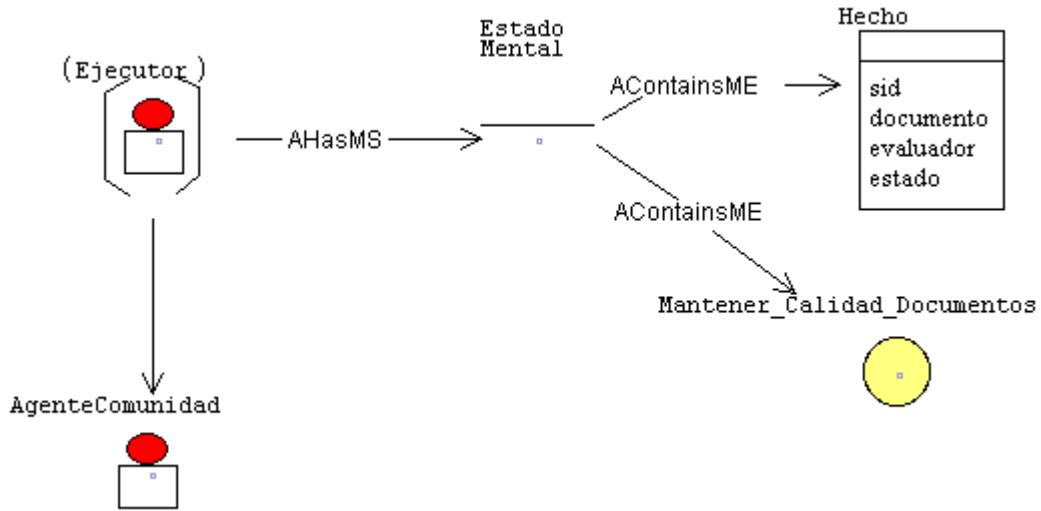
Las entidades de control son el objetivo, porque indican al agente qué es lo que le motiva, y el compromiso, porque liga al agente con una acción requerida por otro actor. Los objetivos tienen asociado un estado mediante la propiedad de estados posibles, que es una lista de constantes que denotan los estados por los que pasa el agente. Los objetivos se satisfacen con la presencia de ciertas evidencias y pueden darse por fracasados cuando estas evidencias no se han producido. Hasta llegar a un objetivo satisfecho o fracasado, se pueden recorrer varios caminos. Aquí se propone que un objetivo pueda estar pendiente, recogiendo evidencias, refinado, en proceso, satisfecho o fallido. El ciclo de vida hace referencia a cómo se ejecutan tareas asociadas al objetivo (las que producen las evidencias que necesita). Las tareas no tienen estado en esta metodología, por lo que, en caso de necesitar estructurar su ejecución, las tareas tienen que delegar su estado a otra entidad, el objetivo en este caso. La elección de este ciclo de vida no es obligatoria.



**Figura 8. Ciclo de vida de un objetivo**

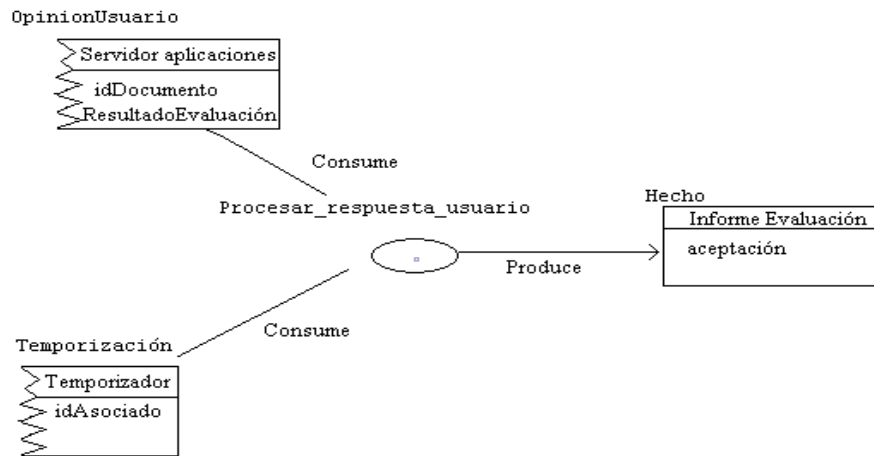
El resto de entidades que definen el estado mental son entidades informativas que el agente utiliza para formarse una imagen del mundo que le rodea. Los hechos reflejan información que es inherentemente cierta, como que “el agua se evapora al aplicarle calor” o bien información resultante de la ejecución de tareas. Las creencias son interpretaciones particulares hechas por el agente de lo que percibe. Así ante el evento

“comunicación con el agente X interrumpida” el agente puede elaborar la creencia “no quiere comunicarse conmigo” o bien “ha ocurrido un fallo en la red” o bien “ha ocurrido un fallo en el ordenador que alojaba al agente X”.



**Figura 9. Ejemplo de estado mental requerido por un agente que decide si debe ejecutar una solicitud de alta en una fuente de información.**

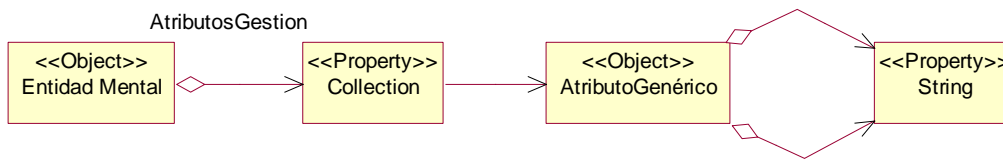
Como ejemplo, la Figura 9 corresponde al estado mental (representado con la línea horizontal) que debe tener un agente de comunidad que quiera procesar una petición de inscripción en la comunidad. El diagrama dice que el agente persigue el objetivo *Mantener calidad documentos* y que además existe una sesión de evaluación en curso. El elemento ejecutor identifica al agente que ejecuta la tarea y la asociación con el agente indica que el ejecutor es del tipo *Agente de Comunidad*.



**Figura 10. Ejemplo de utilización de eventos y hechos**

La Figura 10 muestra una tarea cuyo propósito es generar un informe de evaluación acerca de la conveniencia de aceptar un nuevo usuario en una comunidad. La tarea procesar respuesta usuario (representada con un óvalo) puede arrancar porque el usuario haya evaluado la petición de suscripción (evento *OpiniónUsuario*) o porque haya saltado un temporizador por el retraso del usuario en ejecutar la evaluación (evento *Temporización*). De cualquier forma, estos dos eventos se transforman en un nuevo hecho (*Informe de Evaluación*). Obsérvese la forma en la que se representan los hechos y eventos.

El estado mental es extensible para expresar entidades mentales más elaboradas. Por ejemplo, se plantean las dos extensiones de la Figura 11 que permiten incluir creencias acerca del estado mental de otros agentes o de sus capacidades.



**Figura 11. Atributos genéricos para la gestión de entidades mentales**

Es deseable asociar atributos a las entidades mentales para propósitos de gestión del conocimiento (Figura 11). Ejemplos de tales atributos son los temporales que especifican la fecha de creación de la entidad mental y un estimativo de fecha de terminación. La naturaleza y propósito de estos atributos de gestión es variada. Los compromisos, por ejemplo, pueden ser válidos sólo hasta que se sobrepase una fecha, los eventos sólo durante unos segundos. La elección de los atributos de gestión comunes a todas las entidades mentales es función del tipo de gestión que se vaya a realizar. Por ello, se ha añadido en el meta-modelo sólo la asociación de atributos genéricos.

### 1.2.1 Justificación de la definición de estado mental

La representación del estado mental tiene como fin permitir al diseñador definir diferentes situaciones en la configuración del estado mental. Una buena representación de conocimiento debe reunir las siguientes propiedades:

**Suficiencia de la representación.** La capacidad de representar todos los tipos de conocimiento necesarios en el dominio.

**Suficiencia deductiva.** La capacidad para manipular las estructuras de la representación con el fin de obtener nuevas estructuras que correspondan a un nuevo conocimiento deducido a partir del antiguo.

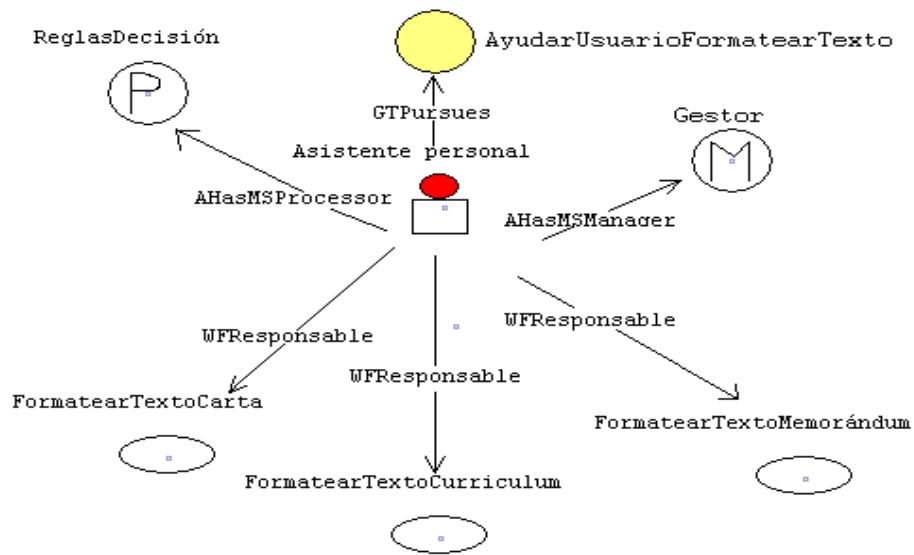
**Eficiencia deductiva.** La capacidad de incorporar información adicional en las estructuras de conocimiento con el fin de que los mecanismos de inferencia puedan seguir las direcciones más prometedoras.



**Eficiencia en la adquisición.** La capacidad de adquirir nueva información con facilidad. El caso más simple es aquél en el que una persona inserta directamente el conocimiento en la base de datos. Idealmente, el programa sería capaz de controlar la adquisición de conocimiento por sí mismo.

### 1.3 Ejemplo: asistente personal

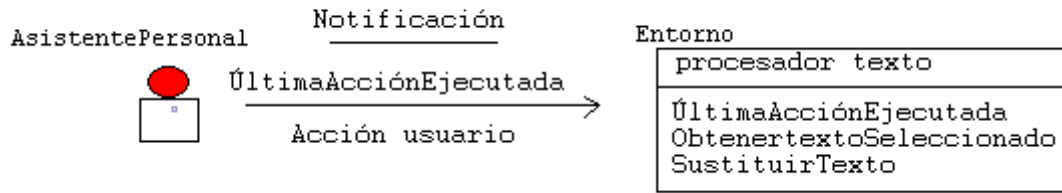
Se quiere modelar un agente de interfaz que asista al usuario en el manejo de un procesador de texto. El agente intercepta los eventos enviados por el usuario a la aplicación de la interfaz gráfica y utiliza esta información para asesorar al usuario acerca de secuencias de acciones complementarias que pueden mejorar los resultados obtenidos. Este ejemplo se centra en acciones relacionadas con el formateo de texto.



**Figura 12. Objetivos del asistente personal (modelo de agente)**

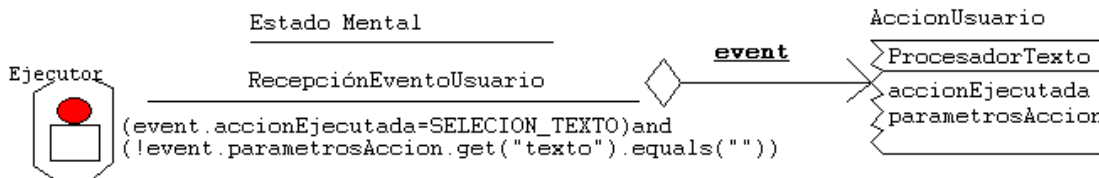
La Figura 12 muestra que el agente, *Asistente Personal*, tiene como objetivo el ayudar al usuario en labores de formateado de texto. Inicialmente, el asistente sabe cómo convertir el texto seleccionado en una carta (tarea *FormatearTextoCarta*), como un currículum (tarea *FormatearTextoCurrículum*) o como un memorándum (tarea *FormatearTextoMemorandum*). El estado mental del agente se expresará como un conjunto de hechos, por lo que sólo se necesitan primitivas para aceptar y eliminar hechos. El procesamiento del estado mental se hará mediante reglas de producción.

El agente percibe las acciones del usuario a través del procesador de texto. Para representar esto es necesario utilizar un modelo de entorno (Figura 13).



**Figura 13. Percepción del agente utilizando el procesador de texto**

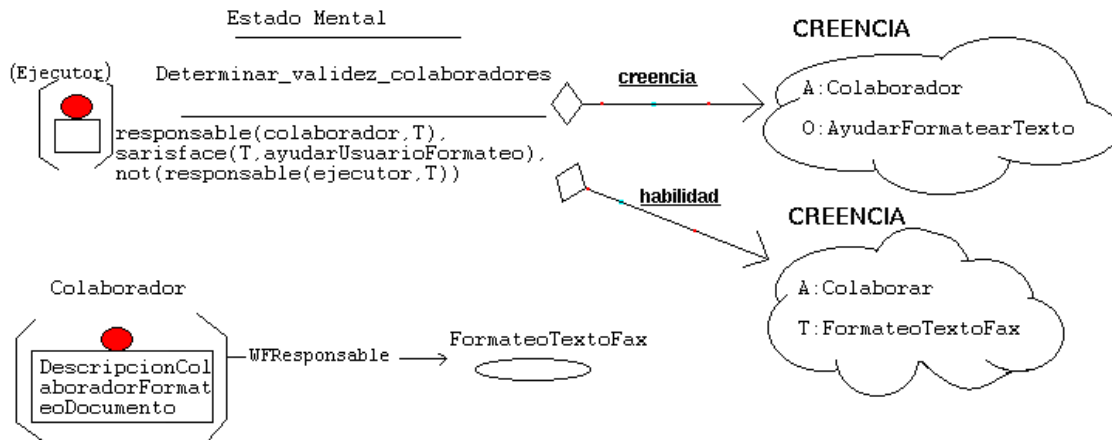
El agente necesita saber qué acciones ejecuta el usuario. Por ello se define una asociación de tipo notificación entre el Asistente Personal y el Procesador de Texto. Esta relación se interpreta como que el agente Asistente Personal debe ser notificado de cambios en el resultado de la operación *últimaAcciónEjecutada*. Cuando se produzca un cambio se creará un evento de la clase *AcciónUsuario*. En este ejemplo se supone que el usuario selecciona texto en el procesador. El procesador modifica su estado para que con la operación *ultimaAcciónEjecutada* se devuelva la acción selección de texto. La percepción por notificación se activa y crea una nueva entidad mental de tipo Evento.



**Figura 14. El agente conectado al procesador de texto recibe un evento**

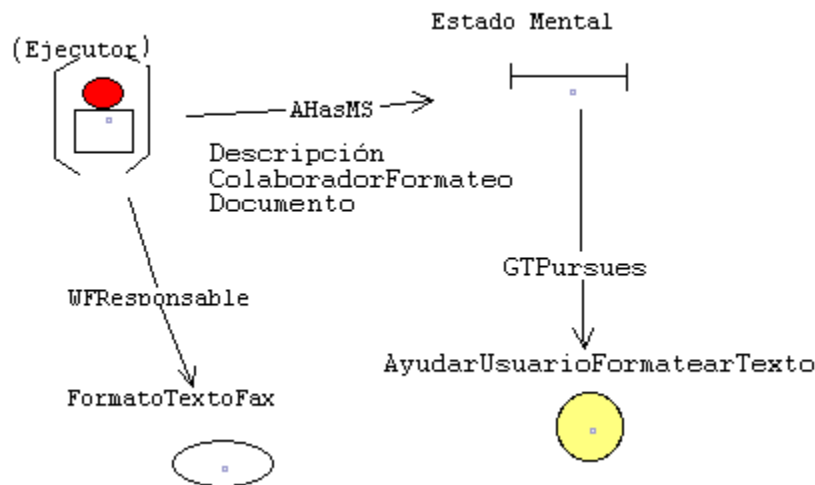
La Figura 14 expresa un estado mental significativo para el agente. Para concretar el estado mental se ha utilizado una expresión JAVA. En este estado se ha recibido un evento *AccionUsuario* cuyo slot *accionEjecutada* indica que se ha seleccionado texto en el procesador. Entre los parámetros de la acción del usuario (*parameterosAccion*) se incluye el parámetro texto, que ha de ser distinto de la cadena vacía.

Dentro de un SMA, el Asistente Personal, para mejorar la asistencia al usuario, puede contactar con otros agentes para preguntarles si saben realizar labores de formateo que él desconoce. Antes de iniciar la comunicación, se plantea el estado mental de la (Figura 15).



**Figura 15. Creencias necesarias para expresar la conveniencia de colaborar con otro agente (modelo de agente)**

El agente con el que se colabora se etiqueta colaborador en la Figura 15. Este colaborador se describe mediante una expresión etiquetada como *DescripcionColaboradorFormateoDocumento*. Las creencias mostradas en la figura indican la existencia de instancias de *WFResponsable* y *GTPersigue* entre un agente y una tarea y entre un agente y un objetivo, respectivamente. En la primera instancia, se asocia un colaborador con la tarea formateo texto fax. En la segunda se asocia un colaborador con el objetivo ayudar usuario a formatear texto. Del agente colaborador no se puede esperar un tipo concreto, por ello se describen únicamente sus cualidades: que debe perseguir el objetivo *AyudarUsuarioFormatearTexto* y que debe tener alguna tarea que la satisfaga, *FormateoTextoFax* en este caso (Figura 16). Sin embargo, hace falta algo más, esto es, que las tareas que sepa ejecutar el colaborador no sean conocidas por el Asistente Personal. Esto se expresa dentro del estado mental de la Figura 15 en forma de predicado PROLOG. Se pide que colaborador sepa ejecutar una tarea T que satisfaga *AyudarUsuarioFormateo* que no esté disponible en el asistente (`not responsable(ejecutor, T)`).



**Figura 16. Descripción del agente colaborador (modelo de agente)**

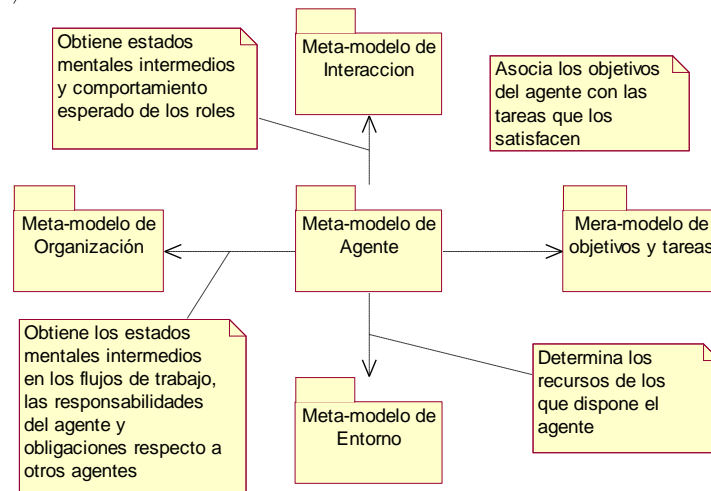
#### 1.4 Integración con otros meta-modelos

El meta-modelo de agente contiene algunos de los elementos más comunes en todos los meta-modelos, como los agentes o roles (Figura 17).

En el meta-modelo de agente se asocian agentes con los roles, que luego participan en el meta-modelo de interacción. De esta participación se extraen un conjunto de estados mentales y de acciones que debe ejecutar el agente.

El meta-modelo de organización sitúa al agente con relación a otros agentes. Además, proporciona una descripción del papel del agente en los flujos de trabajo existentes. Mediante esta descripción se explica cómo colabora el agente con otros agentes para lograr objetivos comunes.

Para terminar, el meta-modelo de entorno proporciona una descripción de las aplicaciones y recursos requeridos por el agente o por las tareas de las que es responsable. También sirve como descriptor de la percepción del agente. Como se verá más adelante, en el meta-modelo de entorno, el agente se puede conectar a aplicaciones ya existentes para obtener información del entorno. La aplicación puede ser software como un procesador de texto o ser un recubrimiento de dispositivos hardware, como un controlador de una compuerta en una canalización de agua.



**Figura 17. Dependencias del meta-modelo de agente respecto de otros meta-modelos**

Para asegurar la consistencia de un modelo de agente respecto a otros modelos de un SMA es necesario satisfacer los siguientes requisitos:

- Toda tarea asociada al agente debe aparecer en alguna instancia de meta-modelos de interacción, si implica interacción entre agentes, y en el modelo de organización, indicando su papel dentro de la estructura global de tareas.
- Para cada tarea es necesario indicar cómo afecta al estado mental del agente. La ejecución de cualquier tarea siempre conlleva alguna modificación. Esta modificación debe aparecer en los modelos de tareas y objetivos.
- El modelo del estado mental debe tener en cuenta la ejecución en paralelo de distintas tareas que modifiquen el estado mental. Es admisible que dos tareas tengan efectos contrarios sobre la misma entidad mental. Este hecho debe comprobarse en modelos de objetivos y tareas.
- Para cada agente del modelo de organización, debe existir una instancia del meta-modelo de agente y viceversa.
- Cada rol jugado por el agente debe aparecer en algún modelo de organización o en algún modelo de interacción
- Un agente no puede jugar dos roles que interactúan entre sí.

Si se tienen en cuenta las siguientes tres ideas, el agente alcanzará los objetivos fijados:

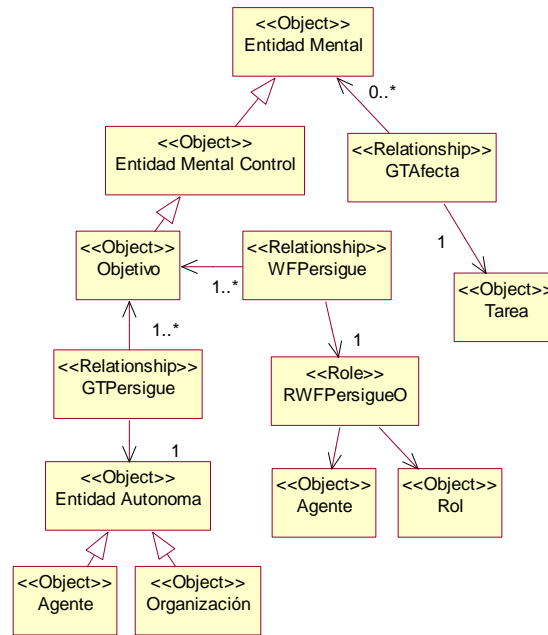
- Cuando un agente se asocie a una tarea, los objetivos que se alcancen han de favorecer la consecución de los objetivos que persiga el agente (propios y obtenidos por asociación a roles). Esta información debe estar presente en los modelos de objetivos y tareas.
- El agente siempre ejecuta acciones que le hacen alcanzar objetivos. Esta información se obtiene de este modelo.
- Cuando un agente alcanza los objetivos fijados el problema pasa a ser un tema de control y debe ser tratado como tal. Dependiendo de la situación, es posible:
  - Plantear nuevos objetivos que pueden ser réplica de los alcanzados. La creación de nuevos objetivos puede tratarse a nivel del procesador de estado mental o como resultado de la ejecución de tareas (ver meta-relación *GTCrea* en meta-modelo de tareas y objetivos). Nuevamente, esta información debe aparecer en modelos de objetivos y tareas.
  - Finalizar la ejecución. Alcanzados los objetivos fijados inicialmente, ¿existe motivo para continuar? Si no es así, quizá el agente deba dar por terminado su tiempo de vida.

## 2. META-MODELO DE OBJETIVOS Y TAREAS

El meta-modelo de objetivos y tareas tiene como propósito recoger las motivaciones del SMA, definir las acciones identificadas en los modelos de organización, o interacciones de agentes y cómo afectan estas a sus responsables. Este modelo trata de poder expresar cuáles son las razones y consecuencias de ejecutar las tareas.

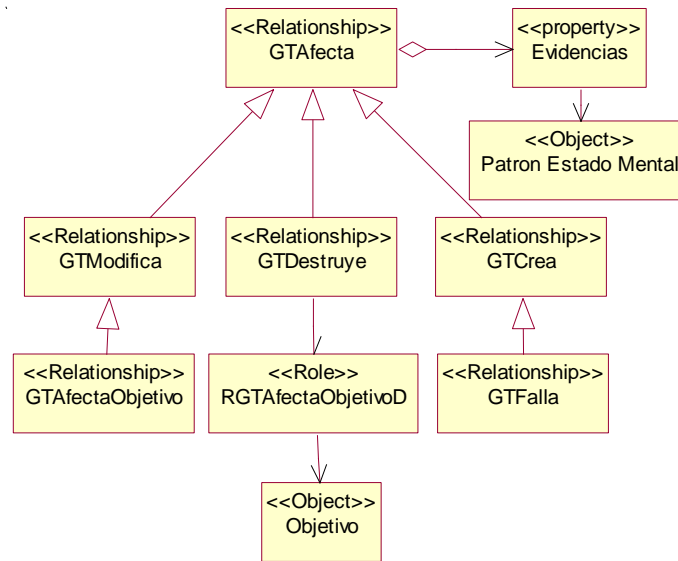
El meta-modelo de tareas y objetivos se usa para expresar la motivación que hay detrás de las tareas y qué opciones de actuación se le presentan a un agente en un momento dado. La Figura 19 muestra la parte del meta-modelo de tareas y objetivos dedicada a este fin.

En el meta-modelo se refleja la relación de los objetivos con los agentes, roles y organizaciones. Una organización, como entidad autónoma al igual que un agente persigue objetivos (*GTPersigue* o *GTPursues*). Los roles, por el contrario, se asocian con los objetivos en el marco de los flujos de trabajo, por ello se asocian con otra meta-relación, *WFPersigue*. Los agentes, por poder aparecer en los flujos de trabajo y ser una entidad autónoma, como la organización, pueden asociarse con cualquiera de las dos.



**Figura 18. Relaciones entre objetivos y tareas**

Las tareas se asocian con los objetivos mediante la meta-relación *GTAfecta* (ver Figura 19), cuya semántica es tal que la ejecución de la tarea afecta de una forma específica a una entidad mental. El ámbito de actuación de la tarea, en cuanto a *GTAfecta*, se limita a las entidades mentales del ejecutor de la tarea. La modificación del estado mental de otros agentes se modela haciendo que una tarea produzca una interacción con otro agente. Esta interacción provoca la ejecución en el otro agente de tareas que son las responsables finales de realizar el cambio.



**Figura 19. Formas en que una tarea afecta a un objetivo**

Se distinguen tres formas de alterar el estado mental: destruyendo una o varias entidades (GTDestruye), creándolas (GTCrea) o modificándolas (GTModifica). Todas ellas se ilustran con un patrón de estado mental para indicar en qué condiciones la creación, modificación o destrucción tienen lugar.

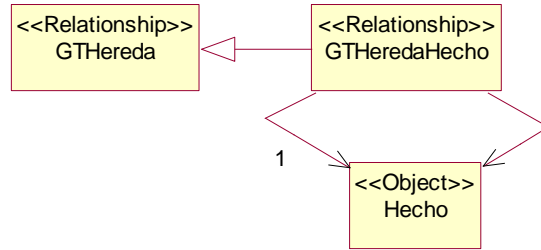
Para el caso de los objetivos, se ha especializado la relación GTModifica. Como el desarrollo que se propone se basa en la utilización de objetivos, es justificable el considerar cómo fracasan o se satisfacen. Por ello se ha dispuesto de dos especializaciones que concretan la forma de modificar un objetivo: satisfaciéndolo (GTSatisface) o haciéndolo fallar (GTFalla). Estas dos últimas se condicionan a la existencia de ciertas evidencias que indiquen cuándo tiene lugar el fallo o la satisfacción del objetivo (propiedad Evidencia). Esta condición se expresa con patrones de estado mental, ya mencionados en el capítulo anterior. Queda pendiente el decidir qué hacer cuando se alcanza un objetivo o cuando este fracasa. Aquí se consideran dos enfoques:

**Delegar en el gestor y procesador de estado mental.** De forma genérica se establece cuál es el tratamiento de los objetivos fracasados o satisfechos. Este tratamiento entraría dentro de las especificaciones del gestor y procesador de estado mental.

**Definir nuevas tareas para tratar con objetivos fracasados o satisfechos.** El cómo se realiza la gestión de entidades se detalla mediante tareas que destruyen, crean o modifican las entidades mentales existentes. La ventaja de este enfoque es que se hace explícito el tratamiento de gestión.

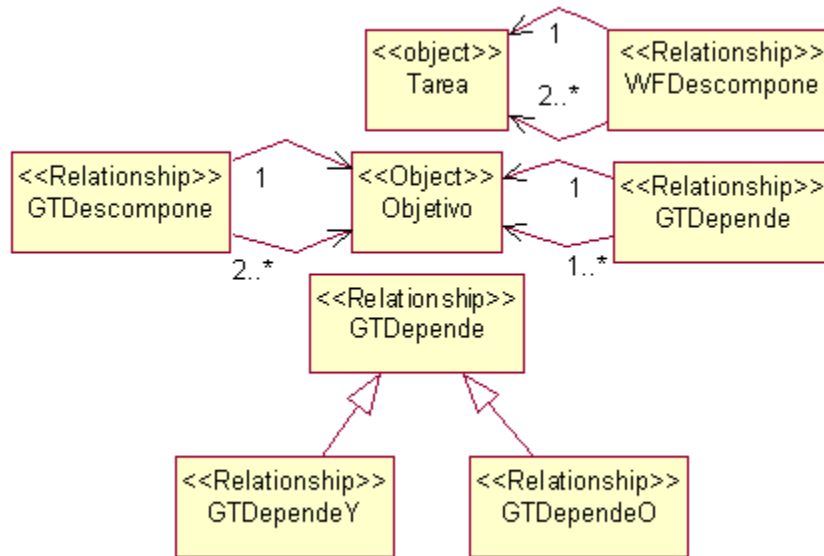


Para que este último enfoque funcione correctamente, es necesario definir relaciones de herencia entre hechos. El resto de entidades mentales también podría necesitar este tipo de relaciones, pero de momento no ha sido necesario.



**Figura 20. Relación de herencia entre hechos**

La herencia definida es del tipo simple y establece que lo que se hereda son las propiedades definidas en el ancestro.



**Figura 21. Meta-modelo de objetivos y tareas. Descomposición de tareas y objetivos.**

Debido a la complejidad que puede existir en los objetivos y las tareas, se han incorporado meta-relaciones que representan la descomposición de objetivos y tareas (*GTDescomponere* y *WFDescomponere*). Estas relaciones de descomposición dan lugar a grafos acíclicos donde se distinguen tareas, objetivos, sub-tareas y sub-objetivos. Este aspecto estructural se complementa con la expresión de dependencias entre objetivos. Nótese que no se incorporan dependencias entre tareas, aunque éstas son expresables mediante otras relaciones que se verán a continuación. La descomposición de objetivos da lugar a árboles Y/O, que están soportados por instancias de *GTDependereY* y *GTDependereO*. *GTDependereY* asume que cuando todos los sub-

objetivos (los definidos por esta relación) han sido satisfechos, el objetivo padre puede darse por alcanzado. La definición de *GTDependeO* se diferencia en que sólo se requiere que al menos uno de los objetivos referenciados haya sido satisfecho. Estas instancias también se usan para transmitir un fallo desde un sub-objetivo al objetivo padre. La transmisión de fallos desde los sub-objetivos al padre se hace en función del tipo de dependencia. *GTDependeY* hace que el objetivo padre falle si cualquiera de sus sub-objetivos falla. *GTDependeO* hace que el objetivo padre falle si todos sus sub-objetivos fallan.

Se pueden definir un bosque de descomposición de objetivos y relaciones de dependencias Y/O entre objetivos pertenecientes a distintos árboles de descomposición estructural.

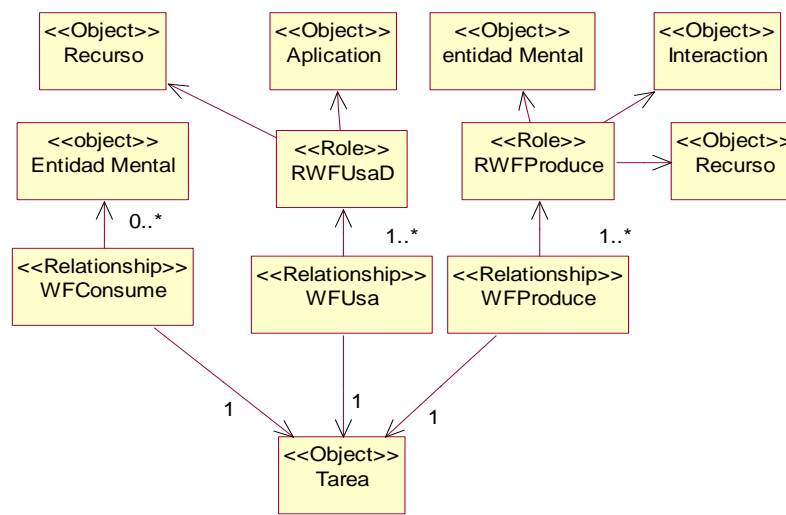
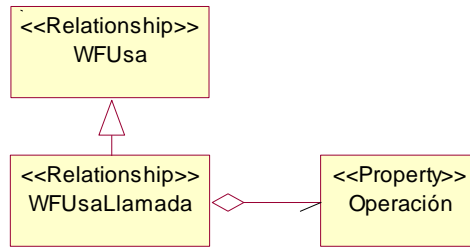


Figura 22. Meta-modelo de tareas y objetivos. Descripción de tareas

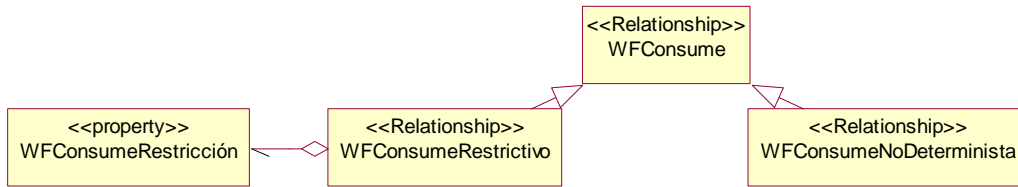
Las tareas (Figura 22) se describen mediante meta-relaciones que definen precondiciones (*WFConsume*, *WFUsa*, *GTAfecta*) y postcondiciones (*WFProduce*, *GTAfecta*). *WFConsume* indica que para que se ejecute la tarea se necesita que existan determinadas entidades mentales (objetivos, creencias, hechos, etc.). La meta-relación *WFUsa* indica que se necesitan algunos de los recursos que se identifican en el meta-modelo de entorno. *GTAfecta* indica que se actúa sobre una entidad mental, por lo que se requiere de su existencia. Las postcondiciones se representan con *WFProduce* y *GTAfecta*. *WFProduce* indica la creación de entidades mentales, interacciones o reposición de recursos en el seno de un flujo de trabajo. *GTAfecta* también puede señalar la creación de entidades mentales, pero se usa para indicar cambios en las entidades mentales no relacionadas con el flujo de trabajo en que se enmarca.

Aunque la ejecución de la tarea debe decidirla el procesador de estado mental, se asume que no se pueden ejecutar tareas que no satisfagan las precondiciones especificadas o que no dispongan de recursos o que, cuando se trate de tareas ejecutadas en el curso de una interacción, no se satisfagan las condiciones impuestas en instancias de *UIInicia* y *UIColabora*.



**Figura 23. Meta-modelo de tareas y objetivos. Uso de entidades del entorno**

Durante la ejecución de la tarea se ejecutan acciones sobre el entorno, que en este trabajo se considera como una agregación de agentes, aplicaciones y recursos. Las acciones sobre otros agentes se consideran como interacciones en el sentido de instancias del meta-modelo de interacción. Sin embargo, las acciones sobre otras aplicaciones o recursos están dentro de un orden distinto. Al principio del desarrollo, estas acciones se representan con *WFUsa* sin llegar a explicitar exactamente qué operación se está utilizando. En etapas avanzadas se emplea *WFUsaLlamada* donde se anexa como información qué operación se está utilizando.

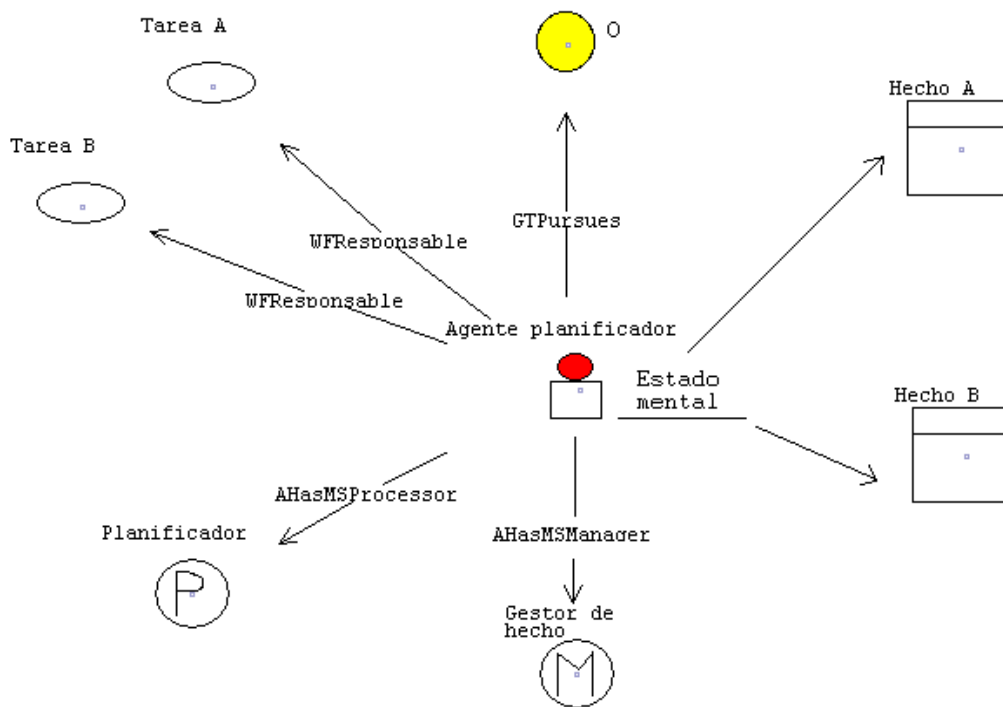


**Figura 24. Meta-modelo de tareas y objetivos. Restricciones sobre elementos consumidos**

Si dentro del mismo flujo de trabajo existen varias instancias de la misma entidad mental, se hace una selección no determinista (*WFConsumeNoDeterminista*) o bien una selección basada en restricciones sobre los atributos asociados a la entidad (*WFConsumeRestricitivo*).

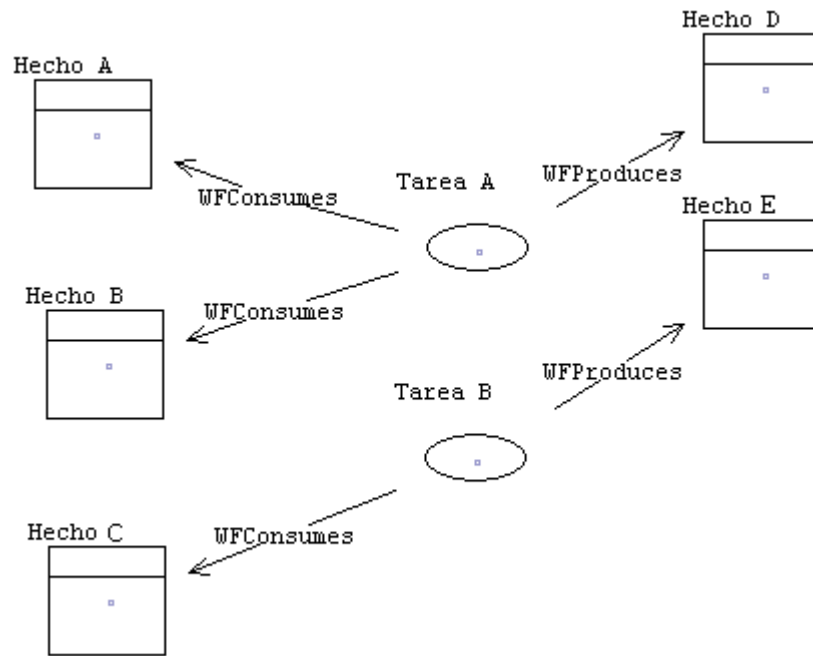
**2.1 Ejemplo: un agente planificador de tareas**

El ejemplo utilizado es el modelado de un agente planificador de tareas. El agente sabe ejecutar dos tipos de tareas: tarea A y tarea B. De la utilización de éstas depende que se alcance el objetivo **O** (Figura 25).



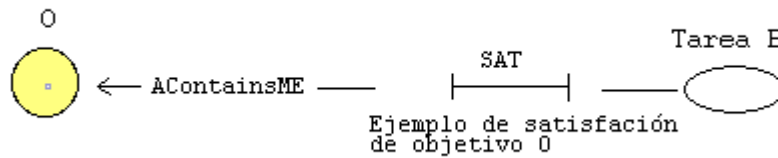
**Figura 25. Modelo de agente para representar el agente planificador.**

Inicialmente, el agente cuenta con los hechos A y B. Con esta información, el procesador de estado mental del agente, que es en realidad un planificador de tareas, se pregunta si será capaz de alcanzar el objetivo propuesto (O). Para determinar si será capaz o no, hace falta más información, concretamente bajo qué condiciones el objetivo O se alcanza y qué efectos tiene la ejecución de las tareas A y B.



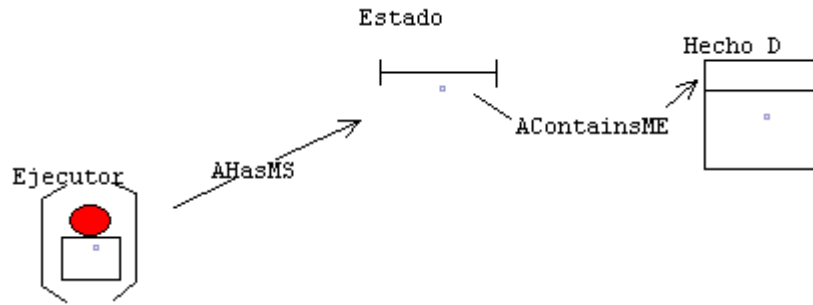
**Figura 26. Modelo de tareas y objetivos para especificar las tareas A y B**

La Figura 26 muestra las consecuencias de ejecutar las tareas A y B. En este caso, ambas se limitan a producir nuevas entidades mentales.



**Figura 27. Modelo de tareas y objetivos para mostrar la satisfacción del objetivo O por la ejecución de la Tarea B**

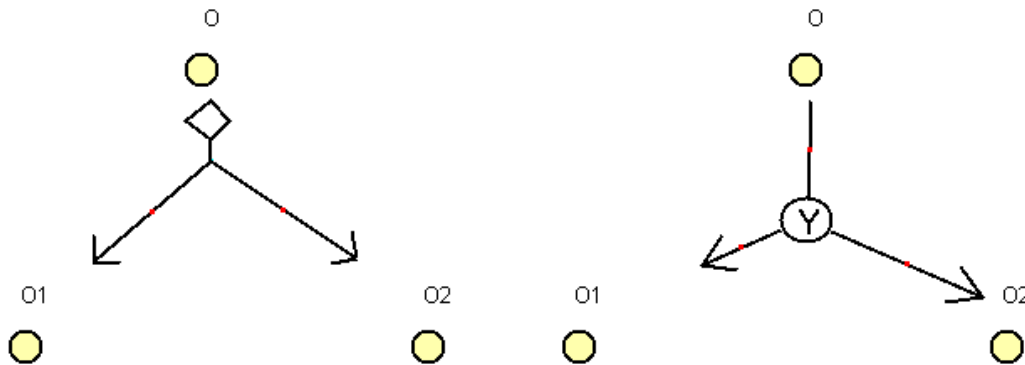
La asociación de objetivos con tareas es fundamental. Como se ha mencionado con anterioridad, hace falta justificar por qué el agente elige un curso de acciones. Aquí, la elección de la Tarea B se justifica porque con ello se alcanza el objetivo O. La relación se complementa con un patrón de estado mental (*Ejemplo de satisfacción del objetivo O*) que denota el estado mental requerido para dar por satisfecho O.



**Figura 28. Modelo de agente para representar el estado mental requerido para alcanzar el objetivo O.**

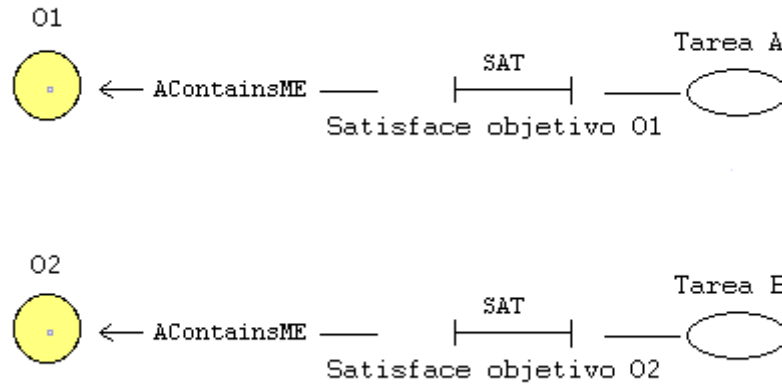
La Figura 28 muestra lo que faltaba, una descripción del estado mental requerido para alcanzar el objetivo O. En este caso, se requiere que quien persiga al objetivo O contenga también el hecho D. A partir de este momento, el procesador de estado mental del agente, que es un planificador, tiene información suficiente para deducir que debe ejecutar la Tarea A antes de ejecutar la Tarea B para alcanzar el objetivo O.

El problema se podría haber especificado de una forma alternativa. En lugar de asociar directamente el objetivo O con la Tarea B, se podría haber aplicado descomposición de objetivos y asociar los nuevos sub-objetivos con las tareas existentes.



**Figura 29. Modelo de objetivos y tareas para mostrar la descomposición de O en O1 y O2 (izquierda) y las dependencias entre éstos (derecha)**

La Figura 29 muestra la descomposición del objetivo O en otros dos (O1 y O2) con los que se establece una dependencia Y. Esto significa que cuando O1 y O2 hayan sido alcanzados, automáticamente, O se dará por alcanzado.

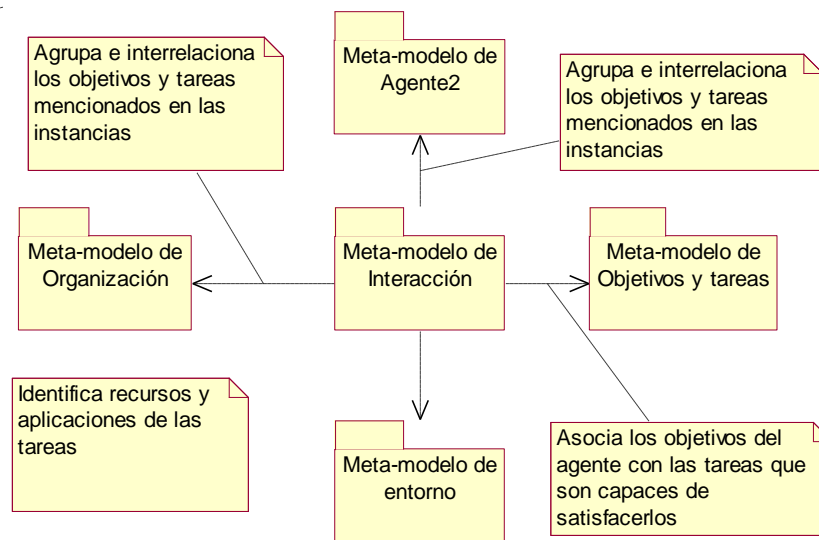


**Figura 30. Modelo de objetivos y tareas para mostrar la asociación de las tareas A y B con los objetivos O1 y O2**

La Figura 30 sirve para asociar los objetivos O1 y O2 con las tareas que sabe ejecutar el agente. Aquí se omite la descripción detallada del estado mental requerido por O1 y O2. Basta decir que para alcanzar O1 y O2 se necesita que existan los hechos C y D.

## 2.2 Integración con otros meta-modelos

Este meta-modelo se orienta principalmente a justificar la ejecución de tareas basándose en los objetivos, que a su vez, se van modificando tras dicha ejecución. Por ello, la mayoría de relaciones con los modelos consiste en recoger, agrupar e interrelacionar los objetivos que aparecen en ellos, mostrándolos como la justificación de la ejecución de tareas (ver Figura 31). Así, las relaciones con los meta-modelos de agente, organización e interacción se reduce a recoger todos los objetivos y tareas que allí aparecen y trabajar con ellos.



**Figura 31. Dependencias del meta-modelo de objetivos y tareas respecto de otros meta-modelos**

La única asociación de propósito diferente es la establecida con el meta-modelo de entorno. Este meta-modelo proporciona recursos y aplicaciones para habilitar las tareas.

Por lo tanto, la mayoría de los criterios de consistencia en este meta-modelo se restringen a verificar que se satisfacen las siguientes condiciones:

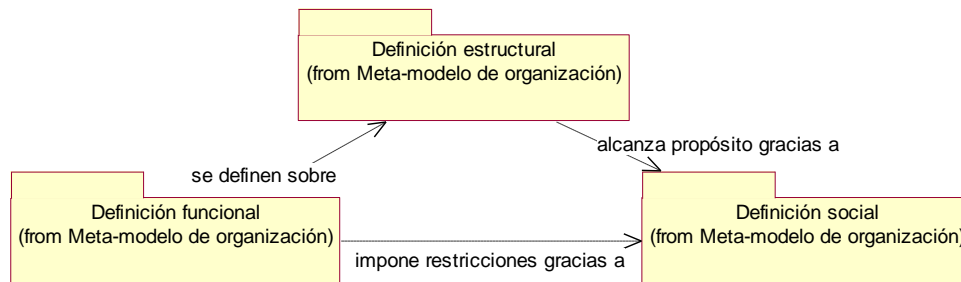
- Todas las tareas y objetivos que aparecen en un modelo de tareas y objetivos también deben aparecer en algún modelo de agente o en algún modelo de organización.
- Si una tarea produce una interacción, debe existir un modelo de organización donde se enmarque esta tarea dentro de un flujo de trabajo y un modelo de interacción para la interacción producida.
- Cuando los resultados de una tarea se necesiten en otra, se entiende que se tiene un flujo de trabajo. Por lo tanto, debe crearse una nueva entidad flujo de trabajo en un modelo de organización y especificar allí cómo se conectan las tareas.
- Todo recurso que aparece en este modelo debe aparecer en un modelo de entorno.
- Las entidades mentales consumidas, producidas, modificadas o destruidas deben pertenecer al estado mental del agente ejecutor. Para modificar el estado mental de otro agente se utiliza una interacción.



### 3. META-MODELO DE ORGANIZACIÓN

El modelo de organización es el equivalente a la arquitectura del sistema en un SMA. El valor principal de un modelo de organización, como ocurre en las organizaciones humanas, son los flujos de trabajo que define. Del estudio de estos flujos surgen nuevas interacciones que reflejan con detalle cómo se coordinan los participantes del flujo. El modelo de organización también contribuye al modelo de tareas y objetivos identificando las tareas relevantes para la organización así como los objetivos que se persiguen globalmente. También define restricciones en el comportamiento de los agentes mediante relaciones como la de subordinación. Gracias a estas restricciones, el diseñador asegura que unos agentes obedecerán a otros o que se comprometerán a la ejecución bajo demanda de tareas respetando sus prioridades.

Para simplificar la presentación del meta-modelo de organización, se han separado tres aspectos del mismo, como muestra la Figura 32.



**Figura 32. Las tres vistas de una organización**

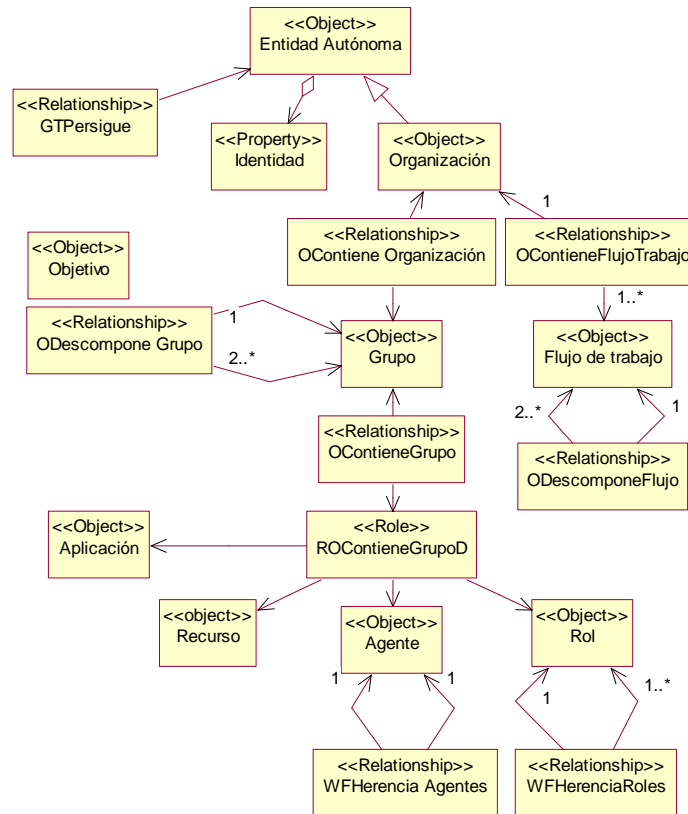
La estructura de la organización define los elementos principales que la componen. Las relaciones sociales establecen relaciones de alto nivel entre los componentes para poner restricciones al comportamiento de la organización. Por último, la definición funcional establece qué ofrece la organización.

#### 3.1 Descripción estructural de la organización

La organización, desde el punto de vista estructural, es un conjunto de entidades asociadas por relaciones simples de agregación y herencia. En esta estructura se define el esqueleto donde van a existir los agentes, los recursos, tareas y objetivos. Sobre este esqueleto se definen una serie de relaciones que inducen la formación de flujos de trabajo y de restricciones sociales (ver siguientes secciones).

Según la Figura 33, la organización es una *entidad autónoma*, como lo es también un agente. Los objetivos perseguidos por la organización son los objetivos comunes a los agentes que la componen y el motivo por el cual se han agrupado. No obstante, una organización no es un agente. La diferencia fundamental es que la organización no tiene capacidad de ejecutar tareas ni de tomar decisiones, son los agentes que la componen quienes se encargan de ello.

La visión estructural de la organización proporciona la descomposición de la organización en grupos (*OContieneOrganizacion*) y flujos de trabajo (*OContieneFlujoTrabajo*). La descomposición recursiva de grupos y flujos se realiza mediante *ODescomponeGrupo* y *ODescomponeFlujoTrabajo*, respectivamente.



**Figura 33. Meta-modelo de organización. Visión estructural**

Cada grupo (*OContieneGrupo*) contiene agentes, recursos, aplicaciones o roles. La asignación de estos elementos a un grupo obedece a propósitos organizativos, esto es, están agrupados porque clarifican la creación de flujos de trabajo. Cualquiera de ellos puede pertenecer a otros grupos de la misma u otras organizaciones. La utilidad de los grupos aparece cuando el número de estos elementos (agentes, roles o recursos) empieza a ser poco manejable. Los grupos son herramientas inestimables para la estructuración de la organización, ya que permiten al desarrollador aplicar divide-y-vencerás para facilitar el problema de la definición del sistema. En cuanto a la gestión de los grupos, a nivel de meta-modelo no se expresa, ya que se pueden definir roles de tipo *gestor de comunidad* desempeñados por agentes que realicen estas funciones.

Para aumentar la capacidad de abstracción e incrementar la reusabilidad de los agentes y roles, se ha incluido la posibilidad de extenderlos. La herencia entre agentes es una herencia simple, mientras que con los roles se admite herencia múltiple. La semántica de la herencia es parecida a la semántica de la herencia de objetos.

Así, si un agente A extiende un agente B, el agente A puede sustituir a un agente B en cualquier situación. Y si un rol A, extiende los roles B1 hasta Bn, entonces A puede sustituir a cualquier Bi en cualquier situación. Como restricción, se exige que no se redefina ningún aspecto ya establecido en el ancestro (por ejemplo asociaciones con tareas, estados mentales, participación en flujos de trabajo) y que las nuevas capacidades de los agentes no creen conflictos con las de sus ancestros (por ejemplo asociando tareas que hagan lo contrario de otras que ya existan).

### 3.2 Descripción funcional

El objetivo del flujo de trabajo es establecer cómo se asignan los recursos, qué pasos (tareas) son necesarios para la consecución de un objetivo, y quiénes son los responsables de ejecutarlas. En el flujo de trabajo, se habla de actividades en lugar de tareas, aunque en este contexto se pueden emplear indistintamente.

Dentro del flujo de trabajo se presentan dos tipos de información: el primero tiene que ver con la forma en que se asocian unas tareas con otras y el segundo, cómo se ejecutan. La forma de presentar estos dos aspectos es a través de instancias de meta-relaciones que interconectan entradas y salidas de las tareas y los responsables de ejecutarlas (Figura 34).

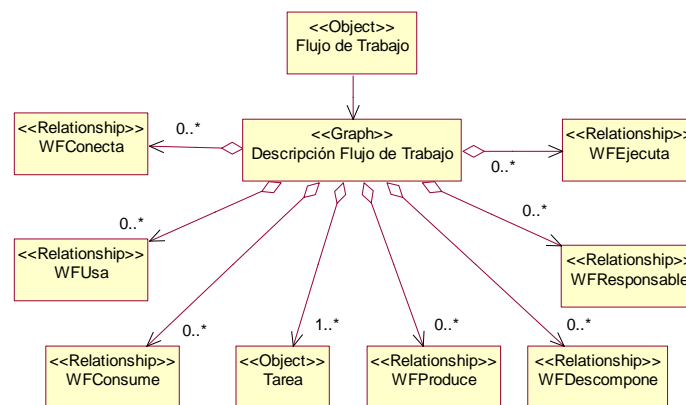


Figura 34. Meta-modelo de organización. Elementos que componen un flujo de trabajo

Algunas relaciones (concretamente *WFConsume*, *WFUsa*, *WFProduce*, *WFDescompone*) ya se han visto dentro del meta-modelo de objetivos y tareas. Las únicas nuevas son *WFEspecificaEjecucion* y *WFConecta* (ver Figura 35). *WFEspecificaEjecucion* indica que la especificación concreta de las condiciones de ejecución de una tarea se presenta dentro de una interacción. Se ha adoptado esta solución para correlacionar tareas ejecutadas por distintos actores. *WFConecta* se corresponde con la conexión de las salidas (instancias de *WFProduce*) de una tarea con las entradas de otra (instancias de *WFConecta*). La conexión no es completa en el sentido de que hay necesidad de que todo lo producido por una tarea deba ser consumido por la otra. Para determinar exactamente qué se consume, hay que considerar las instancias de *WFConsume*.

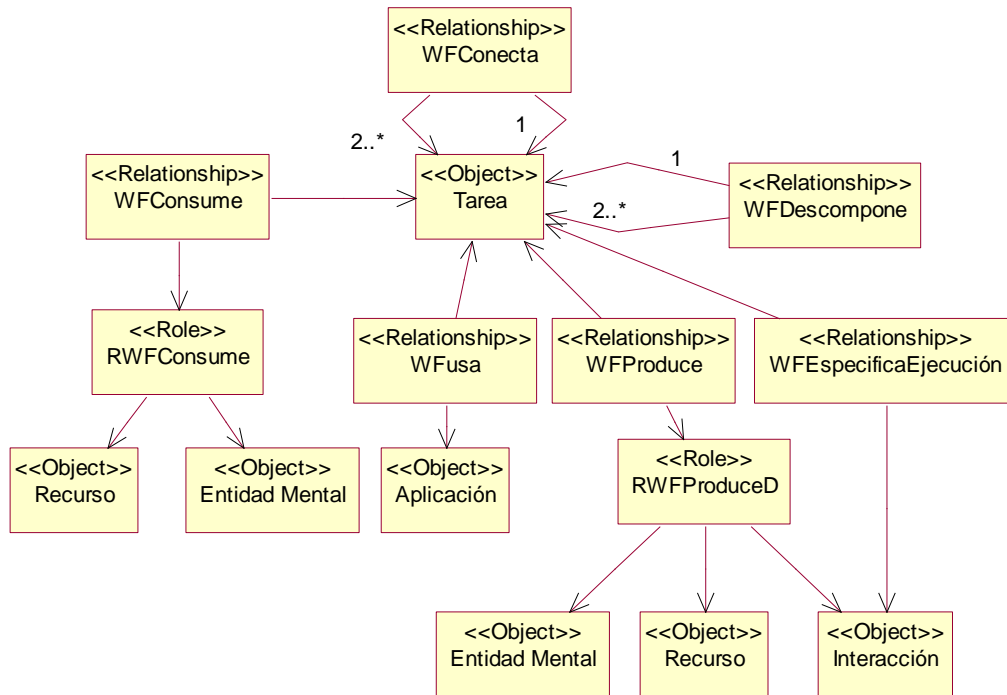


Figura 35. Meta-modelo de organización. Relación entre los elementos del flujo de trabajo

Para permitir descripciones incrementales, se permite que la entidad Flujo de Trabajo actúe como extremo de las relaciones *WFProduce*, *WFConeccta* y *WFConsume* de la misma forma en que lo hace la entidad tarea. De esta forma, se pueden incluir flujos de trabajo dentro de la descripción de los flujos de trabajo.

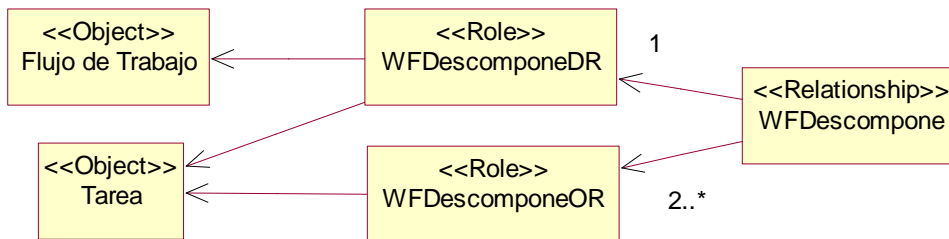
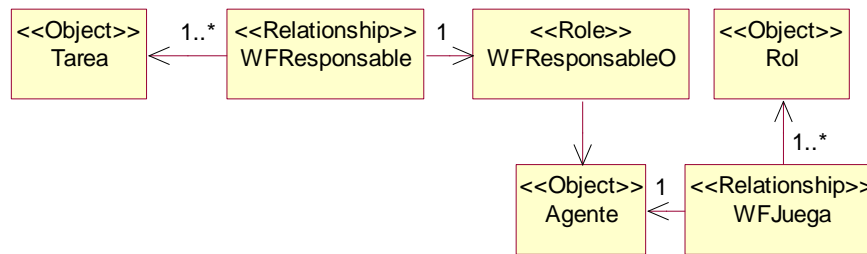


Figura 36. Descomposición de tareas y flujos de trabajo

Es posible además, que un flujo de trabajo se descomponga en otros flujos de trabajo o tareas. No se permite la descomposición de una tarea en flujos de trabajo.

Las tareas son ejecutadas por agentes que son directamente responsables de ellas (mediante instancias de *WFResponsable*) o indirectamente a través de roles (mediante instancias de *WFJuega*), como indica la Figura 37.

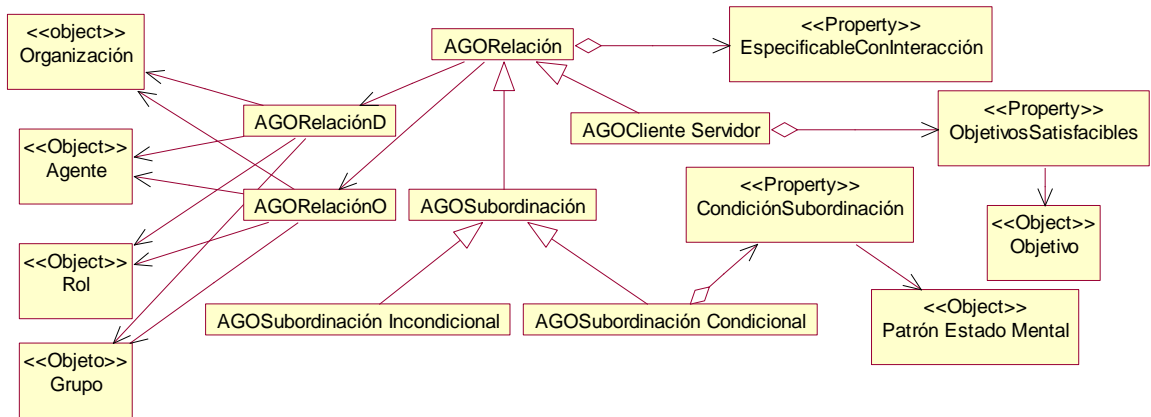


**Figura 37. Meta-modelo de organización. Asociación de tareas con sus ejecutores.**

La ejecución puede ser local o involucrar la ejecución de otras tareas en otros agentes. Ambas variantes se inician de la misma forma y tienen efectos locales idénticos: un agente inicia la tarea motivado por la satisfacción de un objetivo (principio de racionalidad) e indica lo que se produce por instancias de *WFProduce* y su efecto sobre el entorno con *WFUsa* (sobre aplicaciones y recursos). La diferencia entre ambos tipos es que la tarea que involucra ejecución de otras tareas en otros agentes produce interacciones. Las interacciones, indican bajo qué condiciones tiene lugar la ejecución de tareas (instancias de *UIInicia* y *UIColabora*). Para concretar quien participa en las interacciones, se utilizan instancias de *ConsultaEntidadAutónoma*. Estas instancias pueden asociarse a un agente o rol mediante la meta-relación *InstanciaDe*. Las etiquetas asociadas a *ConsultaEntidadAutónoma* harán referencia a elementos consumidos o producidos por las tareas, Figura 47.

### 3.3 Descripción social

Las relaciones que se mencionan aquí entre Organizaciones, Agentes y Grupos, configuran restricciones sociales que limitan la interacción entre estas entidades. En concreto, aquí se incluyen relaciones de subordinación, de prestación de un servicio y de cliente de un servicio. Estas relaciones se definen teniendo como extremos posibles organizaciones, agentes, roles y grupos.



**Figura 38. Meta-modelo de organización. Relaciones sociales entre los componentes de la organización**

Las relaciones de subordinación (*AGOSubordinación*) obligan al subordinado a obedecer en todo y permiten al subordinante dar cualquier orden. Esta relación contradice uno de los supuestos sobre los agentes, que tienen libertad de actuación. Aunque es deseable en algunas ocasiones, es cierto que esta libertad añade nuevas variables al sistema haciéndolo mucho más complejo. Así, estas relaciones son interesantes desde el punto de vista de la ingeniería porque permiten al diseñador asegurar que un agente no deja de obedecer órdenes críticas. Esta obediencia se limita a los agentes que el diseñador establezca, respetando en otros la libertad de desobedecer.

El siguiente paso es descomponer esta relación en obediencia incondicional (*AGOSubordinaciónIncondicional*) y condicionada (*AGOSubordinaciónCondicional*). La incondicional consiste en obedecer todas y cada una de las órdenes. La condicional es asimilable a un contrato, donde si se infringen algunas de las condiciones impuestas, el contrato deja de tener validez. Las condiciones del contrato se establecen utilizando un *Patrón de estado mental*. Con este tipo de patrón es posible expresar:

- **Acciones toleradas.** Aquellas permitidas en el marco de la subordinación. Según el meta-modelo de agente, esto se hace mediante instancias de *CreenciaExistencia*, para relacionar la creencia de la existencia de relaciones *WFJuega* con roles que soporten las acciones requeridas.
- **Fecha hasta la que se mantiene la subordinación.** Se trata de mantener la relación sólo un periodo concreto de tiempo. El periodo se expresa como un intervalo de tiempo, entendiendo que comienza en el momento de establecer la relación. Si se modela la subordinación como una *CreenciaExistencia* cuyo tiempo de vida es limitado y se ha definido un período de validez de conocimiento en el *Gestor de Estado Mental*, basta con comprobar la existencia de la creencia anterior.

- **Estado mental que hay que mantener a lo largo de la subordinación.** Puede entenderse como el invariante de la relación. Si este estado mental no se mantiene, entonces la relación se puede dar por terminada. Se modelaría con una instancia directa de las entidades mentales que deben mantenerse.

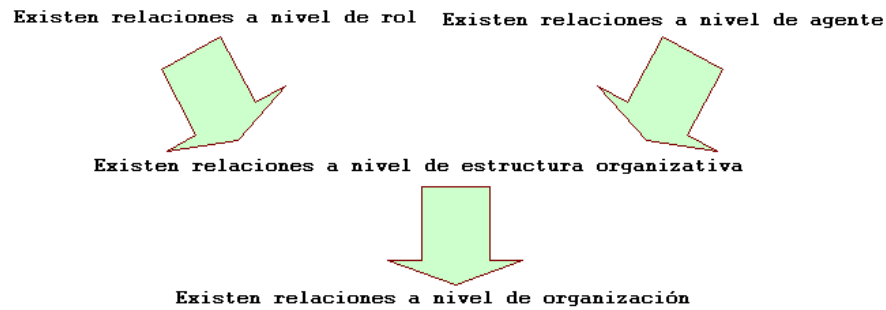
Respecto a las relaciones Proveedor y Cliente (*AGOClienteServidor*), hacen referencia a que existe un servicio ofrecido por un *Servidor* que es requerido por el *Cliente*. En la descripción de proveedor y cliente, y del servicio en sí, existen múltiples variantes. Aquí se propone una alternativa coherente con el meta-modelo ya descrito, en la cual los agentes buscan satisfacer objetivos, y si se ponen en contacto unos con otros, es para conseguir satisfacerlos.

En cierto modo la subordinación condicional es similar a la prestación de un servicio, ya que en ambos se busca ejecutar una acción para otro. Sin embargo, son diferentes en la forma. En la prestación de un servicio, el proveedor puede denegar la petición, mientras que en la subordinación existe la obligación de ejecutar lo pedido.

### 3.3.1 Prioridad entre las relaciones sociales

Entre las distintas entidades de la organización (ver Figura 39) surgen dieciséis posibles combinaciones de dependencias entre organizaciones, grupos, roles y agentes. Para limitar y estructurar estas relaciones, se admiten sólo relaciones entre entidades del mismo tipo. Sobre las relaciones resultantes se plantea una política de aplicación. Se establece que entre dos entidades de la organización pueden existir como máximo relaciones a tres niveles. De mayor a menor nivel, estos niveles son:

- **Nivel organizativo.** Se trata de las relaciones organizacionales. Se dan entre las entidades de más alto nivel y son el equivalente a las relaciones inter-empresariales.
- **Nivel de estructura de organización.** Se trata de las relaciones estructurales. Serían las equivalentes a relaciones interdepartamentales.
- **Nivel de agente y rol.** Se trata de las relaciones de agencia. Serían equivalentes a relaciones directas entre el personal.



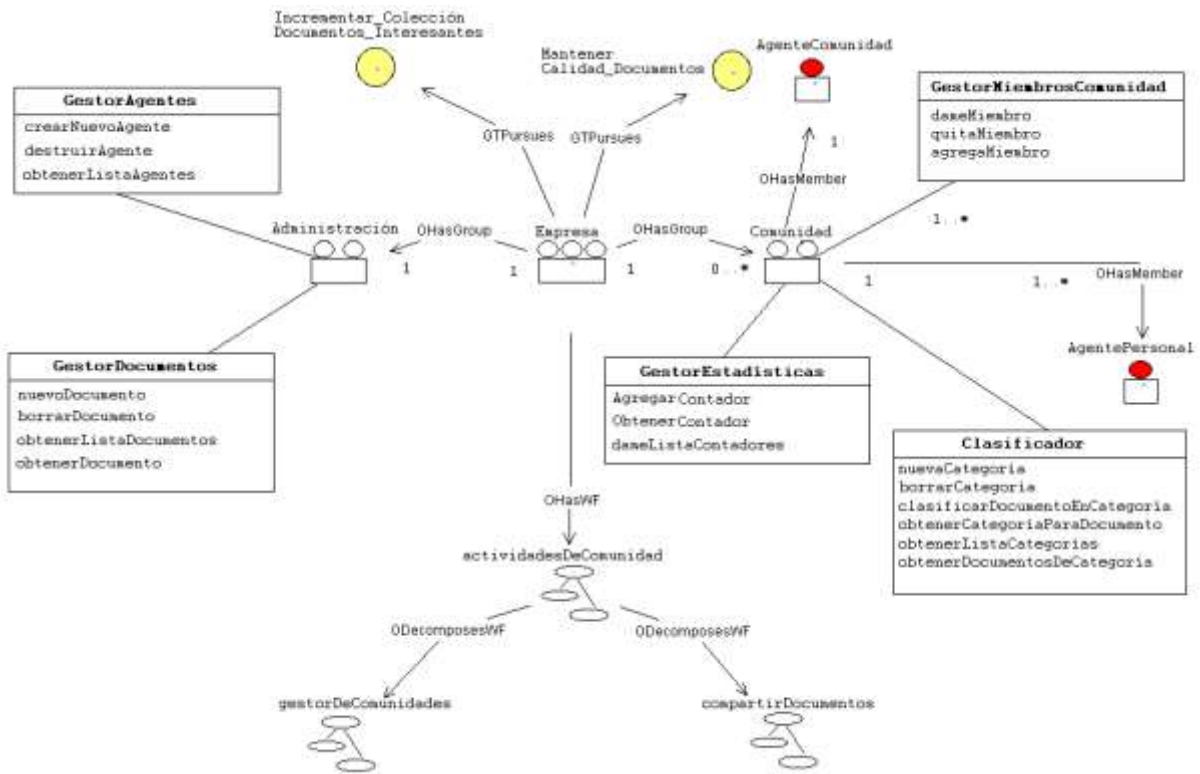
**Figura 39. Meta-modelo de Organización. Semántica sobre las meta-relaciones sociales**

No puede haber relaciones sociales entre dos entidades entre las cuales no exista una relación a nivel inmediatamente superior.

### 3.4 Ejemplos de modelo de organización

Se quiere diseñar un sistema de filtrado colaborativo de documentos donde los usuarios se suscriban a grupos de interés para recibir documentos que coincidan con sus gustos. En este sistema, los usuarios actúan a través de *Agentes Personales*. Estos agentes se suscriben a los grupos o comunidades de interés interactuando con el representante de la comunidad, el *Agente de Comunidad*. Se admite que un usuario pueda estar suscrito a múltiples comunidades. Una vez suscritos, los usuarios pueden sugerir documentos a la comunidad y recibir sugerencias de otros usuarios de la misma comunidad a través de su *Agente Personal*. Como arquitectura del sistema de este SMA, se propone el modelo de organización de la Figura 40.





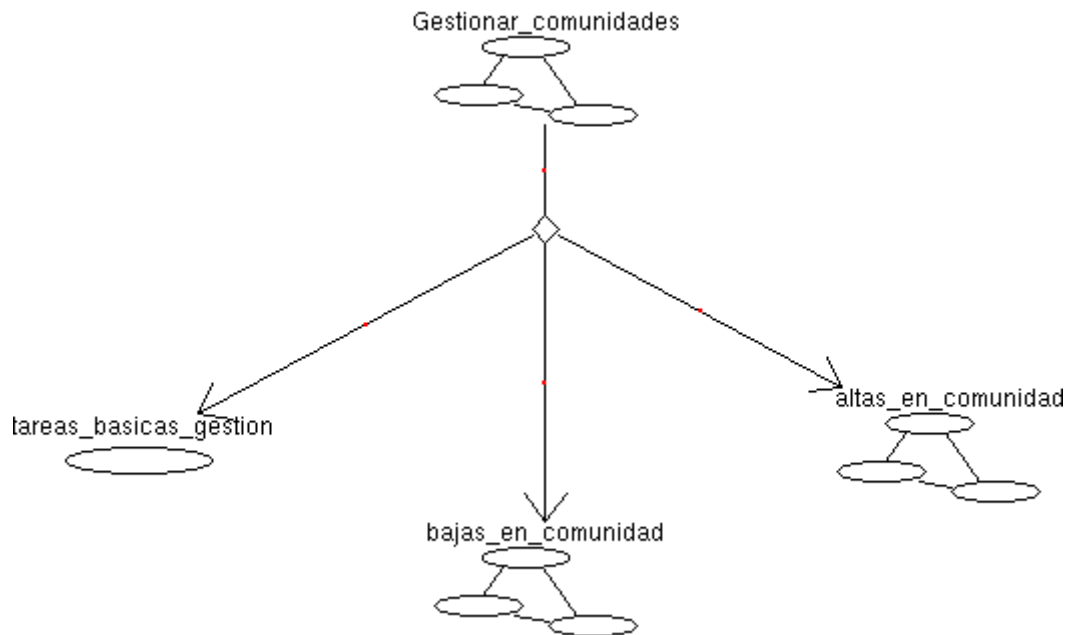
**Figura 40. Estructura de una organización dedicada a la distribución de documentos de interés para sus usuarios**

La empresa consta de dos grupos, uno que representa la administración de recursos compartidos (Administración) y otro que representa comunidades de usuarios (Comunidad). Dentro del grupo Comunidad existe un agente responsable de su administración (altas y bajas de usuarios), el *Agente de Comunidad*, y otros agentes que actúan como miembros de la comunidad, *Agentes Personales*. Los *Agentes Personales* representan al usuario en los procesos internos del sistema. Aquí se verá como ejemplo su participación en la evaluación de nuevos candidatos desempeñando los roles *Miembro de la comunidad* y *Solicitante de suscripción*.

Los flujos de trabajo *Actividades de la comunidad*, *Gestionar comunidades* y *Compartir Documentos* aparecen para estructurar las tareas identificadas con modelos de interacción y con modelos de tareas y objetivos. En estos modelos se ha estudiado qué comportamiento básico se espera de los agentes a la hora de suscribirse a comunidades y sugerir documentos y cómo se alcanzan los objetivos iniciales asociados a los agentes.

Los flujos de trabajo se pueden refinar para obtener flujos de trabajo más especializados. En la Figura 41 se refina la gestión de comunidades en tareas básicas de gestión (añadir un miembro a la comunidad, quitar un

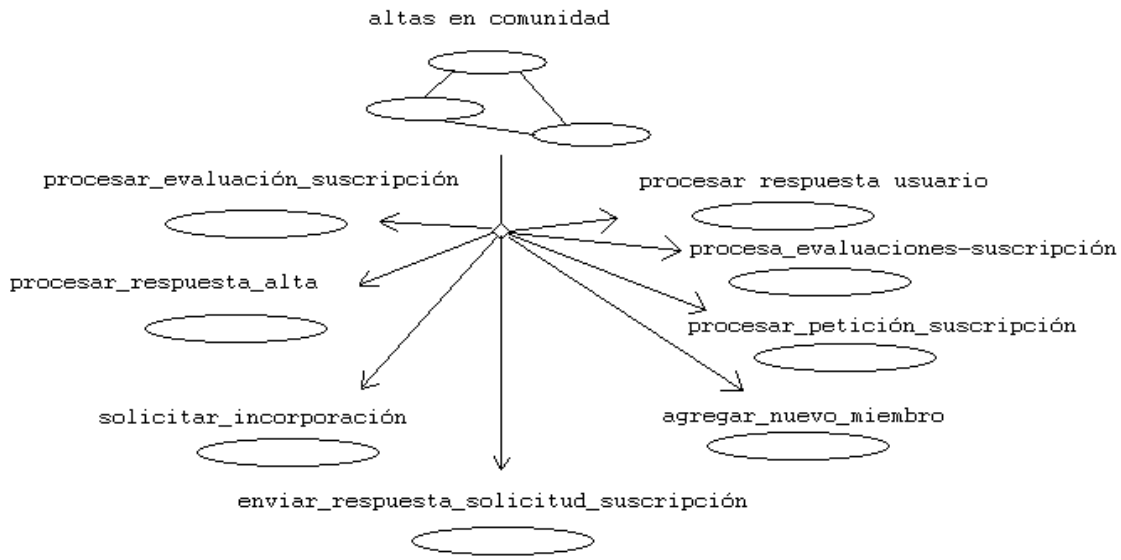
miembro de la comunidad, monitorizar miembros de la comunidad), y flujos de trabajo para determinar bajas y altas en la comunidad.



**Figura 41. La gestión de la comunidad refinada en flujos**

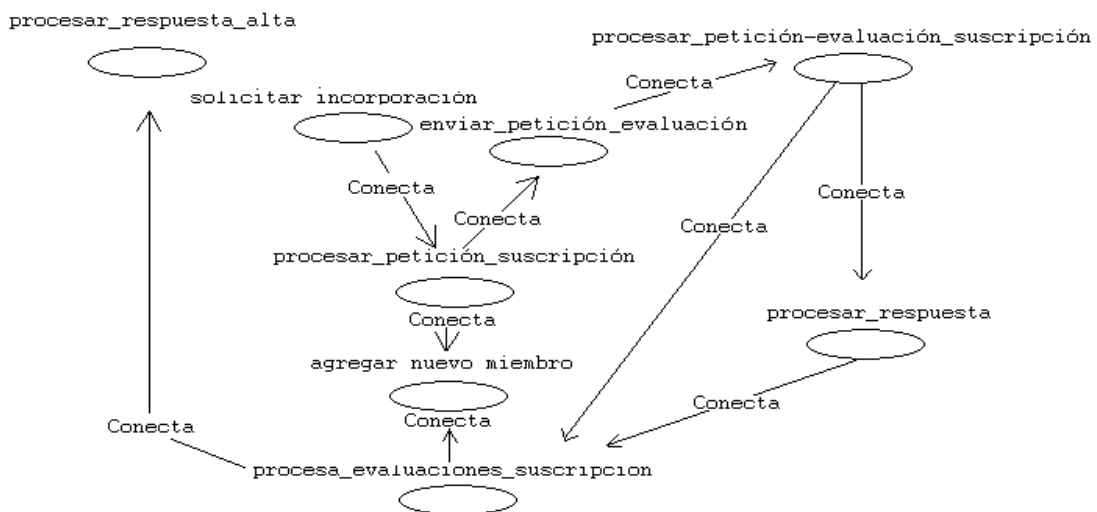
La descripción de los flujos de trabajo comienza indicando qué tareas forman parte del flujo. Como ejemplo, la Figura 42 muestra las tareas que componen el flujo de trabajo para dar de alta a un nuevo miembro en la comunidad.

Las altas en la comunidad se deciden primero en función de la comparación de los gustos del solicitante y los gustos conocidos de los miembros existentes, y después en función de la opinión del resto de los miembros de la comunidad. Para no molestar a todos, se elige un subconjunto de miembros para que emitan su voto. Si la mayoría vota afirmativamente, el solicitante es admitido como nuevo miembro.



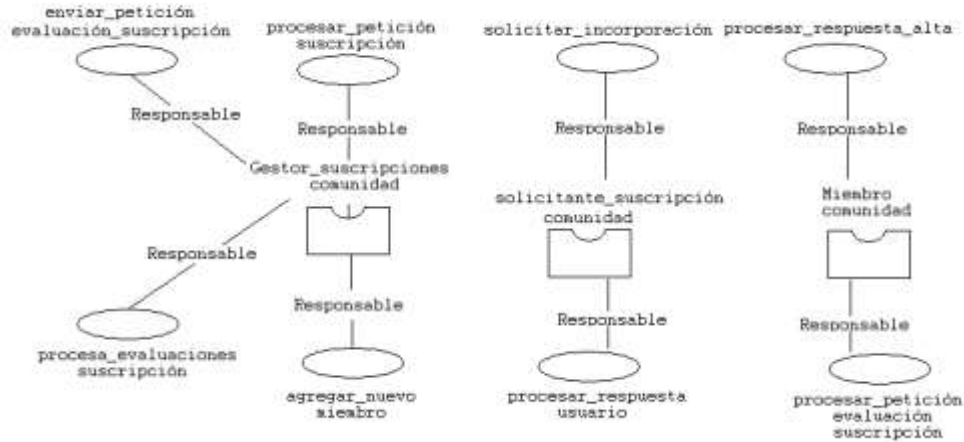
**Figura 42. Tareas asociadas al flujo de trabajo para dar de alta en una comunidad**

Las tareas anteriores se interconectan según las relaciones de la Figura 43. Estas relaciones muestran cómo se conectan las entradas y salidas de cada tarea. La primera tarea en ejecutarse es *solicitar incorporación*, hecha por el solicitante. A continuación, la comunidad hace la solicitud con *procesar petición suscripción*. Si la solicitud coincide con los gustos de los usuarios ya registrados, se procede a ejecutar *enviar petición de evaluación de suscripción* para conocer la opinión de los miembros de la comunidad. Los miembros deciden qué hacer mediante *procesar petición evaluación suscripción*. La comunidad recolecta todas las opiniones de los miembros de la comunidad con *procesa evaluaciones de suscripción*, tras lo cual se procede a incluir el nuevo miembro (*agregar nuevo miembro*) y a informar al solicitante (*procesar respuesta alta*).



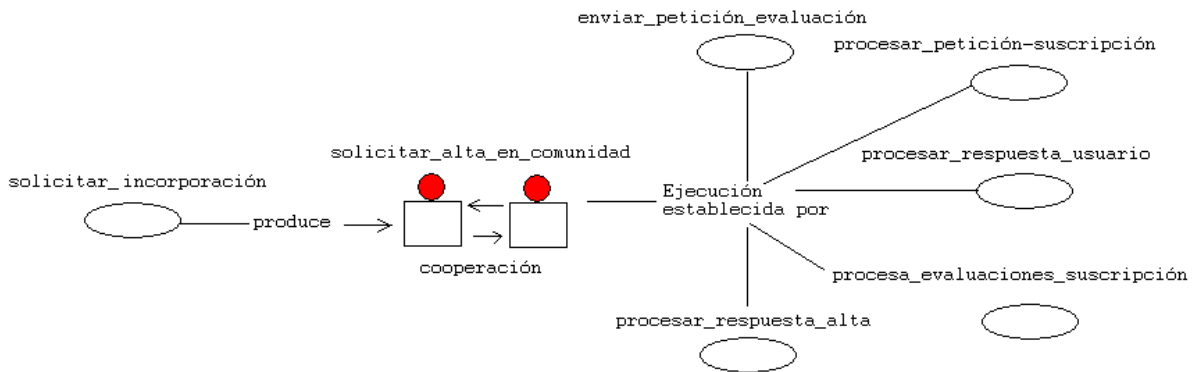
**Figura 43. Asociación de tareas en el flujo de trabajo para dar de alta en una comunidad**

Cada tarea de la Figura 43 es asignada a tres posibles roles: el *gestor de suscripciones de la comunidad*, el *solicitante de alta en comunidad* y los *miembros de la comunidad*. El gestor de suscripciones se encarga de tramitar las solicitudes de altas y bajas en la comunidad. El solicitante de alta en comunidad es el que ha pedido ingresar en la comunidad. El rol miembro de la comunidad designa a cualquier agente que pertenezca a una comunidad. La asociación de estos roles con las tareas que tienen que saber ejecutar se muestran en la Figura 44.



**Figura 44. Responsabilidades asociadas a roles dentro de un flujo de trabajo**

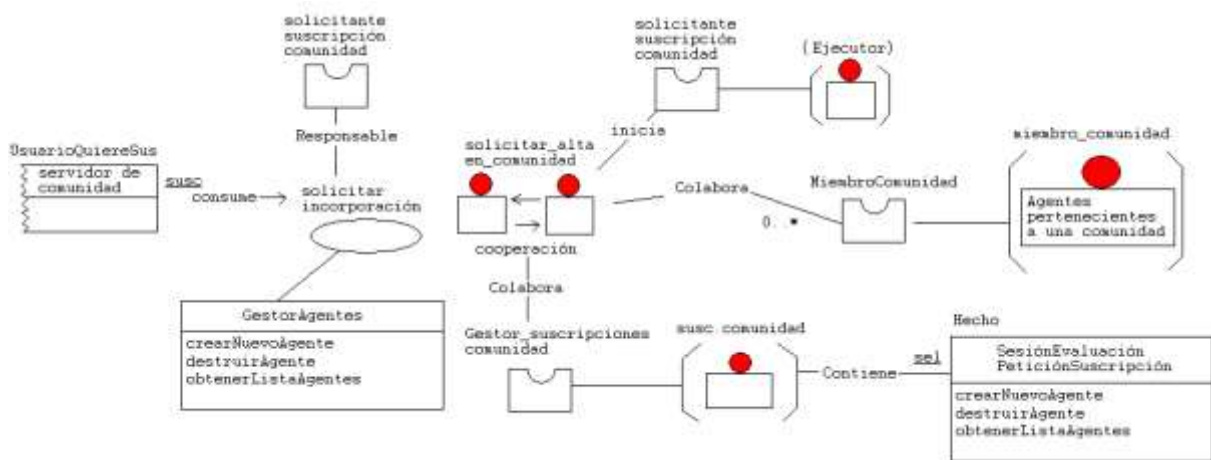
Como hay tareas que no se van a ejecutar por el mismo agente (no se espera que estos tres roles sean desempeñados por un único tipo de agente), se hace que la primera tarea que inicia el flujo de trabajo produzca una interacción donde se indica quién ejecuta qué y cuándo. En el flujo de trabajo se indica una tarea que ocasiona la interacción y otras tareas que se ejecutan como consecuencia de la primera (Figura 45).



**Figura 45. Tareas del flujo de trabajo asociadas a la interacción para dar de alta en una comunidad**

La ejecución de los flujos de trabajo obedece a condiciones mentales de sus ejecutores. Estas condiciones mentales se explicitan dentro de la interacción, indicando por qué tiene lugar la primera y siguientes unidades de interacción.

Para completar la definición del flujo de trabajo, habría que indicar cómo se decide quién cumple los roles que participan en la interacción *solicitar alta en comunidad*. Esta interacción utiliza los roles *gestor de suscripciones de la comunidad*, *el solicitante de alta en comunidad* y *los miembros de la comunidad*, lo cual en el análisis es suficiente. Sin embargo, para completar una definición exhaustiva del flujo de trabajo hace falta señalar criterios para decidir quién desempeñará estos roles. Esta información se proporciona con instancias de *ConsultaEntidadAutónoma*. En este caso, los agentes concretos se determinan a través de hechos que deben existir en el agente ejecutor, como indica la Figura 46. La relación etiquetada como *Contiene* es usada para denotar que la entidad mental pertenece a un estado mental asociado a *ConsultaEntidad Autónoma*.



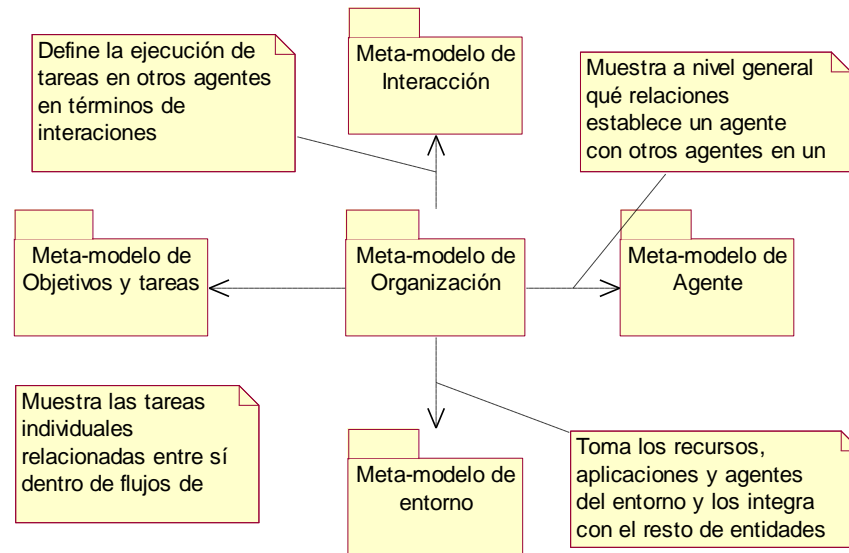
**Figura 46. Descripción de qué agentes desempeñan los roles en la interacción que define la ejecución de tareas del flujo de trabajo para dar de alta en la comunidad**

El Agente Personal, que jugará el rol *solicitante suscripción*, no puede saber quienes jugarán el rol de *miembro comunidad* porque no pertenece todavía a la comunidad. Sólo sabrá quién actuará como *gestor de la comunidad* a la que se quiere suscribir. Este gestor se incluye dentro de un slot (comunidad) del evento etiquetado con *susc*. La localización del agente se hace utilizando *Gestor Agentes* que proporciona la ubicación de todos los agentes del sistema.

El *gestor de la comunidad* fijará más tarde el resto de los roles con el hecho *SelecciónEvaluaciónPeticiónSuscripción*. Este hecho contiene los colaboradores en la interacción, que son los miembros de la comunidad a los que se preguntará acerca de la admisión. Estos miembros se expresan mediante una *ConsultaRequisitosAgente* que describe los agentes involucrados. La descripción es una instancia de *Descripción Agente* denominada *agentes pertenecientes a una comunidad* que codifica los requisitos mediante un predicado PROLOG o en lenguaje natural.

### 3.5 Integración con otros meta-modelos

El meta-modelo de organización mantiene fuertes dependencias con los otros meta-modelos, ya que muestra a nivel general cada entidad representada por estos.



**Figura 47. Meta-modelo de organización. Relaciones con otros meta-modelos**

Las restricciones inducidas por la relación de este Meta-modelo con los demás son las siguientes:

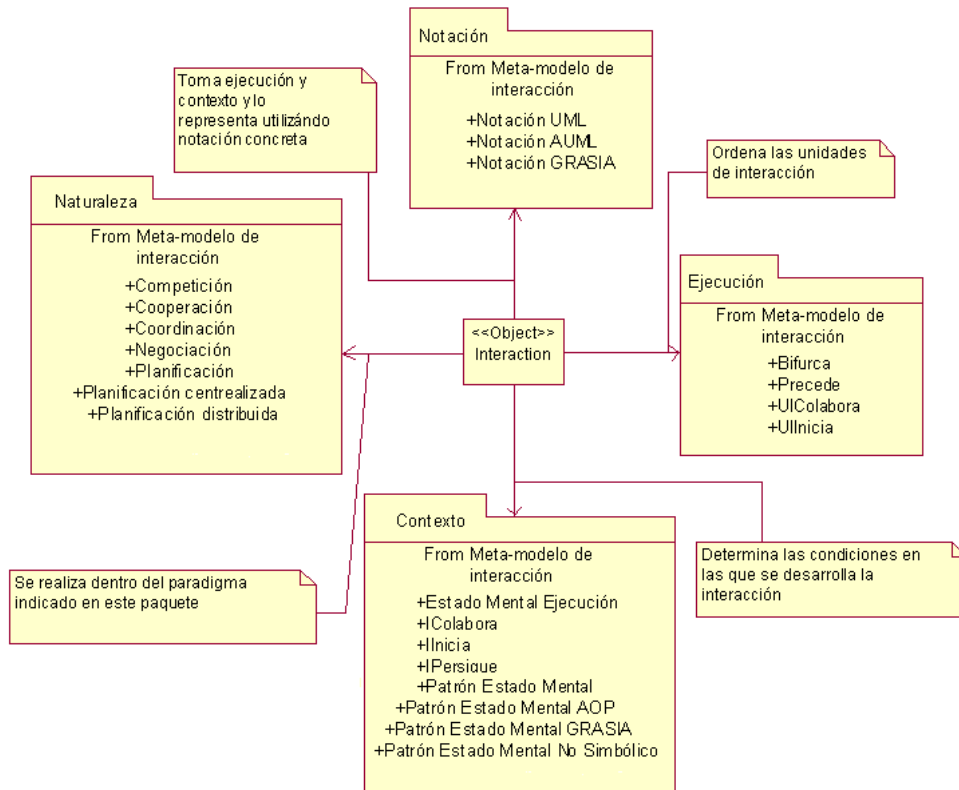
- Las tareas de las que se responsabiliza un agente, así como los roles que juega, deben aparecer también en el modelo de agente donde éste se describe.
- Las tareas que aparecen en la organización han de ser asignadas a algún agente o rol.
- Las tareas que se conectan con otras tareas asignadas a otros agentes o a roles que no puedan ser desempeñados por el mismo agente, deben aparecer en una interacción.
- Las tareas que aparecen conectadas deben compartir al menos una entrada y una salida.
- Todos los recursos y aplicaciones que aparecen en el modelo deben aparecer en algún modelo de entorno.
- Todos los agentes que aparezcan deben tener un modelo de agente donde se les describa. En estos modelos, los agentes se asociarán a los roles identificados en los flujos de trabajo de la organización.
- Las tareas deben especificarse con detalle dentro de un modelo de tareas y objetivos. En este modelo se describirán las entradas y salidas de la tarea, recursos consumidos y aplicaciones utilizadas. Además, la ejecución de la tarea se debe relacionar con la satisfacción de alguno de los objetivos previstos.

#### 4. META-MODELO DE INTERACCIÓN

El meta-modelo de interacciones busca simplificar:

- **La definición del contexto de la interacción.** Esto ayuda al diseñador a saber qué se busca con la interacción y cómo va a afectar al sistema.
- **Las interacciones no están aisladas.** Es posible componerlas y relacionarlas para obtener especificaciones del comportamiento del sistema. Del mismo modo, también se puede partir de interacciones con el sistema y descender hasta las especificaciones de los componentes, que en este caso son agentes.
- **La definición de la ejecución de la interacción.** Ya sea utilizando paso de mensajes, protocolos FIPA o invocación de procedimiento remoto. De esta forma se logra independencia del medio de comunicación utilizado.
- **La representación de la interacción.** Al abstraerse de las posibles unidades de mensajes y de su ordenación, se pueden utilizar diferentes notaciones para representar lo mismo.

El meta-modelo de interacción se construye sobre: agentes, roles, objetivos, interacciones y unidades de interacción. Los agentes y roles son los actores de las interacciones. En las interacciones se ejecutan unidades de interacción (pasos de mensaje, lectura y escritura) en las que hay un iniciador (emisor) y colaboradores (receptores). Además, se justifica la participación de los actores en la interacción y la existencia de esta mediante objetivos.



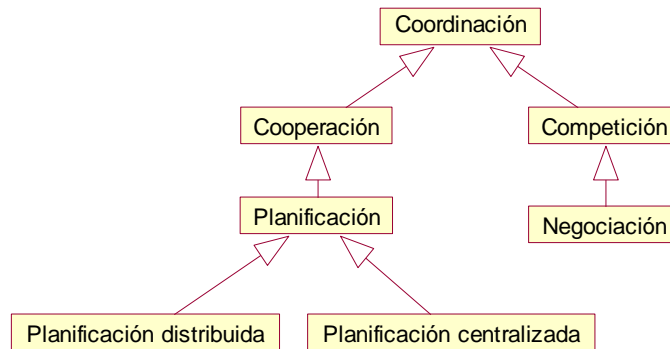
**Figura 48. Relación entre los diferentes aspectos de la interacción**

Estos elementos se interrelacionan mediante el conjunto de asociaciones de la Figura 48. En esta Figura, se categorizan las asociaciones en los cuatro grupos ya vistos: naturaleza, contexto, ejecución y notación.

El contexto ayuda al diseñador a situar la interacción en el marco del SMA, estableciendo en qué condiciones se iniciarán las interacciones. Esta información se utiliza para codificar el comportamiento del agente en forma de reglas de producción o para generar métodos de validación de la interacción. En el metamodelo, el contexto se expresa con los estados mentales en ejecución que deben reunir los agentes y con las meta-relaciones IPersigue, IColabora e Inicia.

La naturaleza de la interacción indica qué clase de interacción se está considerando con respecto al tipo de control. Esta clase de control es de lo que trata la coordinación. La coordinación es la gestión de las dependencias entre actividades [Malone y Crowston 94]. Esta teoría se ha restringido al ámbito de la informática tomando una taxonomía de Huhns [Huhns 00] (Figura 49).



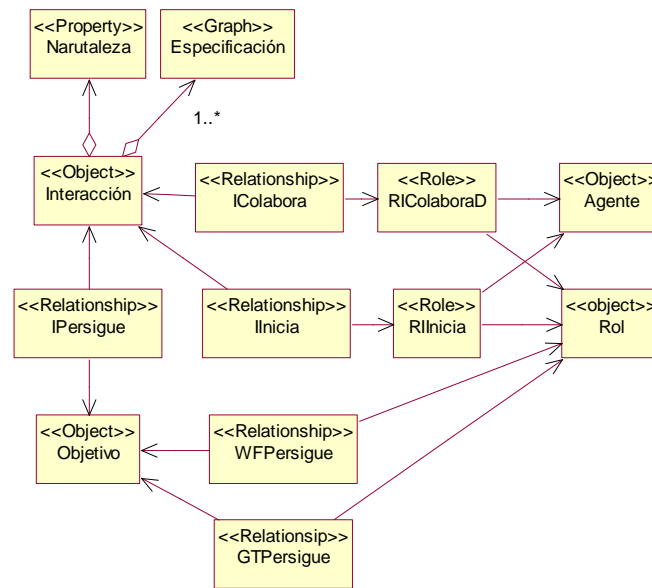


**Figura 49. Taxonomía de métodos de coordinación entre agentes**

La naturaleza de la interacción es importante porque determina qué algoritmos, herramientas y guías pueden ayudar al diseñador. El tipo de interacción también muestra qué elementos tienen que considerarse. Por ejemplo, si la naturaleza es competitiva entonces debería existir en la interacción una referencia al objeto por cuya posesión o uso se compete.

La ejecución de la interacción se refiere al conjunto de actividades requeridas para desarrollarla. En la ejecución existe un orden impuesto sobre las acciones y unidades de interacción. Este orden establece el protocolo de la interacción, mientras que las unidades representan mensajes o protocolos, dependiendo del nivel de abstracción.

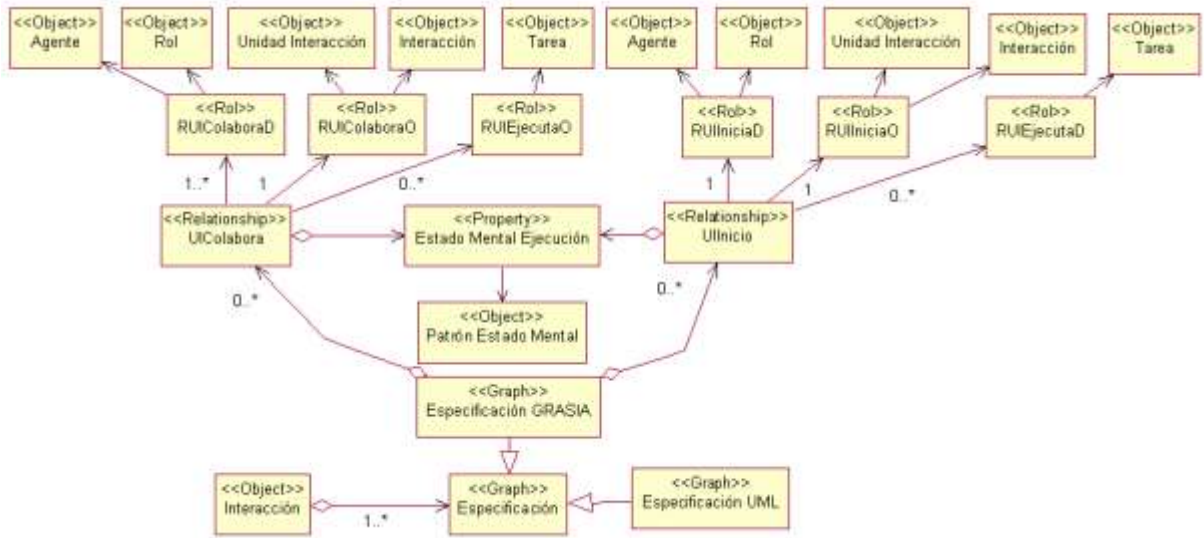
Finalmente, la representación toma la información de la ejecución y la naturaleza para crear una representación gráfica o formalismo textual sencillo de manejar para los diseñadores. Por ejemplo, una negociación que siga el protocolo de interacción contract-net [Smith 80] puede expresarse con diagramas de protocolo de Agent UML, diagrama de protocolos FIPA o varios diagramas de secuencia UML.



**Figura 50. Meta-modelo de interacción**

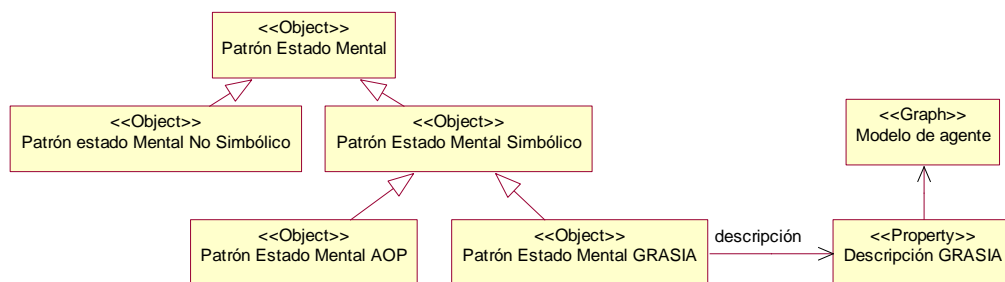
Este meta-modelo cubre los aspectos mencionados referidos en la Figura 50: contexto, naturaleza, ejecución y representación. El contexto aparece como la motivación de la interacción (meta-relación *IPersigue*) y de los roles en el momento de participar en la interacción (meta-relaciones *WFPersigue* y *GTPersigue*). La naturaleza se asocia directamente como propiedad de la interacción. La ejecución y representación han sido agrupadas dentro la entidad *Especificación*, que es del tipo Graph. Según GOPRR, esto implica que *Especificación* es un conjunto de relaciones, objetos y roles. Esta abstracción se usa para denotar las posibles formas de especificar la ejecución y tipo de las unidades de interacción. En este trabajo se ha experimentado con dos tipos de especificación:

- **Diagramas de colaboración UML.** En estos diagramas, las unidades de interacción son mensajes y la asociación entre emisor y receptor son las relaciones *UIInicia* e *UIColabora*. Las tareas pueden agruparse al igual que los agentes o roles, pudiendo recibir mensajes. El orden de ejecución así como las condiciones mentales se expresan con una Expresión de secuencia (ver [OMG 00b]).
- **Diagramas especializados GRASIA.** Los diagramas de colaboración UML no están pensados para el modelado de interacciones entre agentes. Así, la justificación de por qué se está ejecutando la interacción, detalles acerca de por qué se están aceptando ciertos mensajes y por qué transcurre la interacción de un modo concreto, no son fácilmente expresables. Por ello, se ha generado una especificación GRASIA adaptada a su uso dentro de la metodología (ver Figura 51). Esta última variante de especificación da cabida a la representación del estado mental del iniciador y de un colaborador en una unidad de interacción.



**Figura 51. Unidades de interacción y su relación con tareas y roles dentro de una especificación.**

Las unidades de interacción se detallan en la Figura 51. La composición de unidades de interacción siguiendo un orden determinado lleva a la especificación de la ejecución de la interacción. De forma similar a las interacciones, una unidad de interacción también describe quién la inicia (*UIInicia*), quién colabora (*UIColabora*) y qué tareas se ejecutan en cada una (*RUIEjecutaD*). Aunque no se menciona explícitamente, hay que entender que en una especificación GRASIA, también se tienen tareas, agentes, roles, interacciones y unidades de interacción.



**Figura 52. Patrones de estado mental asociables con las unidades de interacción**

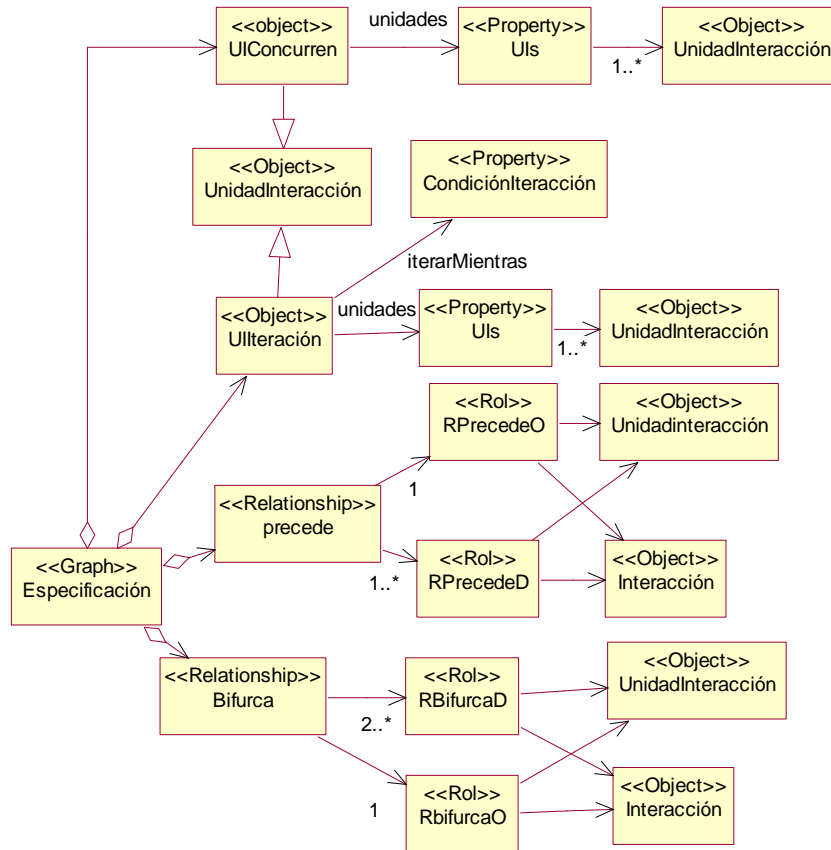
Los patrones de estado mental (ver Figura 52) se usan para expresar condiciones de comienzo y colaboración en términos de entidades de estado mental. La ejecución de tareas no tiene un patrón de estado mental asociado porque la condición implícita de ejecución de una tarea es la asociada a la meta-relación *UIInicia* o *UIColabora*. Así, cuando un agente comienza una unidad de interacción, también ejecuta las tareas asociadas mediante *RUIEjecutaD*.

---

Las condiciones que más se han estudiado en este trabajo son las expresables con la gramática expuesta en Agent0 [Shoham 93] (Patrón Estado Mental AOP) y las expresables como agregación de entidades mentales (Patrón Estado Mental GRASIA), ya que su traducción a reglas de producción es sencilla. En ambas variantes, aparece una descripción de cómo se espera que sea el estado mental del agente en un momento concreto. En el caso de Patrón Estado Mental GRASIA, la descripción es un modelo de agente donde se indica qué entidades deben existir asociadas al estado mental del agente y qué condiciones se deben satisfacer. Cada entidad mental se etiqueta, en caso necesario, para referenciarlas cuando se exprese qué se requiere de estas entidades.

Finalmente, existe una meta-representación de los posibles ordenes que pueden aplicarse sobre las unidades de interacción (ver Figura 53). Los diseñadores pueden elegir el tipo de orden más apropiado para el problema que les ocupe. De esta forma, es factible definir un orden como el de los behaviour de JADE [Bellifemine, Poggi y Rimassa 01] usando las primitivas de comportamiento y ejecutando una traducción directa desde la especificación a las estructuras finales de JADE.

Para representar los posibles órdenes de ejecución de JADE, UML y AUML, se utilizan tres primitivas: orden secuencial (*Precede*), orden no determinista (*UIConcurren*), selección (*bifurca*) e iteración (*UIIteracion*)



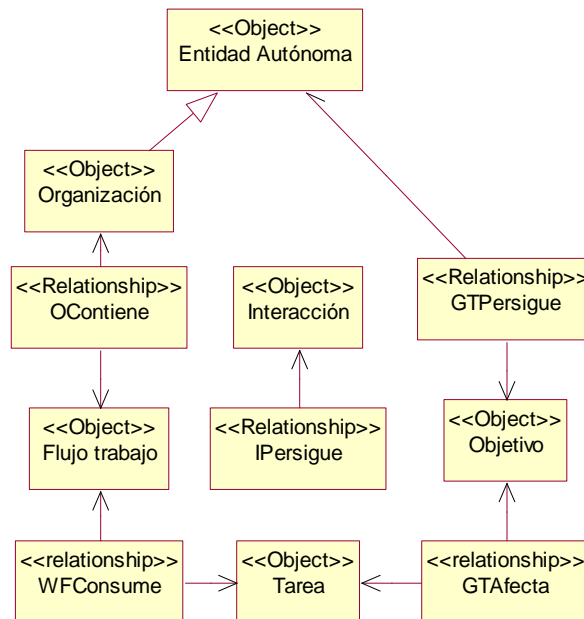
**Figura 53. Ordenes de ejecución utilizables sobre las unidades de interacción**

La meta-relación *Precede* define secuencias de unidades de interacción. Siendo A y B1,...,Bn unidades de interacción; la asociación A - Precede - (B1,...,Bn) significa que la unidad A tiene lugar siempre antes que la unidad Bi. La meta-relación *Bifurca* sirve para expresar condiciones de precedencia entre varias unidades de interacción. Siendo A y B1,...,Bn unidades de interacción, la asociación A - Bifurca - (B1,...,Bn) significa que sólo un Bi tendrá lugar después de A. Cuál será, depende de las condiciones de colaboración o iniciación dadas por *Estado Mental Ejecución* en la Figura 52, ya que para que tenga lugar una unidad de interacción se han de cumplir las condiciones de colaboración e iniciación asociadas. En caso de que varias se cumplan, se elegirá una de forma no determinista. La entidad *UIIteración* define la ejecución repetida de un conjunto de unidades de interacción. Estas entidades aparecen asociadas como una colección (UIs). El orden de ejecución dentro del bucle se determina aparte con las otras primitivas. La condición de terminación es que ninguna unidad de interacción de las contenidas sea ejecutable o bien que se cumplan las condiciones de inicio y colaboración de la unidad de interacción que se vaya a ejecutar a continuación. La entidad *Concurren* especifica un orden de ejecución no determinista sobre las unidades de interacción asociadas. La unidad de

interacción *Concurrent* se da por terminada cuando todas las unidades de interacción que tiene asociadas hayan tenido lugar.

Por medio del meta-modelo de interacción se determina cómo reaccionan los agentes frente a las acciones de otros agentes. La composición de instancias de este meta-modelo da lugar a la definición detallada del comportamiento del sistema. Esta composición se realiza en el marco de la organización y más concretamente dentro de los flujos de trabajo.

Para cada una de las instancias se define un contexto concreto, que es el fijado por el flujo de trabajo de las organizaciones. Al final, las interacciones entre agentes se inician porque se han ejecutado tareas que satisfacen un objetivo en la organización u organizaciones a las que pertenecen los participantes de cada interacción.



**Figura 54. Relaciones entre la interacción y la organización**

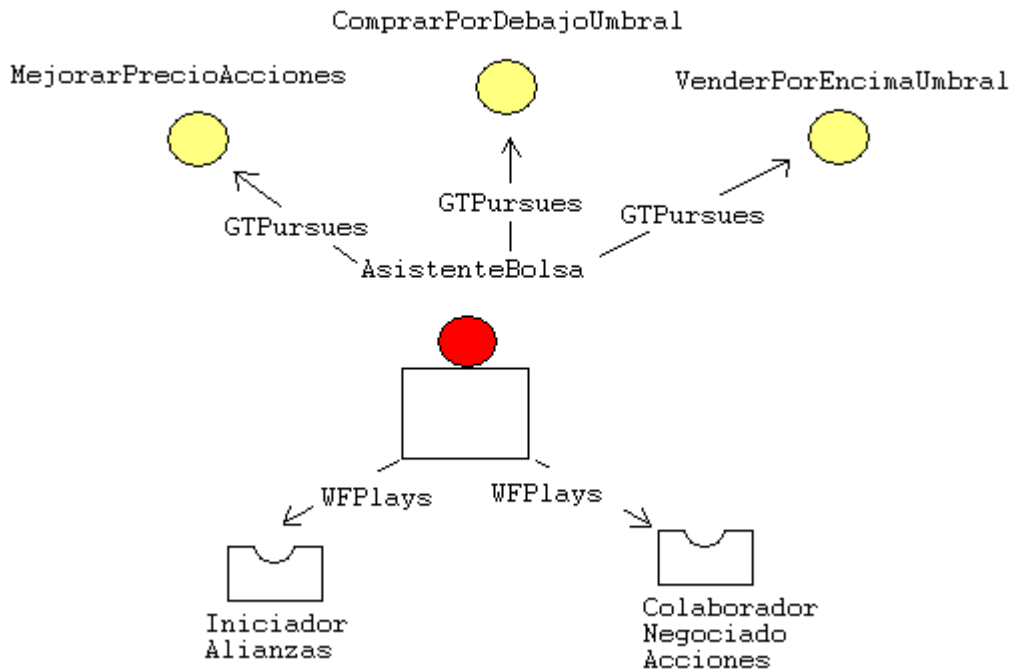
La Figura 54 muestra las dependencias entre la interacción y la organización. El nexo entre interacciones y organización es el objetivo perseguido por ambas. Como las tareas, la organización se vale de interacciones entre sus miembros para alcanzar sus objetivos.

#### 4.1 Ejemplo: agente de bolsa

Se quiere diseñar un asistente para vender y comprar acciones en la Bolsa de una ciudad. El asistente representa a un usuario con el que se confirma las transacciones de las acciones vía móvil. Existe la posibilidad de que el agente desarrolle su labor sin intervención del usuario. Esta situación se da cuando el

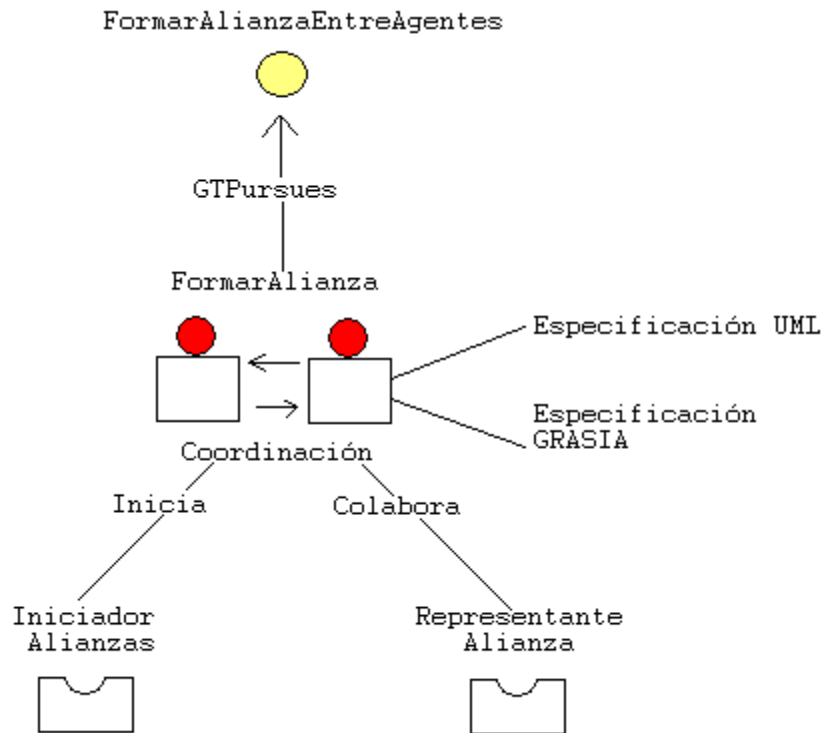
usuario ha configurado un margen de variación en el precio de las acciones que dispara la compra (en caso de que bajen las acciones que interesen al usuario) y la venta (en caso de que suban las acciones del usuario). Para realizar las operaciones de compra y venta, la Cámara de Comercio ha habilitado servidores con los que se puede obtener información en tiempo real del valor de las acciones.

Buscando influir sobre el precio de las acciones en bolsa, el asistente acude a directorios de localización de agentes para localizar a otros agentes de compra-venta de acciones con los que aliarse. Una vez localizados, nuestro asistente plantea alianzas estratégicas con las que hacer variar la bolsa.



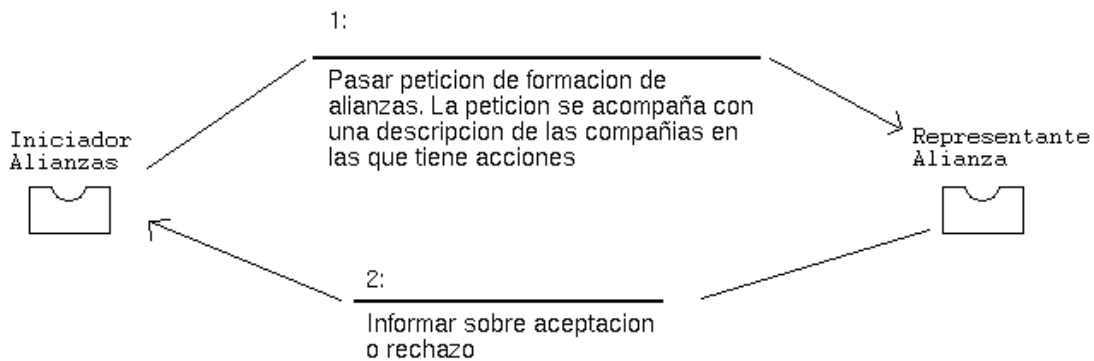
**Figura 55. Definición del asistente de bolsa (modelo de agente)**

Las alianzas se plantean mediante negociaciones con representantes de agentes ya aliados. En el caso de que no estén aliados, el agente con el que se contacta actúa como su propio representante. Tras asociarse con otro agente o agentes, hay que darse de alta en un registro de asociaciones, para que otros agentes puedan unirse a esta alianza. El protocolo de creación de alianzas se representa mediante una interacción *FormarAlianza*.



**Figura 56. Interacción para la formación de alianzas**

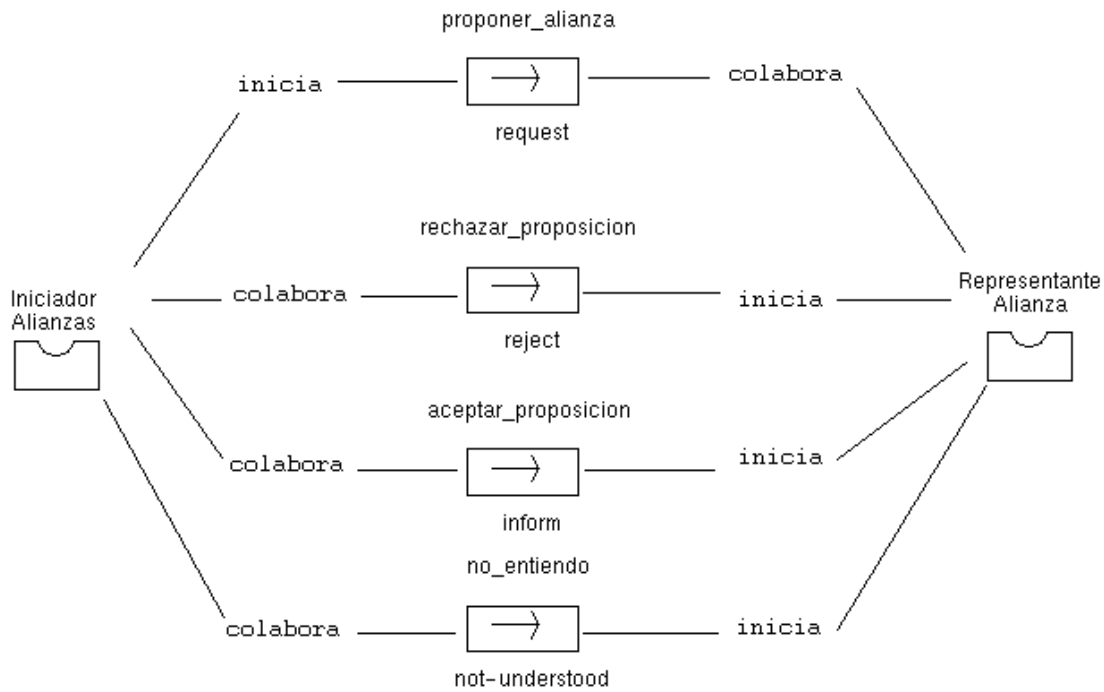
*FormarAlianza* persigue el único objetivo de formar una alianza entre agentes para conseguir mover las acciones de bolsa. La descripción del protocolo se describe inicialmente con un diagrama de colaboración (ver Figura 57).



**Figura 57. Descripción inicial de la interacción para formar alianzas con diagramas de colaboración**

La descripción con diagramas de colaboración no refleja, por ejemplo, el acto del habla involucrado en cada paso de mensaje. A la hora de dar más detalle, se elige una especificación GRASIA para representar la misma interacción (ver Figura 58).





**Figura 58. Boceto de descripción GRASIA de la interacción para formar alianzas**

Con esta especificación se añaden las posibilidades de que la proposición sea rechazada, aceptada o simplemente no comprendida. Esta última posibilidad tiene que tenerse en cuenta en situaciones en que el software que interacciona es heterogéneo, como aquí. El asistente de bolsa no sabe quién ha hecho el otro agente ni que interfaces soporta. Se asume que existe una ontología compartida por los participantes en la que se definen un conjunto de contenidos válidos de mensajes. Al término de la negociación, el solicitante recibe un certificado personalizado que servirá para certificar futuras peticiones de la alianza.

Una vez se pertenece a la alianza, se plantean negociaciones en las que los agentes que pertenecen a la alianza acuerdan nuevos parámetros de compra-venta de acciones para lograr efectos significativos en bolsa. La negociación se hace mediante un protocolo FIPA contract-net [FIPA 01]. Este protocolo se encapsula dentro de otra interacción, *NegociacionMovimientoBolsa*.

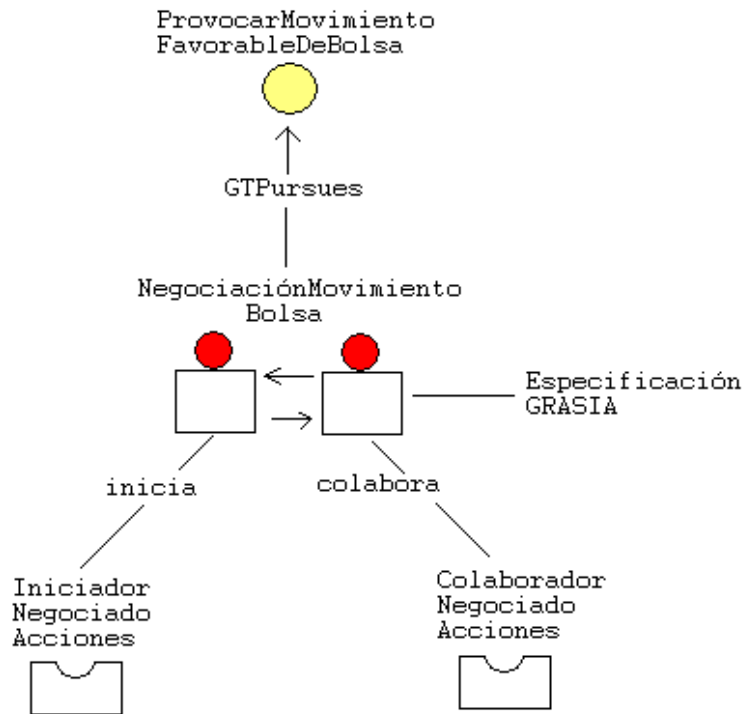


Figura 59. Interacción para conseguir movimientos en bolsa

Se definen las acciones sobre las que se va a actuar, su número y su valor. Después se acuerda un momento de actuación. Si llegado el momento se satisfacen los parámetros de actuación acordados en la negociación, entonces los agentes proceden a realizar las transacciones pertinentes con la Bolsa.

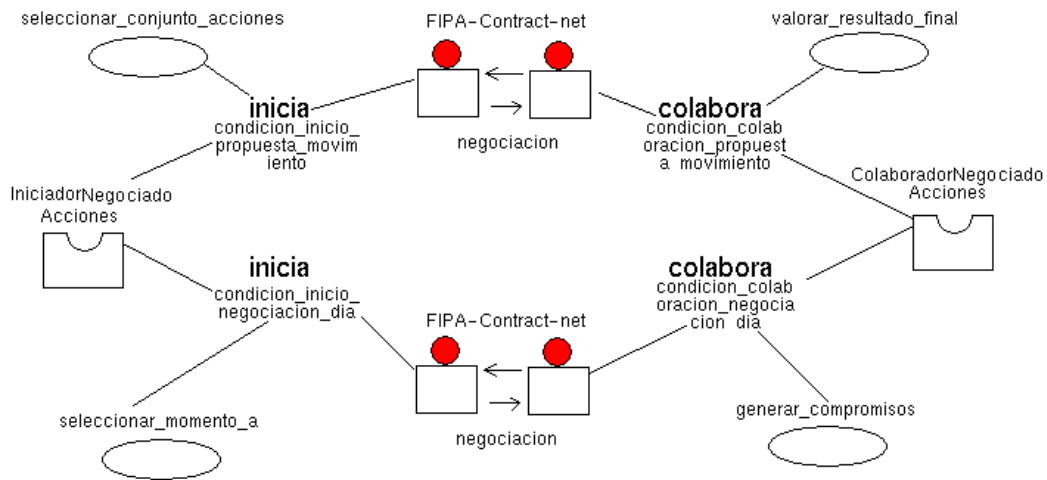
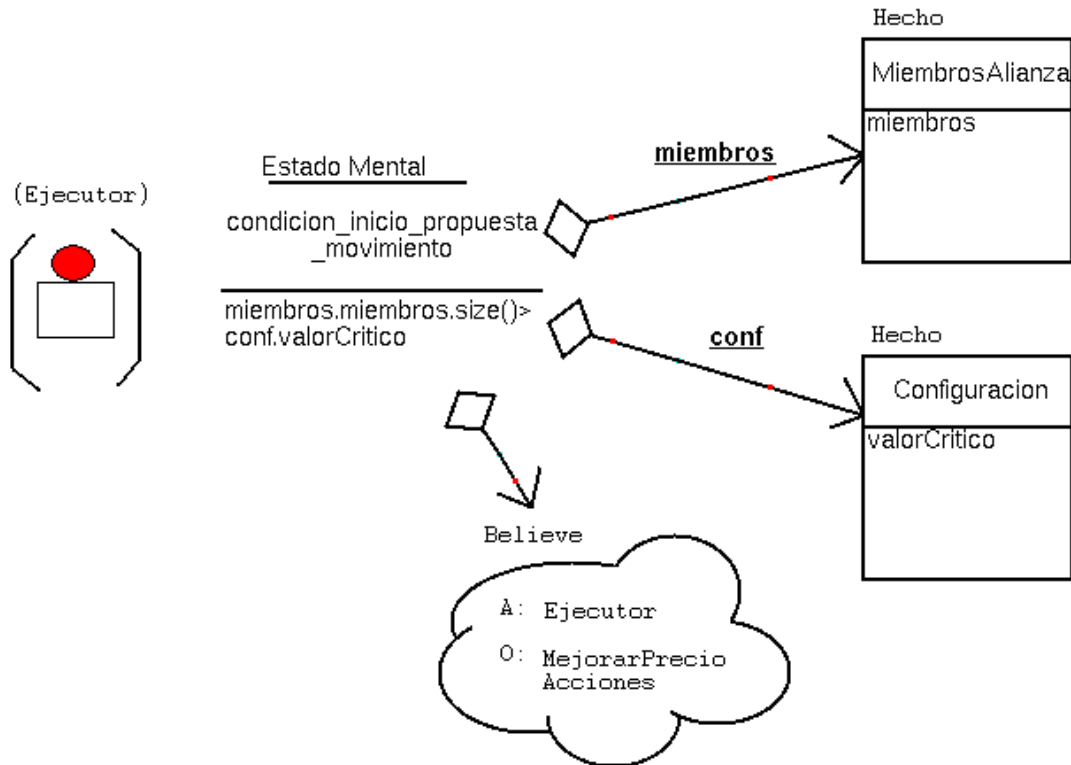


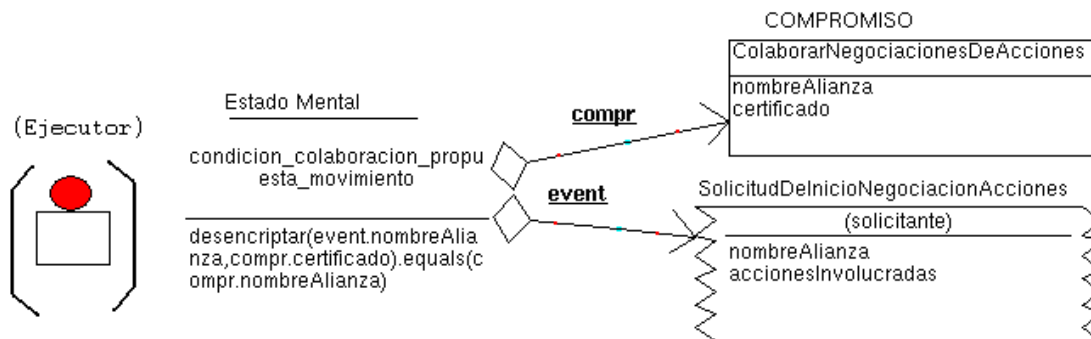
Figura 60. Descripción detallada de unidades de interacción involucradas en el acuerdo de acciones a realizar en la alianza

Para que el sistema funcione debe haber un número crítico de de agentes dentro de la alianza. Por ello se define que el iniciador de la negociación no comience hasta que se alcance un número crítico y se tenga la creencia de que se puede alcanzar el objetivo *Mejorar Precio Acciones*.



**Figura 61. Condición mental contenida en Condición inicio propuesta movimiento**

La creencia antes mencionada sería el fruto de una tarea de análisis del estado de la bolsa. El colaborador en la interacción *NegociaciónMovimientoBolsa* participa en las negociaciones en virtud del compromiso adquirido al formar una alianza (ver Figura 62).



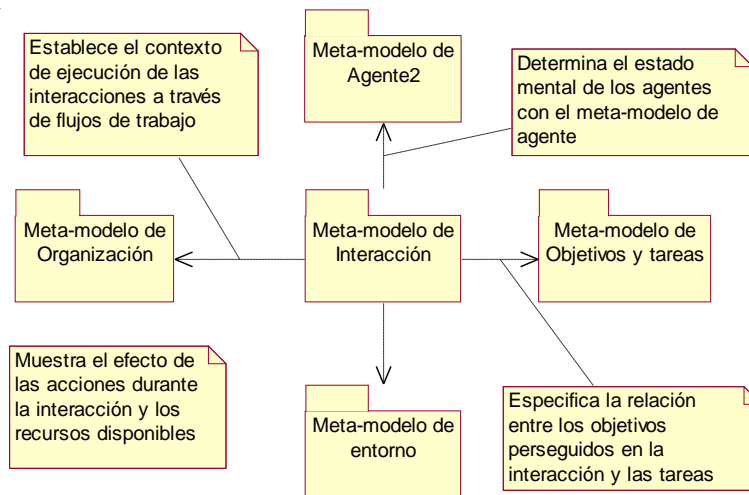
**Figura 62. Estado mental requerido por un colaborador al inicio de las negociaciones**

Para participar en una negociación para el movimiento de acciones, tiene que existir el compromiso generado en la interacción *FormarAlianza*. Este compromiso lleva implícito un certificado con el cual se puede verificar si una solicitud de participación es válida o no.

Al final de la interacción, en los agentes participantes se crea un compromiso de determinado día a determinada hora, si se cumplen las condiciones negociadas con anterioridad (umbral de beneficio dentro de los parámetros negociados y de lo configurado por el usuario), se realiza la venta o compra.

#### 4.2 Integración con otros meta-modelos

El meta-modelo de interacción constituye una fuente de información importante para determinar cómo ha de ser el control del agente. Parte del comportamiento del agente se puede ver a través del conjunto de interacciones que definen cómo reacciona el agente ante peticiones de información de otros agentes. Mediante las interacciones, además, puede deducirse cómo se modifica el estado mental de los agentes que participan, ya que las tareas ejecutadas durante la interacción son en parte responsables de los cambios internos en los agentes.



**Figura 63. Relación del meta-modelo de interacción con otros meta-modelos**

Las tareas ejecutadas en una interacción generalmente hacen uso de los recursos que configuran el entorno. La interacción, como en el caso del comportamiento del agente, también identifica los momentos en que se pueden disparar cambios en el entorno. Además, existe la posibilidad de que la propia interacción tome como medio de comunicación el propio entorno. En tal caso, el papel de los elementos recogidos en el meta-modelo de entorno es clave para reflejar qué cambios en el entorno son entendidos por otros agentes como actos de comunicación.

Finalmente, una interacción se asocia con las metas de la organización a la que pertenecen sus participantes. El contexto de la interacción se establece a través de objetivos y patrones mentales [Shoham 93]. Los objetivos se asocian directamente a las interacciones a través de la meta-relación *IPersigue* e indirectamente con las metas asociadas con la organización.

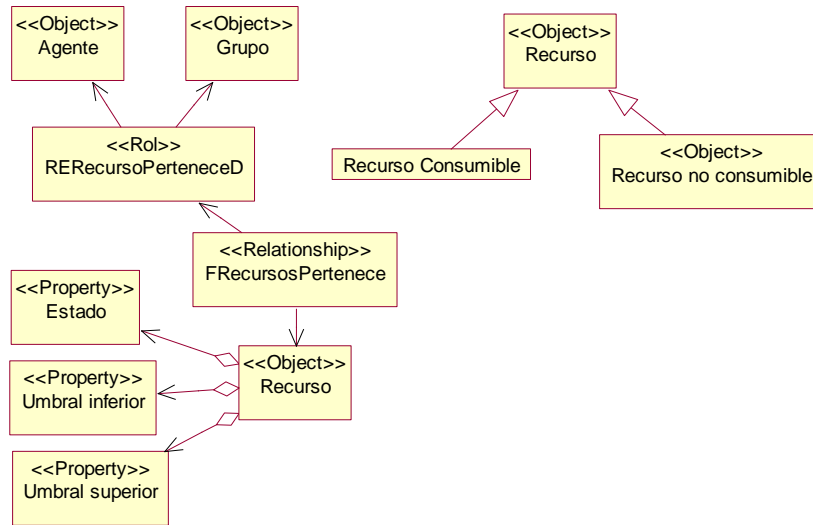
Basándose en estas relaciones, se plantean los siguientes criterios de consistencia del modelo de interacción:

- Los objetivos perseguidos por la interacción (instancias de *IPersigue*) han de pertenecer a la organización o bien estar relacionados con estos por instancias de *GTDescompone* en modelos de objetivos y tareas. La relación entre los objetivos de la organización y los de la interacción se muestran en la Figura 63. Una organización de un SMA es una entidad con identidad y propósito que agrupa juntos agentes, roles, recursos y tareas [Garijo, Gomez-Sanz y Massonet 01]. El propósito de la organización se expresa a través de objetivos que indican el motivo por el cual todos los elementos anteriores están juntos. De acuerdo con el modelo BDI (Beliefs, Desires, Intentions) [Kinny y Georgeff 97] y el diagrama de la Figura 63, los objetivos de una interacción se hallan íntimamente relacionados con los de la organización. Si los objetivos de la interacción son alcanzados, entonces los objetivos de la organización se hayan más próximos a su satisfacción.
- Todos los participantes en la interacción persiguen objetivos relacionados con los de la interacción. Un criterio para determinar si dos objetivos están relacionados sería que existieran entre ellos relaciones de descomposición dentro de un modelo de tareas y objetivos (instancias de la meta-relación *GTDescompone*) o bien por igualdad de los mismos. Otro criterio posible sería determinar si las tareas ejecutadas para satisfacer uno de ellos proporcionan evidencias que ayuden a satisfacer los otros.
- Las tareas ejecutadas a lo largo de la interacción permiten satisfacer los objetivos de la interacción. Para verificarlo, se utilizan las relaciones *GTSatisfice* del modelo de objetivos y tareas y la descomposición de objetivos (*GTDescompone*).
- Las interacciones plantean estados mentales en los agentes que se pueden expresar con modelos de agente. Sería necesario comprobar si estos estados mentales son consistentes con las acciones que los producen. El estado mental del agente surge como resultado de la actuación de su procesador de estado mental, de su gestor de estado mental, y sobre todo de las tareas ejecutadas. Así, debiera justificarse en los modelos de agente de dónde vienen las entidades mentales referidas a lo largo de la interacción y si realmente es posible que tal estado mental se dé.

## 5. META-MODELO DE ENTORNO

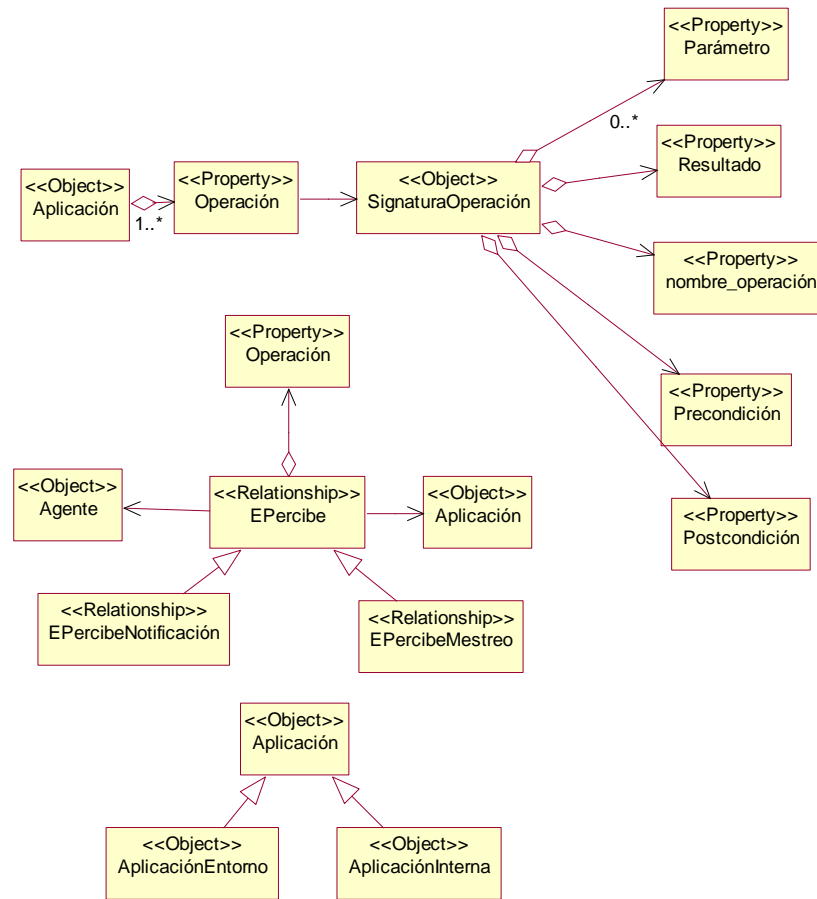
El meta-modelo de entorno restringe el tipo de elementos que van a aparecer. Se distinguen tres posibles: agentes, recursos y aplicaciones. Por recurso se entiende todo objeto del entorno que no proporciona una

funcionalidad concreta, pero que es indispensable para la ejecución de tareas y cuyo uso se restringe a consumir o restituir. Cuando el uso sea más complejo, como la funcionalidad requerida de una base de datos, se empleará el término aplicación. Por último, la denominación de agente se emplea cuando la entidad satisfaga el principio de racionalidad.



**Figura 64. Meta-modelo de entorno. Recursos.**

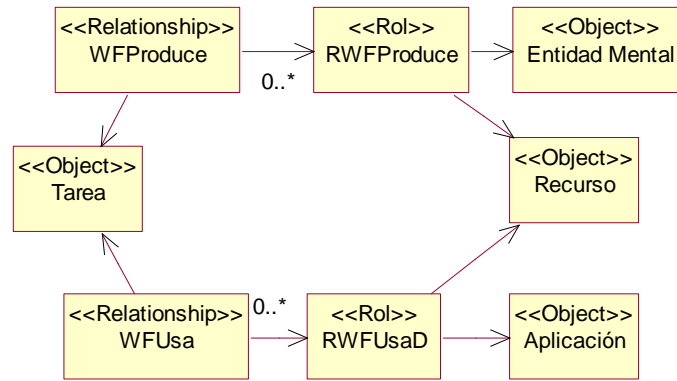
Los recursos pertenecen a un agente o a un grupo (*RERecursoPertenece*). Esta relación es similar a (*OContieneGrupo*) donde se indica únicamente si el grupo tiene un recurso. Dentro de esta relación se permite también que el recurso esté bajo el cuidado de un agente. Cada recurso se caracteriza con tres propiedades básicas: el estado actual, el umbral inferior y el umbral superior. Se ha incorporado el tipo de recurso (consumible o no consumible) mediante herencia.



**Figura 65. Meta-modelo de entorno. Aplicaciones y percepción**

Las precondiciones y postcondiciones se trasladan directamente de las especificaciones de la aplicación (las aplicaciones ya están desarrolladas así como la interfaz para interactuar con ellas) cuando esta sea un software existente. Estas operaciones se utilizan para modelar la percepción del agente. Inicialmente se distinguen sólo dos tipos: percepción por muestreo y por notificación. En ambos tipos se asocia la percepción (*EPercibe*) con una operación concreta. En el caso de muestreo, *operación* es una operación que se va a ejecutar con una frecuencia determinada. Tiene sentido cuando *aplicación* representa un dispositivo hardware que hay que muestrear. En el caso de *notificación*, se percibe únicamente si se recibe un resultado distinto del que se recibió en la última invocación.

Las aplicaciones pueden existir con anterioridad al desarrollo actual (*AplicacionEntorno*) o ser desarrolladas ad-hoc para los propósitos actuales (*AplicacionInterna*). Las primeras se obtienen de la captura de requisitos, mientras que las segundas se generan mediante técnicas convencionales de ingeniería del software.

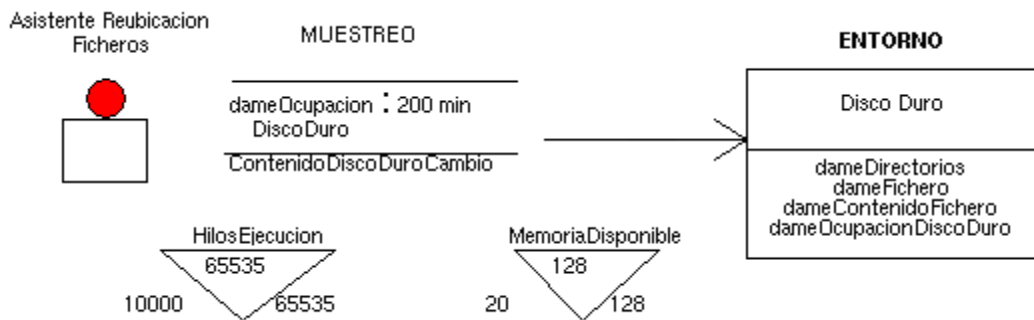


**Figura 66. Meta-modelo de entorno. Tareas, recursos y aplicaciones**

Tareas, recursos y aplicaciones se relacionan en la Figura 66. Como ya se ha visto en el meta-modelo de tareas y objetivos y en el meta-modelo de organización, la tarea consume y restituye recursos, pero también usa aplicaciones.

### 5.1 Ejemplos

Se diseña un asistente del sistema operativo que gestiona la ubicación de ficheros en el disco duro. La gestión consiste en reorganizar los ficheros automáticamente según criterios extraídos por el agente del estudio de la relación fichero-directorio en función del tipo de fichero, palabras clave presentes en el contenido del fichero o en el nombre del fichero.



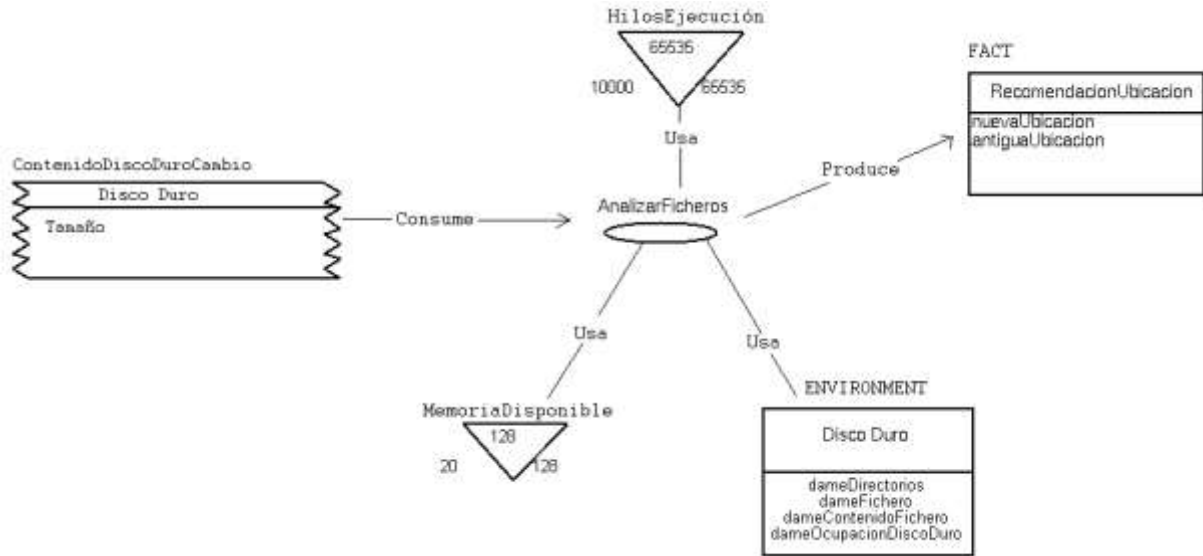
**Figura 67. Descripción de aplicaciones y recursos para el caso de estudio**

Para no afectar el funcionamiento normal del usuario, el agente atiende a la presencia de los recursos Hilos de Ejecución disponibles y Memoria Disponible. En la definición de los recursos se establecen el nivel mínimo de estos que se debe observar (10000 hilos de ejecución y 20 Mb de memoria).

El agente respeta la organización de directorios existente ayudando a ubicar los nuevos ficheros en su lugar. El agente se activa cuando se produce un cambio en el espacio de disco ocupado (ver Figura 67). La consulta



de espacio ocupado se realiza cada 200 minutos a través de la operación *dameOcupacionDiscoDuro* de una interfaz de acceso al disco duro (Disco Duro). Cuando se produce un cambio en el tamaño del disco duro, se genera un evento *ContenidoDiscoDuroCambio* según lo establecido en la relación de percepción por muestreo de la Figura 67. Este hecho posibilita la ejecución de la tarea *AnalizarFicheros* tal y como indica la Figura 68.



**Figura 68. Tarea asociada al asistente para deducir la nueva ubicación de los ficheros**

La tarea *Analizar Ficheros* utiliza los recursos *Hilos Procesador* y *Memoria Disponible* y consume el evento *ContenidoDiscoDuroCambio*. Si no hubiera suficientes recursos, la tarea simplemente no se ejecutaría. Además, utiliza la aplicación *Disco Duro* (interfaz para acceder al disco duro) para obtener información acerca de qué existe en el dispositivo de almacenamiento. Como resultado de la tarea, se obtiene un hecho *Recomendación Ubicación* donde se indica qué movimientos de ficheros son necesarios.

El asistente puede coexistir con otros para lograr la automatización de tareas del Sistema Operativo. Los asistentes se dispondrán mediante un modelo de organización como el de la Figura 69.

Los recursos de la máquina del usuario se meten en el grupo *Maquina Local*. Estos recursos son utilizados por los agentes pertenecientes al resto de grupos (*Gestión Aplicaciones*, *Gestión Ficheros*, *Gestión Red*)

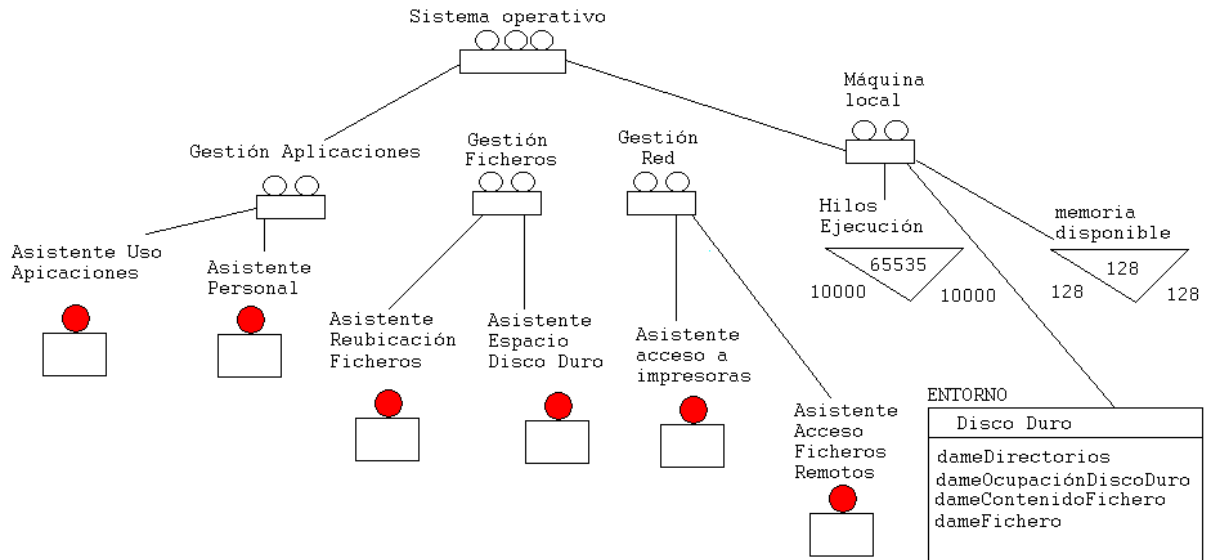


Figura 69. Organización de agentes asistentes en la máquina del usuario

## 5.2 Integración con otros meta-modelos

La principal función del meta-modelo de entorno es identificar los elementos del entorno y relacionarlos con el resto de entidades del sistema. El meta-modelo de entorno se relaciona con tres meta-modelos (Figura 70). El entorno, tal y como ha sido planteado, no aporta nada a las interacciones, aunque se vea afectado por las tareas que son ejecutadas dentro de las mismas. No obstante estas consecuencias ya se reflejan en otros modelos.

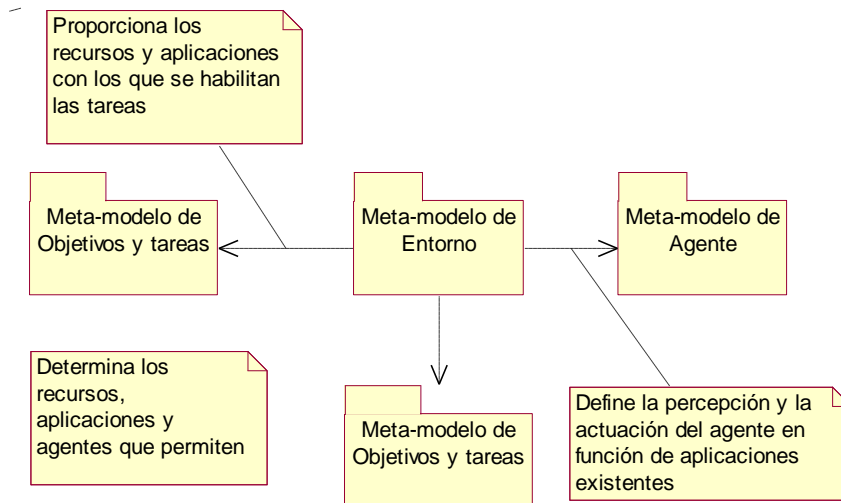


Figura 70. Meta-modelo de entorno. Relaciones con otros meta-modelos

La definición de la percepción de cada agente aporta al meta-modelo de agente. Esta percepción, como se ha visto, se expresa mediante relaciones con aplicaciones. Al meta-modelo de organización le da el conjunto de recursos, aplicaciones y agentes disponibles. Por último, el meta-modelo de objetivos y tareas obtiene del entorno el conjunto de recursos que habilitan la tarea y las aplicaciones utilizadas para expresar las acciones realizadas.

Basándose en estas relaciones, se proponen los siguientes criterios de validación:

- Todo recurso o aplicación que aparezca en los modelos de organización o tareas y objetivos, debe aparecer también en algún modelo de entorno.
- Todo agente que aparezca en algún modelo de la organización o de agente y que según los requisitos del sistema deba percibir cambios en el entorno, debe aparecer en el de entorno asociado con una aplicación.
- Las instancias de aplicación deben recoger el conjunto de operaciones descubiertas al detallar el funcionamiento de las tareas dentro de los modelos de tareas y objetivos.
- Los recursos y aplicaciones deben asignarse a un grupo o agente. Esta información se indica en un modelo de organización.

---

---

## REFERENCIAS

- [1] Malone, T. W. and Crowston, K., The Interdisciplinary Study of Coordination, ACM Computing Survey, vol. 26, no. 1, pp. 87-119, Mar.1994.
- [2] Huhns, M. S., Multiagent Systems and Societies of Agents, en Multiagent Systems. MIT Press, 2000.
- [3] Smith, R. G., The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, IEEE Transactions on Computers, vol. C-29(12) pp. 1104-1113, 1980.
- [4] OMG:MOF. Meta Object Facility (specification). Informe. 3-4-2000b
- [5] Shoham, Y., Agent Oriented Programming, Artificial Intelligence, vol. 60 pp. 51-92, 1993.
- [6] Bellifemine, F., Poggi, A. y Rimassa, G.: JADE: a FIPA2000 compliant agent development environment. Actas de conferencia. Proceedings of the fifth international conference on Autonomous agents, ACM. 2001.
- [7] FIPA: FIPA Request Interaction Protocol Specification. <http://www.fipa.org>
- [8] Garijo, F., Gomez-Sanz, J. J. y Massonet, P.: Multi-Agent System Organization. An Engineering Perspective. Actas de conferencia. MAAMAW 2001. Por publicar.
- [9] Kinny, D. and Georgeff, M., Modelling and Design of Multi-Agent Systems, Springer Verlag, 1997