PROTOTIPO DE UNA PLATAFORMA DE NEGOCIACIÓN ONLINE PARA EL MERCADO DE RESIDUOS.

TR-IIIA-2013-02

RENE MONTERO

DAVID DE LA CRUZ

PABLO NORIEGA

IIIA-CSIC

Proyecto GreenIDI.

Innpacto IPT-310000-20120-039

Contenido.

1. Preliminares.

Como paso previo a la explicación del proyecto vamos a conceptualizar y describir brevemente algunos aspectos en los que se fundamenta el desarrollo del mismo. En primer lugar abordamos el concepto de *Institución Electrónica* y posteriormente describimos parte del entorno de desarrollo (EIDE) en el cual se implementó la institución electrónica que modela el prototipo desarrollado.

1.1 Instituciones Electrónicas (IE).

Como se establece en [1]:

"[…] las interacciones humanas están reguladas por *instituciones*, que representan las reglas del juego en una sociedad, incluyendo cualquier restricción (formal o informal) que los seres humanos diseñan para articular sus interacciones. Así, la instituciones constituyen el marco en el que las interacciones humanas tienen lugar, definiendo lo que los individuos pueden y no pueden hacer y bajo qué condiciones. Las organizaciones humanas y los individuos cumplen las reglas de las instituciones a fin de legitimar sus acciones y recibir soporte legal. El establecimiento de una estructura estable para las interacciones humanas se presenta como la razón de ser de las instituciones".

Para abordar la complejidad de los sistemas multiagentes adoptamos una estrategia "mimética" con respecto al concepto general de instituciones. A continuación identificamos las nociones centrales en las que fundamentamos nuestra concepción de institución electrónica:

- Agentes y roles. Los agentes son los "jugadores" en una institución Electrónica que interactúan a través de ilocuciones (actos de habla) y los roles se definen como patrones estandarizados de comportamiento de los agentes. Mientras que los roles internos son aquellos que se establecen para mantener y garantizar las normas institucionales, los roles externos son aquellos que deben cumplir con tales normas.
- Marco Dialógico. Las instituciones electrónicas establecen los actos de habla aceptables definiendo la ontología y el lenguaje común para la comunicación y la representación del conocimiento, los cuales son agrupados en lo que denominamos el marco dialógico.
- Escena. Las interacciones entre agentes son articuladas por medio de agrupaciones de agentes que denominamos escenas, regidas por un protocolo de comunicación. Consideramos el protocolo de una escena como una especificación de los posibles diálogos que los agentes pueden mantener.
- Estructura performativa. Las escenas se conectan componiendo un flujo de trabajo (workflow) denominado estructura performativa. La especificación de una estructura performativa describe cómo los agentes pueden pasar de una

escena a otra, definiendo las precondiciones para incorporarse a una escena o abandonarla.

- Reglas normativas. En el contexto de una institución, las acciones de los agentes tienen consecuencias en forma de compromisos que obligan o restringen sus actos de habla en las escenas donde éstos actúan o actuarán en el futuro. El propósito de las reglas normativas es imponer condiciones o prohibiciones en el comportamiento de los agentes.

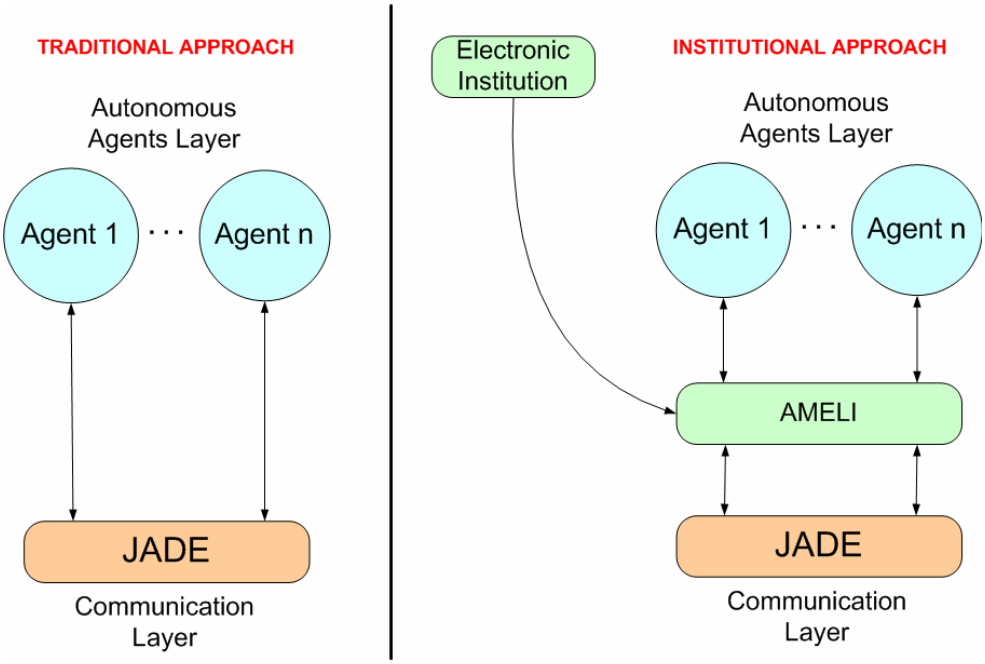


Figura 1: Mediación de agentes vía instituciones electrónicas (Fuente:[1])

"Los conceptos anteriores dibujan la estructura reguladora de una IE como un "workflow" (estructura peformativa) de protocolos multiagente (escenas) junto con una colección de reglas (normativas) que pueden ser activadas mediante las acciones de los agentes (actos de habla). En tiempo de ejecución, los agentes se incorporan y abandonan una IE, incorporándose a y abandonando escenas que son creadas y destruidas dinámicamente, y actuando mediante la realización de actos de habla. El objetivo principal de una IE es forzar el cumplimiento de las normas especificadas a los agentes participantes en tiempo de ejecución. A tal fin, todas las interacciones de los agentes son intermediadas por la IE como ilustra la figura 1. A diferencia del enfoque tradicional, que permite que los agentes interactúen abiertamente a través de una capa de comunicación, nuestra versión computacional de una IE debe ser considerada como un *middleware* social entre los agentes externos participantes y la capa de comunicación escogida encargado de aceptar o rechazar sus acciones".

1.2 Entorno de desarrollo: EIDE (Electronic Institutions Development Environment)[1] (Ver [2]).

Como se describe en [1]:

El entorno de desarrollo integrado para IEs, está constituido por un conjunto de herramientas destinadas a facilitar la ingeniería de los sistemas multiagentes como instituciones electrónicas. Los agentes *software* son la tecnología de base de nuestra aproximación. Así, las IEs encapsulan los mecanismos de coordinación que arbitran las interacciones entre agentes *software* que representan a diferentes partes como se muestra en la figura 1. EIDE permite la ingeniería tanto de instituciones electrónicas como de sus agentes participantes.

Para los fines de este trabajo, vamos a destacar las siguientes componentes del EIDE:

- **ISLANDER:** Una herramienta gráfica que permite la especificación de las reglas y protocolos en una institución electrónica (Ver [3,4]).
- **AMELI:** Plataforma *software* que permite la ejecución de instituciones electrónicas especificadas con *ISLANDER* (Ver [4]).
- **aBUILDER:** Herramienta de desarrollo de agentes (Ver [4]).

Diseño. Las instituciones electrónicas pueden ser gráficamente especificadas con la ayuda de *ISLANDER*. Éste permite la definición de una ontología común, todas las interacciones que los agentes pueden llevar a cabo y las consecuencias de tales interacciones. El resultado es una descripción precisa de los tipos y el orden de los mensajes que los agentes en un sistema multiagentes pueden intercambiar junto con una colección de normas que regulan tales interacciones. Es de hacer notar que la especificación de una IE se centra en aspectos macro (sociales), en lugar de en aspectos micro (internos) de los agentes.

Verificación. Una vez especificada una institución, debería ser verificada antes de desarrollarla y activarla para permitir el acceso a agentes externos. Este paso se realiza en dos etapas. Mientras que la primera etapa analiza las propiedades estructurales, *estáticas*, la segunda etapa analiza el comportamiento esperado, *dinámico*, de la IE.

La *verificación estática* de las IEs, funcionalidad que provee ISLANDER, consiste en la comprobación de la validez estructural de las especificaciones (por ejemplo la comprobación de que los protocolos han sido correctamente especificados).

El proceso de *verificación dinámica* de las IEs se inicia con la definición de poblaciones de agentes de diferentes características capaces de operar en la IE especificada. A fin de facilitar esta tarea, se utiliza aBUILDER, un *generador de*

[1] Entorno de desarrollado de aplicaciones para instituciones electrónicas desarrollado en el instituto de Investigación en Inteligencia Artificial (CSIC-IIIA).

agentes capaz de generar, a partir de una especificación, un esqueleto de agente dependiendo de los roles e interacciones en los que éste pueda eventualmente participar. Para obtener un agente, los desarrolladores de agentes deben completar el esqueleto generado con mecanismos de decisión.

Implantación. Una IE define un entorno normativo que articula las interacciones de agentes. Dado que los agentes pueden ser heterogéneos y tienen sus propios intereses, no podemos esperar que siempre se comporten de acuerdo a las reglas institucionales. Por lo tanto, toda IE es ejecutada vía *AMELI*, una infraestructura que intermedia y facilita las interacciones entre agentes al tiempo que obliga al cumplimiento de las reglas institucionales. *AMELI*, para validar las acciones que los agentes intentan realizar, mantiene el estado de ejecución y lo utiliza junto con la especificación. De aquí que la ejecución de una IE comience con la inicialización de *AMELI* una vez cargada la especificación. A partir de entonces, los agentes externos pueden acceder a la institución para interactuar con los otros agentes a través de *AMELI* (ver figura 1).

2 Descripción general.

El prototipo consiste en la creación de una plataforma web para proveer servicios especializados para negociar acuerdos online. Dicha plataforma ha sido concebida como un "sistema multiagentes" (ver [1] y sección 1) donde los agentes (humanos y/o de software) interactúan regulados a través de una institución electrónica (Ver [1,2] y sección 1).

La plataforma en cuestión habilita un vestíbulo permanente en el cual los usuarios pueden contactar e invitar a otros a llevar a cabo negociaciones, estableciendo de antemano: el protocolo que se va a emplear en la negociación, restricciones y características del producto a negociar. Una vez que se hayan establecido las condiciones de entrada para negociar, la plataforma será capaz de guiar la negociación propiamente dicha, validarla y registrar toda la información de las diferentes fases del proceso de acuerdo.

3 Objetivos.

- Crear un prototipo de plataforma web en el espacio de acuerdos del proyecto GreenIDI con el cual se puedan formular, negociar y establecer acuerdos on-line.
- Crear una plataforma web en el espacio de acuerdos del proyecto GreenIDI con el cual se registren contratos electrónicos obtenidos a partir de una negociación on-line.

4 Funcionalidades.

- Login
- Logout
- Visualización de agentes negociadores
- Solicitar sala de negociación para un determinado protocolo

- Invitar a negociar a otros usuarios
- Negociar según protocolo seleccionado
- Validación de acuerdos
- Registro de contratos.
- Creación de agentes de software
- Utilización de agentes de software para negociar.

5 Arquitectura básica de la plataforma.

En términos generales la plataforma está diseñada según una arquitectura de tres capas que describimos brevemente a continuación:

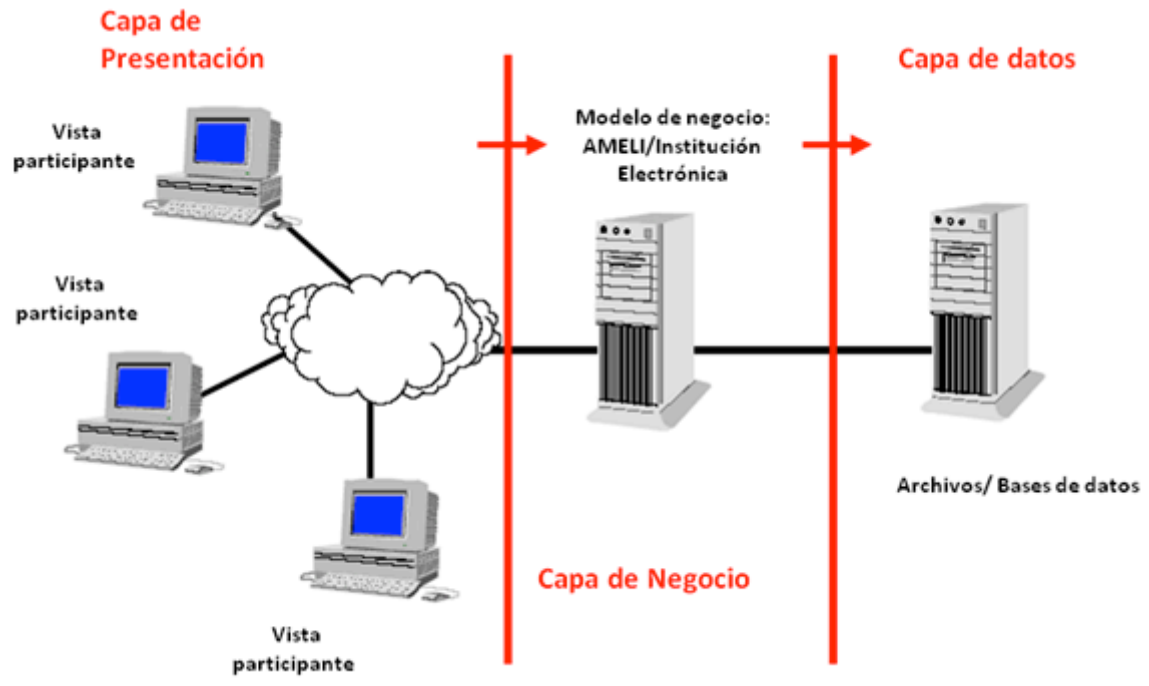


Figura 2. Modelo de tres capas para la plataforma de negociación.

a) Capa de presentación (interfaz usuaria)
b) Capa de negocio:
   La capa de negocio se implementa con una institución electrónica que se ejecuta conjuntamente con AMELI aplicación del entorno EIDE (Ver sección 1 de este documento) y [2]).
c) Capa de datos (interconexión con archivos o BD locales o externas).

6 Conexión con la plataforma GreenIDI.

En esta primera versión del prototipo, como se muestra en la figura 2, la plataforma puede enlazarse con el contexto GreenIDI como una aplicación web proveedora de un servicio especializado a la cual los usuarios autorizados (registrados) acceden para realizar negociaciones online. Aún habrá que definir, al menos: dónde y cómo se hace el registro de usuarios y cómo se distribuye la data (centralizada, local, o mixta) para: permitir el acceso a la plataforma, acceder a información de los usuarios que puede ser necesaria en las diversas fases del proceso de negociación y grabar el resultado de las operaciones realizadas ("Logs", grabar acuerdos, contratos, transacciones efectuadas, etc.).

Un aspecto que merece especial atención es la forma de acceso a la plataforma. Si consideramos la capacidad de la aplicación para incorporar tanto agentes humanos como agentes de software, los usuarios pueden acceder a ella de forma "convencional", es decir, interactuando con la aplicación a través de la interfaz usuaria para llevar a cabo sus negociaciones o también, por medio de agentes de software previamente programados para actuar como negociadores en un determinado proceso de acuerdo.

Para acceder a la plataforma a través de un agente de software, la aplicación debe recibir como entrada un archivo XML que "parametrice" los agentes, es decir, que defina la estructura y el contenido de los datos necesarios para dar "comportamiento" a dicho agente. Establecer específicamente cómo debe entrar el agente de software a la plataforma también dependerá de las decisiones que se tomen en cuanto al acceso a la aplicación y la distribución de la data en el contexto GREEN IDI.

En la sección 9 ("Parametrización de agentes- Ejemplo") se presenta una propuesta para la "parametrización" de agentes de software para la plataforma y ejemplos de archivos XML que la muestran.

7 Descripción de la institución Electrónica que implementa al prototipo.

Antes de empezar a detallar la institución electrónica, cabe recordar que tanto el modelado como la generación de código para la programación de los agentes de software se desarrollan en EIDE. Por lo tanto, primero describiremos ampliamente la fase de diseño elaborada con ISLANDER, para luego pasar a detallar la fase de implantación y construcción de agentes usando como herramienta el aBUILDER.

7.1 Especificación y modelado de la institución electrónica (Diseño).

El espacio de acuerdos está concebido como un recinto que provee los servicios necesarios para que los usuarios ("participantes") negocien sus productos a través de una serie de protocolos de negociación puestos a su disposición y con la posibilidad de fijar condiciones de entrada sobre las cuales llegar a un acuerdo. Al inicio, los participantes ingresan a un vestíbulo en el cual pueden convocar a una negociación y/o esperar a ser invitados por algún convocante. Cuando un participante propone una negociación y ésta es admitida, la institución prepara y activa toda la infraestructura y el personal necesario para que puedan realizarse las sucesivas etapas del acuerdo: sala de entrada (acceso) con su portero correspondiente, sala de negociación con un

intermediario especializado de acuerdo al protocolo seleccionado por el convocante y sala de validación con su certificador.

Cuando una estructura de acuerdo está activa, los participantes invitados recibirán una propuesta formal para negociar en la cual se le especifican: convocante, rol del convocante (comprador o vendedor), producto negociado, protocolo que se empleará para negociar. Si un participante invitado acepta, entra en el proceso de negociación y va recorriendo las diferentes salas activadas para dar servicios a cada etapa del proceso de acuerdo, hasta completarse el mismo (de manera satisfactoria o no). En caso de que el participante no acepta la invitación, permanecerá en el vestíbulo con la posibilidad de convocar o aceptar nuevas propuestas.

Es importante destacar que cada participante puede realizar varias negociaciones de manera simultánea y en cada una de ellas puede actuar como convocante o invitado, ejercer como comprador o vendedor y ofertar o demandar diferentes productos.

A continuación se presenta un cuadro resumen de la IE y posteriormente, se describen detalladamente: el marco dialógico, la estructura performativa, las escenas con sus respectivos protocolos y la ontología (funciones y tipos de datos que se han definido).

Cuadro resumen de la IE:

Tipo de rol	Rol	Escenas en las que participa	Acciones
Interno	recepcionista	Vestíbulo	Abre el vestíbulo, gestiona los requerimientos de los participantes, crea al agente portero y delega en él trabajo de crear la estructura para el proceso de acuerdo.
Externo	participante (convocante/ Invitado)	En todas las escenas	Convoca negociaciones, acepta o rechaza, invitaciones a negociar, negocia y llega a acuerdos con otros participantes. Este agente, aunque esté participando en una o varias negociaciones siempre permanece en el Vestíbulo.
Interno	portero	Vestíbulo, subestructura performativa ProcesoAcuerdo, escena Acceso	Entra en el Vestíbulo a solicitud del recepcionista con el fin de que abra y gestione la subestructura performativa de un proceso de acuerdo ("ProcesoAcuerdo"). En esta subestructura, dependiendo del protocolo, el portero crea un agente intermediario que estará presente en tantas escenas de negociación como sean necesarias abrir y también un agente certificador. Además recibe del convocante toda la información necesaria (producto, la lista de invitados o criterio de invitación, ETC) y la trasmite al heraldo. Adicionalmente, gestiona la entrada al proceso de acuerdo.
Interno	Intermediario	Subestructura performativa ProcesoAcuerdo, escenas: Acceso, Negociación	En Acceso entra a petición del portero para abrir una o varias escenas de Negociación. En ésta, hace todas las labores que les corresponden de acuerdo al protocolo.
Interno	certificador	Subestructura performativa ProcesoAcuerdo, escenas: Acceso y Validación	En Acceso entra a petición del portero para abrir una escena de Validación. En cada una de estas, hace todas las labores que les corresponden de acuerdo al protocolo.
Interno	heraldo	Vestíbulo, Sub-estructura performativa ProcesoAcuerdo, escenas: Acceso.	Entra a la escena Acceso a petición del portero, éste le comunica la lista de invitados y otros datos sobre el proceso de acuerdo. Posteriormente, sale de Acceso y se dirige al Vestíbulo con el fin de dejar la invitación formal al proceso a cada uno de los invitados.

Tabla 1. Cuadro resumen de la IE.

Marco dialógico.

Se definen los agentes (internos y externos) que actúan dentro de la institución electrónica así como también las partículas ilocutorias para todos los actos de habla. Cabe destacar que el agente participante puede ejercer tres roles: participante, convocante e invitado. Las relaciones "ssd" ("Static Separation of Duty") indican que un agente no puede ser al mismo tiempo los roles vinculados. Es decir, que un agente no puede ser al mismo tiempo convocante e invitado. Tampoco puede ser al mismo tiempo recepcionista y portero, recepcionista y heraldo, etc.

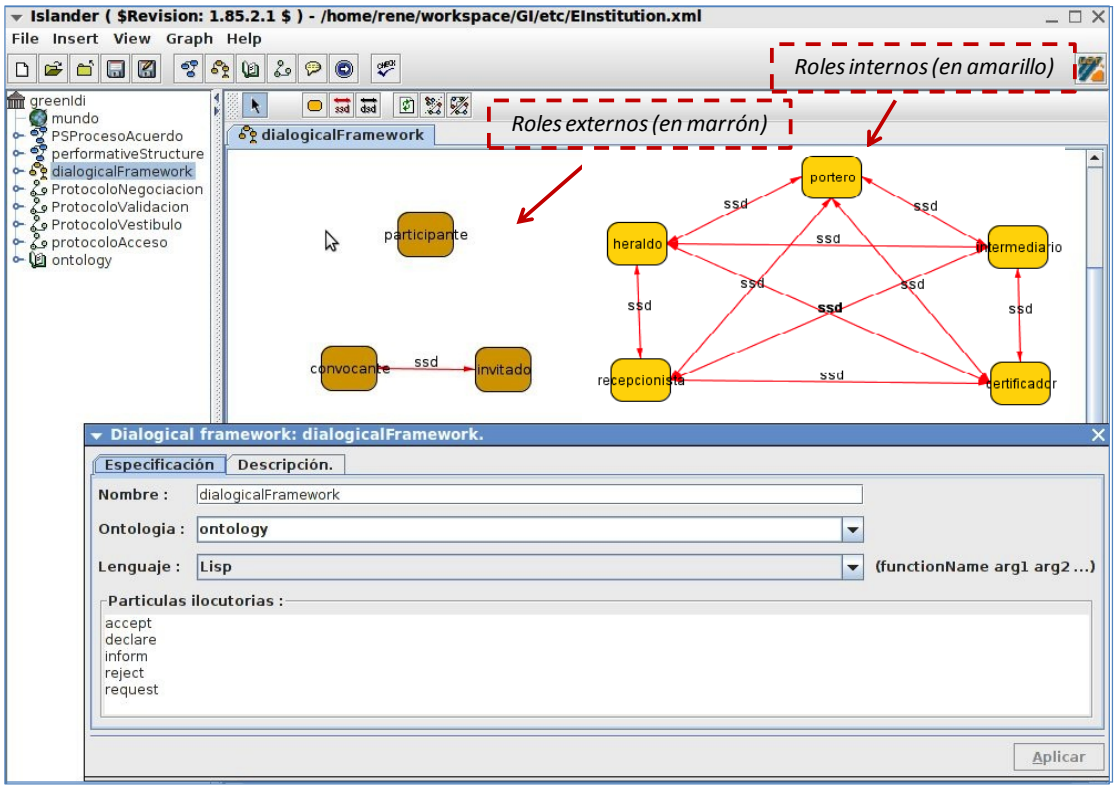


Figura 3. Marco Dialógico.

Estructura Performativa:

Los elementos que componen la estructura performativa son escenas y transiciones (representadas por conectores "AND", "OR" y "XOR"). Mientras que las primeras articulan interacciones entre los agentes, con las transiciones se sincronizan las acciones de un agente consigo mismo o con otros agentes. Las transiciones son los puntos en los cuales se verifica la trayectoria de los agentes dentro de la estructura performativa.

La estructura performativa consta de dos escenas principales: Vestíbulo y ProcesoAcuerdo[2] (es una subestructura performativa que contiene las escenas: Acceso, Negociación y Validación). La escena Vestíbulo es creada por el recepcionista y queda lista para recibir en todo momento a los agentes externos (participantes). En dicha escena, cada vez que un participante (actuando como convocante) hace un requerimiento de negociación, el recepcionista, solicita los ***servicios externos de la institución***[3] para hacer ingresar al Vestíbulo a otro agente interno, el portero, que será el responsable de crear la escena ProcesoAcuerdo.

Una vez creada la escena ProcesoAcuerdo el portero también crea la escena Acceso y en ella solicita los ***servicios externos de la institución*** para hacer ingresar tanto al intermediario, que creará y manejará las escenas de negociación que hagan falta de acuerdo con el protocolo, así como también, al certificador, para que cree y gestione la escena de Validación. De esta forma queda completamente activada y enlazada la estructura para llevar a cabo un proceso de acuerdo.

[2] El modelado de las escenas admite, además de protocolos, estructuras performativas "hijas". En este caso estamos llamando subestructura performativa a "ProcesoAcuerdo" para distinguirla de la estructura performativa "padre".

[3] Llamamos "servicios externos de la institución" a un conjunto de servicios definidos en EIDE que pueden ser invocadas por los agentes cuando la institución ya se encuentra en ejecución. En este caso, invocamos el servicio para hacer entrar un agente interno (institucional).

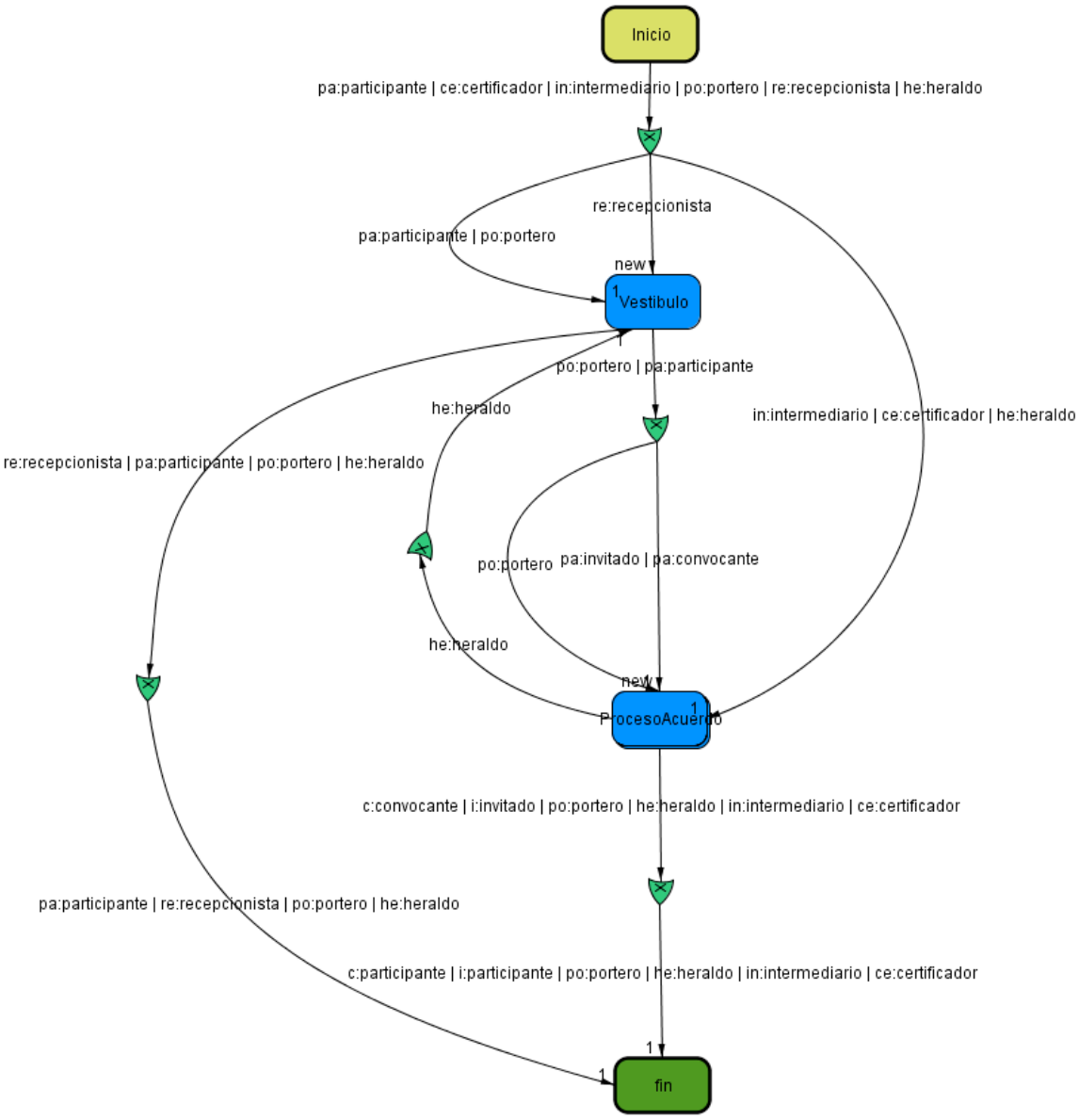


Figura 4. Estructura performativa.

Una vez que la subestructura de acuerdo se encuentra completamente conformada, un participante puede convocar desde el Vestíbulo a un proceso de negociación. Este convocante se trasladará a la subestructura ProcesoAcuerdo, concretamente a la escena de Acceso y comunicará al portero la información necesaria para que tenga lugar la negociación (plazo de respuesta a las invitaciones, restricciones impuestas por el protocolo) y esperará allí hasta que se den las condiciones para continuar, es decir, continuar a las escenas de Negociación y Validación.

En la escena Vestíbulo (siempre activa) tiene lugar el acto de comunicar las invitaciones. El heraldo trasmite el mensaje a cada invitado y éste decide si participa o no en la negociación. Si el participante acepta la invitación, pasará a la sala de Acceso correspondiente y continuará la negociación según su conveniencia.

Cada proceso de acuerdo es una subestructura con tres escenas independientes: Acceso, Negociación y Validación, que estarán activas en cada una de las etapas de los procesos de negociación. Cuando acaba un proceso de acuerdo la subestructura correspondiente desaparece. Cabe destacar que puede haber varios procesos de negociación abiertos de manera simultánea.

Descripción de las Escenas.

- Vestíbulo: Es creada por el agente de rol recepcionista y participan en ella los roles internos: portero y heraldo; y el rol externo participante. Es el lugar donde se inician las propuestas de negociación, también es la escena en la cual se da inicio a la creación de la subestructura performativa ProcesoAcuerdo y donde, a la postre, tienen lugar las invitaciones a negociar. El participante (convocante) solicita la apertura de un proceso (indica protocolo, condiciones del protocolo y rol que desempeñará en la negociación) al recepcionista, éste autoriza la creación del proceso (o no) y solicita a la institución el ingreso del portero, agente que será el encargado de verificar las condiciones de entrada a un proceso de acuerdo. El portero entra al vestíbulo y desde allí el recepcionista le comunica que debe crear y gestionar una subestructura para el proceso de acuerdo solicitado (ProcesoAcuerdo), la crea y le informa al recepcionista el identificador de la subestructura que ha creado para luego abandonar el Vestíbulo. Posteriormente, el recepcionista comunica al convocante que el Acceso al proceso que solicitó ya está listo y el convocante se marcha (a la vez que permanece) del Vestíbulo a la subestructura de ProcesoAcuerdo, concretamente a la escena Acceso, donde comunicará al portero el producto a negociar y bajo qué criterio y con qué parámetros se harán las invitaciones al proceso de acuerdo.

  Los participantes presentes en el Vestíbulo recibirán las invitaciones a un determinado proceso de acuerdo a través del agente heraldo asignado a dicho proceso (habrá tanto heraldos como procesos abiertos haya). Dentro de los parámetros de la invitación se incluye el tiempo límite para que el invitado dé su respuesta de aceptación. Si transcurrido el tiempo indicado no se producido una respuesta, se supone que el invitado no acepta participar en el proceso de acuerdo.

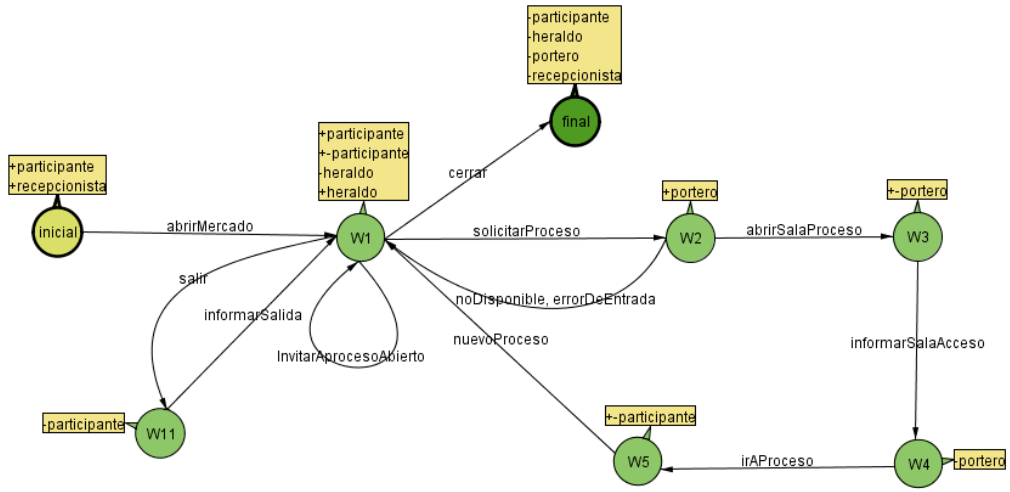


Figura 5. Protocolo del Vestíbulo.

Etiqueta	Ilocución
abrirMercado	(declare (?re recepcionista) (all all) (abrirMercado))
solicitarProceso	(request (?pa participante) (?re recepcionista) (procesoDeAcuerdo ?protocolo ?condiciones ?rolConvocante))
noDisponible	(declare (!re recepcionista) (!pa participante) (noDisponible))
errorDeEntrada	(reject (!re recepcionista) (!pa participante) (errorDeDatos ?msjerror))
abrirSalaProceso	(inform (!re recepcionista) (?po portero) (abrirNegociacion ?convoc ?tipoProtocolo ?condicionesProt ?rolConv))
InformarSalaAcceso	(inform (!po portero) (!re recepcionista) (salaDeAcceso ?numSalaAcceso))
irAProceso	(declare (!re recepcionista) (!pa participante) (irAProceso !numSalaAcceso))
nuevoProceso	(declare (!re recepcionista) (all all) (nuevoProceso))
InvitarAProcesoAbierto	(declare (?he heraldo) (?pa participante) (invitarAProceso ?convocante ?rolConvocante ?producto ?protocolo ?condiciones_prot ?salaAcceso ?horaLimite))
InformarSalida	(inform (!re recepcionista) (all all) (avisoDeSalida ?nomAgente))
salir	(inform (?pa participante) (!re recepcionista) (salir))
cerrar	(declare (!re recepcionista) (all all) (cerrar))

Tabla 2. Acciones de Arco en Protocolo del Vestíbulo.

- Subestructura performativa ProcesoAcuerdo.
-

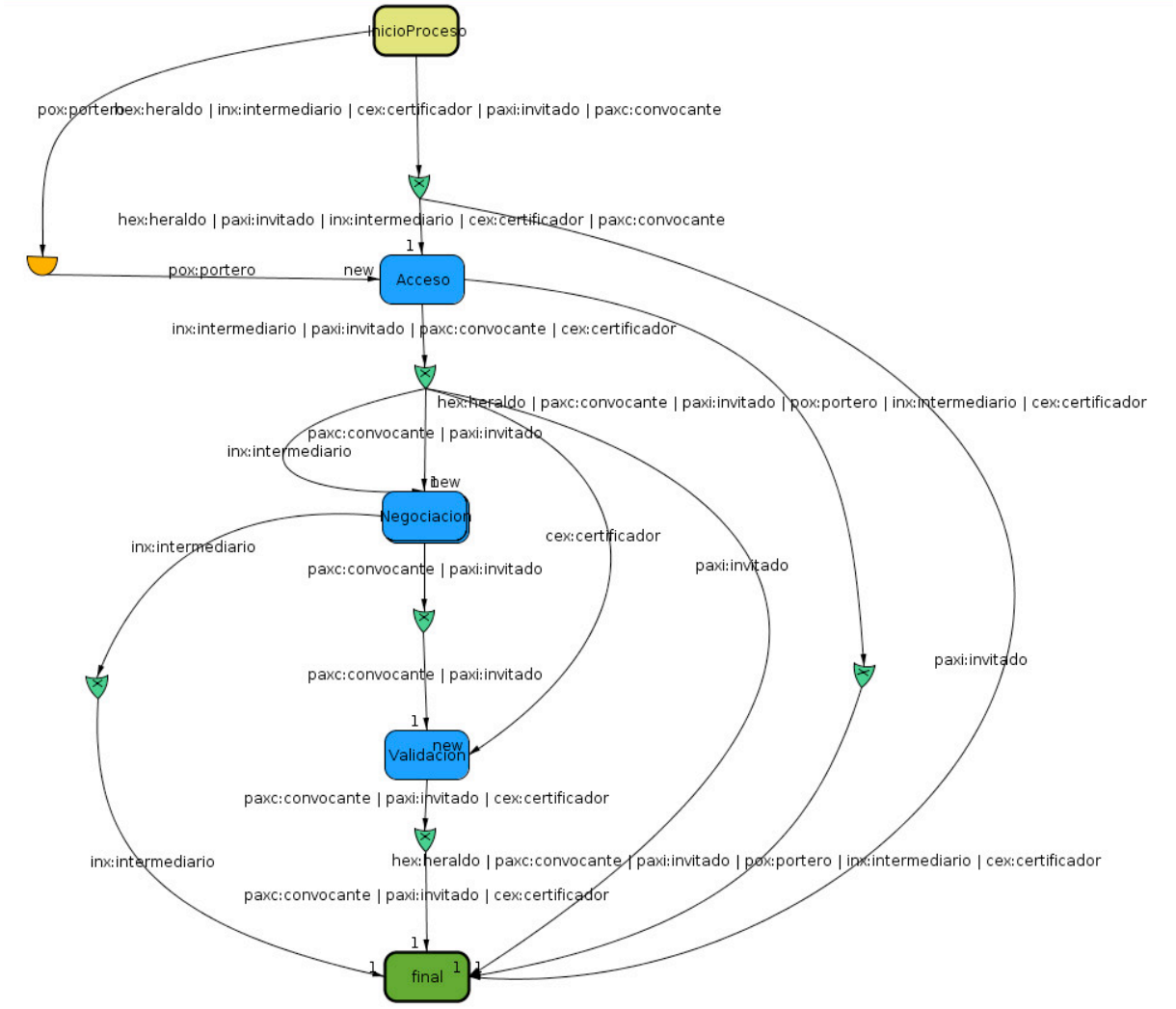


Figura 6. Sub-estructura performativa "ProcesoAcuerdo".

- Acceso: Es creada por el agente de rol portero e intervienen agentes con roles de participante, intermediario, certificador y heraldo. El portero abre la escena conociendo el protocolo de negociación y sus condiciones, rol del convocante y, al iniciar la misma, solicita al convocante la información del producto a negociar y bajo qué criterio y con qué parámetros se harán las invitaciones al proceso de acuerdo. Los parámetros dependerán del protocolo (número mínimo de participantes la negociación, precios de reserva,…) y siempre tendrán como componente obligatorio, el intervalo en el cual se espera una respuesta de aceptación a la invitación efectuada. El criterio de negociación puede ser algo semejante a: "todos los x que compren el producto", "algún x que venda el producto y"…. Con este criterio el portero iniciará un proceso de búsqueda (en una BD, por ejemplo) que dará como resultado una lista de invitados. El criterio también puede indicar que se trata simplemente de una lista de nombres, con lo que no tendría lugar el proceso de búsqueda porque ya tenemos la lista directamente.

El portero, conociendo el protocolo de negociación, continúa sus tareas solicitando los ***servicios de la institución*** para dar entrada a los agentes internos necesarios para llevar a cabo todas las fases del proceso de acuerdo: intermediario, certificador y heraldo. El agente intermediario, irá a crear (a la vez que permanece en la escena Acceso) las escenas de negociación que exige el protocolo y después de crearlas, comunica al portero la identificación de cada una de ellas. En el caso del agente certificador, también va a crear su escena de Validación e informa al portero la identificación de la escena correspondiente.

El portero, ahora, teniendo toda la información necesaria (información propia de la negociación y de la infraestructura de la institución electrónica): convocante, rol del convocante, producto a negociar, protocolo, lista de invitados, identificación de salas de negociación y certificación, hora límite de respuesta a la invitación; la comunica al agente heraldo y éste va al vestíbulo a dejar las invitaciones a cada participante convocado.

Una vez que acaba el tiempo establecido (timeout) para que respondan los invitados, los participantes que tomaron la decisión aceptar, saldrán de la escena Vestíbulo y entrarán a Acceso. En este punto el portero decidirá (en base a la lista de invitaciones) si deja pasar o no a cada uno de los aspirantes a entrar en la negociación e informa al convocante la lista definitiva de invitados que aceptan negociar y la identificación de las salas de negociación activadas. Con esta información, el convocante sale del Acceso y va a la(s) de negociación correspondientes.

Después, el portero, tomando como base las condiciones que previamente había establecido el convocante (mínimo de participantes, la presencia de un invitado en particular) decide si abrir o cancelar la negociación, e informa su decisión. En el caso de que haya decidido abrir la negociación, el mensaje irá dirigido a todos y cada uno de los invitados, que saldrán de Acceso e irán a la sala de negociación que les ha sido asigna. Si no se abre la negociación, todos los agente reciben el mensaje y se cierra la escena. Los agentes intermediario y certificador, en caso de recibir la información de cancelación de la negociación, efectuarán el cierre de las escenas que ya estaban activas (Negociación y Certificación, respectivamente) para llevar a cabo el proceso de acuerdo.

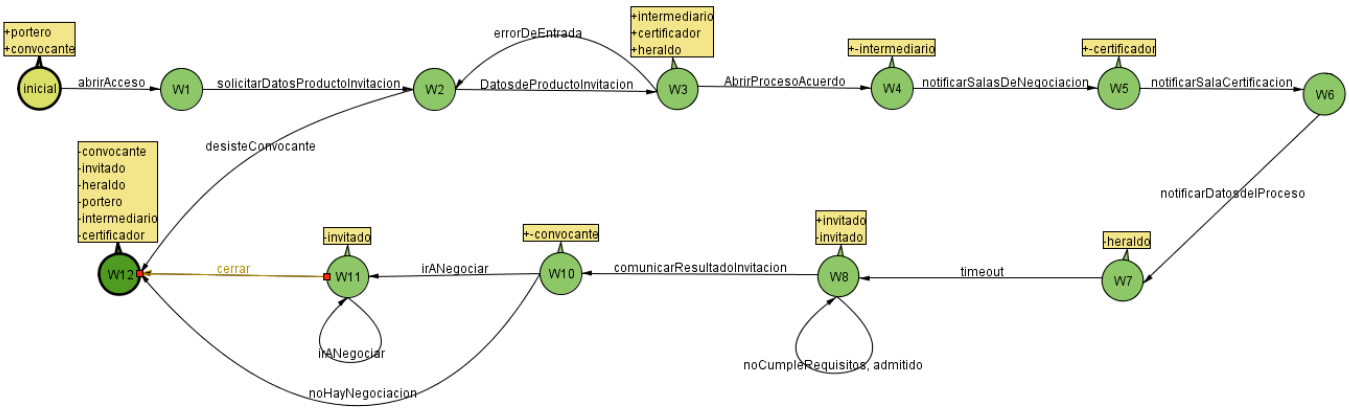


Figura 7.Protocolo de Acceso.

Etiqueta	Ilocución
abrirAcceso	(declare (?po portero) (all all) (abrirAcceso))
solicitarDatosProductoInvitacion	(declare (!po portero) (?pc convocante) (completarDatos))
desisteConvocante	(declare (!pc convocante) (all all) (noHayNegociacion))
datosdeProductoInvitacion	(inform (!pc convocante) (!po portero) (datosProductoInvitacion ?producto ?criterio ?listainvitados ?tiempoDeEspera))
errorDeEntrada	(reject (!po portero) (!pc convocante) (errorDeDatos ?msjError))
abrirProcesoAcuerdo	(declare (!po portero) (?in intermediario) (abrirSalasNegociacion ?numDeSalas ?tipoAgente))
notificarSalasDeNegociacion	(declare (?in intermediario) (!po portero) (salaNegociacion ?listSalaNeg))
notificarSalaCertificacion	(inform (?ce certificador) (!po portero) (salaCertificacion ?salaCert))
notificarDatosDelProceso	(declare (!po portero) (all all) (datosDelProceso ?dpro))
timeout	[!tiempoDeEspera]
aceptado	(inform (!po portero) (?inv invitado) (noAceptar))
noAceptado	(inform (!po portero) (?inv invitado) (aceptar))
ComunicarResultadoInvitacion	(declare (!po portero) (!pc convocante) (aceptanNegociar ?listaParticipantes !listSalaNeg))
irANegociar	(inform (!po portero) (?inv invitado) (irANegociacion ?numSalaNeg ?numSalaCert))
noHayNegociacion	(declare (!po portero) (all all) (noHayNegociacion))
cerrar	(declare (!po portero) (all all) (cerrar))

Tabla 3. Acciones de arco para Protocolo de Acceso.

- Negociación[4]: es la escena en la cual se lleva a cabo la negociación propiamente dicha y dependerá del protocolo especificado por el convocante. Es creada por el agente con rol intermediario que puede especializarse de acuerdo al protocolo. Por ejemplo, para el caso de subastas, subastador; para el caso de una negociación a sobre cerrado, mediador; en el caso de mediación, facilitador, ETC.

  En esta escena, además del intermediario, intervienen los agentes con rol de participantes (convocante e invitados). El intermediario da inicio a la escena y va solicitando y procesando la información necesaria que proporcionan tanto el convocante como los invitados, hasta acabar el proceso (no necesariamente en un acuerdo). Independientemente de que el proceso de negociación acaba en un acuerdo, los participantes salen de Negociación y entran a la escena de Validación.

- Protocolo de subasta a sobre cerrado con precio de reserva. Participan los roles: intermediario, convocante e invitado. Al inicio, el intermediario, puede o no abrir la negociación. Si no lo hace es porque no se ha producido la entrada de los agentes mínimos necesarios para que el proceso se lleve a efecto. Si decide hacerlo (porque se cumplen las condiciones mínimas requeridas), se abre la negociación informando a todos los agentes presentes: los nombres de los participantes y con qué rol actúan

[4] La escena Negociación ahora tiene un solo protocolo (subasta a sobre cerrado con precio de reserva), no obstante, le pueden ser agregados tantos como se deseen (Ver sección 8).

(compradores o vendedores), el producto que se oferta o se compra y el nombre del convocante. Después de dar esta información general, el intermediario indica a los participantes que ha comenzado el intervalo para hacer pujas. Cada invitado envía su oferta al intermediario y este si la oferta es aceptable en términos del precio de reserva (si se trata de una compra, se fija un precio mínimo de venta; si, en cambio, estamos ante una venta, se fija un máximo para comprar). Una vez transcurrido el intervalo el intermediario evalúa si hay suficientes ofertas que satisfacen las condiciones, determina si hay un ganador (escogerá la máxima oferta para una venta y la mínima para una compra) y lo comunica a todos los agentes en la escena. En caso de que no haya ofertas aceptables, el intermediario igualmente hace la comunicación general de que no ha habido acuerdo.

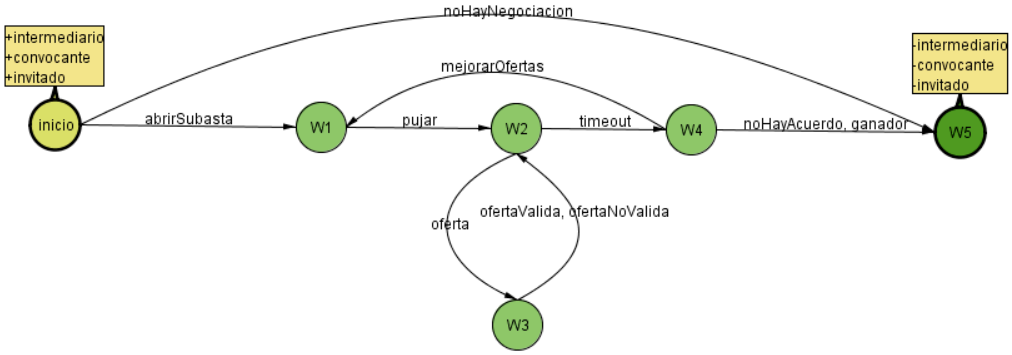


Figura 8. Protocolo de Negociación (Subasta a sobre cerrado).

Etiqueta	Ilocución
abrirSubasta	(declare (?m intermediario) (all all) (abrirSubasta ?venden ?compran ?producto ?convoca))
pujar	(declare (!m intermediario) (all invitado) (pujar ?waitTime))
oferta	(declare (?in invitado) (!m intermediario) (oferta ?of))
ofertaValida	(declare (!m intermediario) (!in invitado) (msgMediador ?st))
ofertaNoValida	(declare (!m intermediario) (!in invitado) (ofertaInvalida))
timeout	[waitTime]
mejorarOfertas	(declare (!m intermediario) (all invitado) (msgMediador ?mo))
noHayAcuerdo	(declare (!m intermediario) (all all) (noHayAcuerdo ?razon))
ganador	(declare (!m intermediario) (all all) (ganador ?nombre ?antiguoPropietario ?prd ?precioAcuerdo))
noHayNegociacion	(declare (?in intermediario) (all all) (noAbrir))

Tabla 4. Acciones de arco para Protocolo de Negociación (Subasta a sobre cerrado).

- Validación: La crea el agente con rol de certificador y entran los agentes participantes (convocante e invitados). En esta escena, el certificador pregunta al convocante si hubo un acuerdo en la negociación, si es así, el convocante le comunica los términos del acuerdo alcanzado (ver tipo de dato Acuerdo). A continuación, el certificador envía a todos los participantes en la escena el "código de validación" (un "string "que puede confeccionarse a conveniencia) y se cierra la escena y el proceso de acuerdo. En caso

de que no se haya podido llegar a un acuerdo en la escena de negociación, el convocante sólo dirá al certificador que no hubo acuerdo y éste no hace nada más que comunicar a los participantes el cierre de la escena. Con el fin de la escena de validación, también se cierra la correspondiente sub-estructura performativa ProcesoAcuerdo.

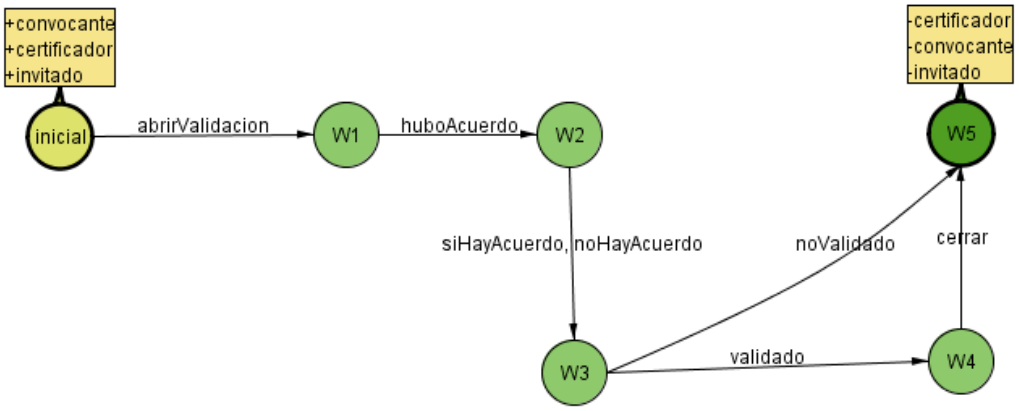


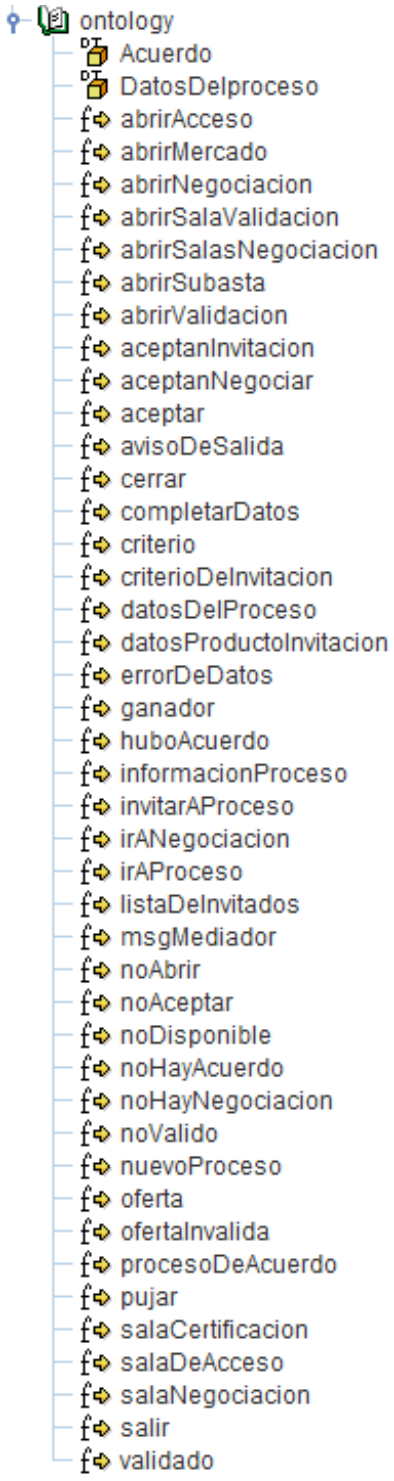
Figura 9. Protocolo de Validación.

Etiqueta	Ilocución
abrirValidacion	(declare (?ce certificador) (all all) (abrirValidacion))
huboAcuerdo	(declare (?ce certificador) (?co convocante) (huboAcuerdo))
siHayAcuerdo	(inform (!co convocante) (!ce certificador) ?acuerdo:Acuerdo)
noHayAcuerdo	(inform (!co convocante) (!ce certificador) (noHayAcuerdo ?msj))
NoValidado	(declare (!ce certificador) (all all) (cerrar))
validado	(declare (!ce certificador) (all all) (validado ?codigoValidacion))
cerrar	(declare (!ce certificador) (all all) (cerrar))

Tabla 5. Acciones de arco para Protocolo de Validación.

Ontología[5]*:*

- Tipos de datos y funciones:



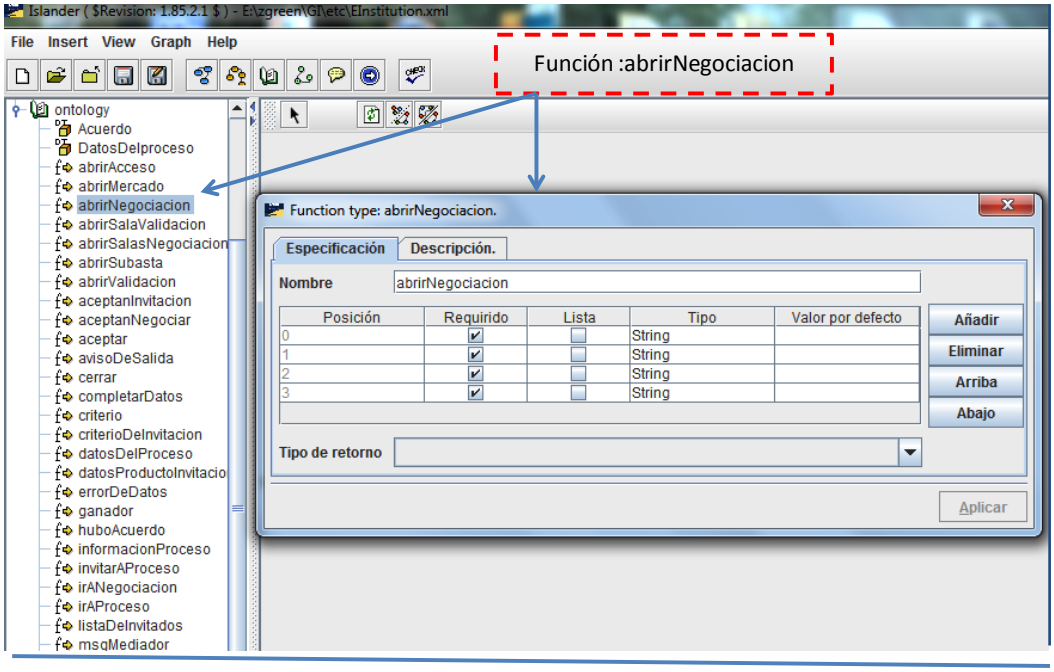
[5] La ontología se va construyendo en la medida en que se definen los actos de habla entre los agentes. Se define el nombre de cada función y los parámetros asociados.

- Contenido de la función “abrirNegociacion”.

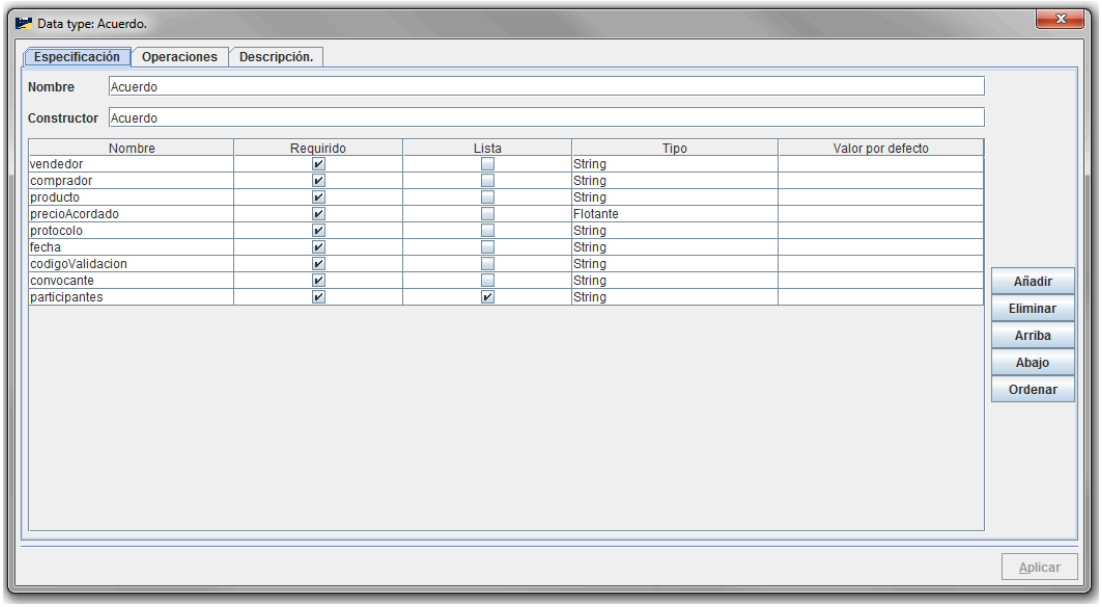
  Ejemplo de uso de la función “abrirNegociacion” (Ver figura 5, Tabla 2) en el protocolo Vestibulo:

```
(inform (!re recepcionista) (?po portero)
    (abrirNegociacion ?convoc ?tipoProtocolo ?condicionesProt ?rolConv))
```

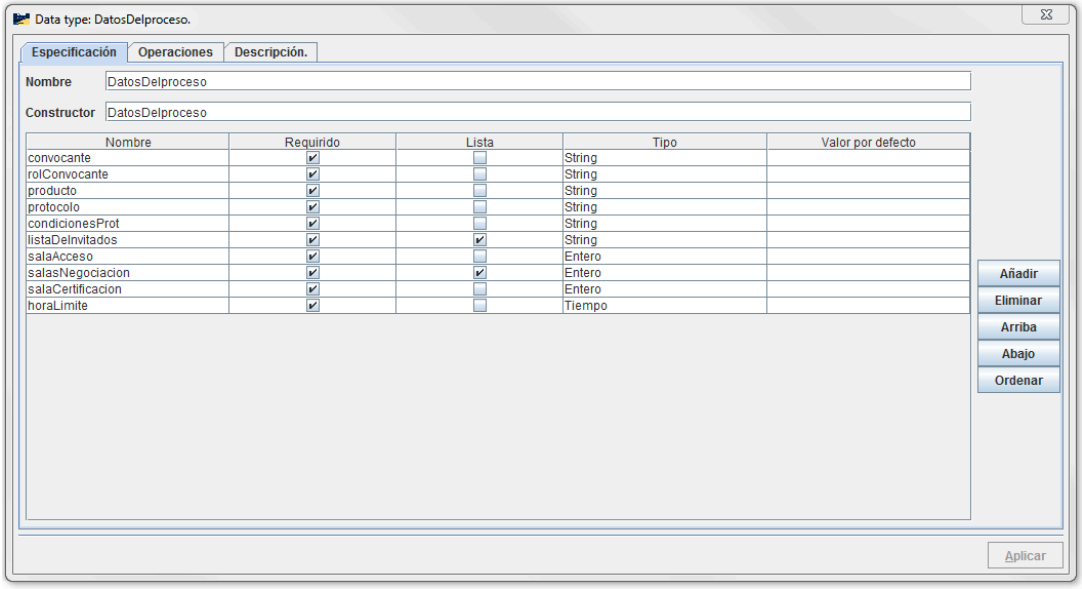
  Descripción de la función dentro de la ontología:



- Tipos de datos.
  - Acuerdo:



- DatosdelProceso:



7.2 Aplicación de software de la institución electrónica (Implantación).

En este punto, ya tenemos completada la especificación elaborada y verificada estáticamente con ISLANDER. Éste, guarda todas las características de la especificación de la IE un archivo XML.

A partir de ahora tenemos que realizar las siguientes tareas:

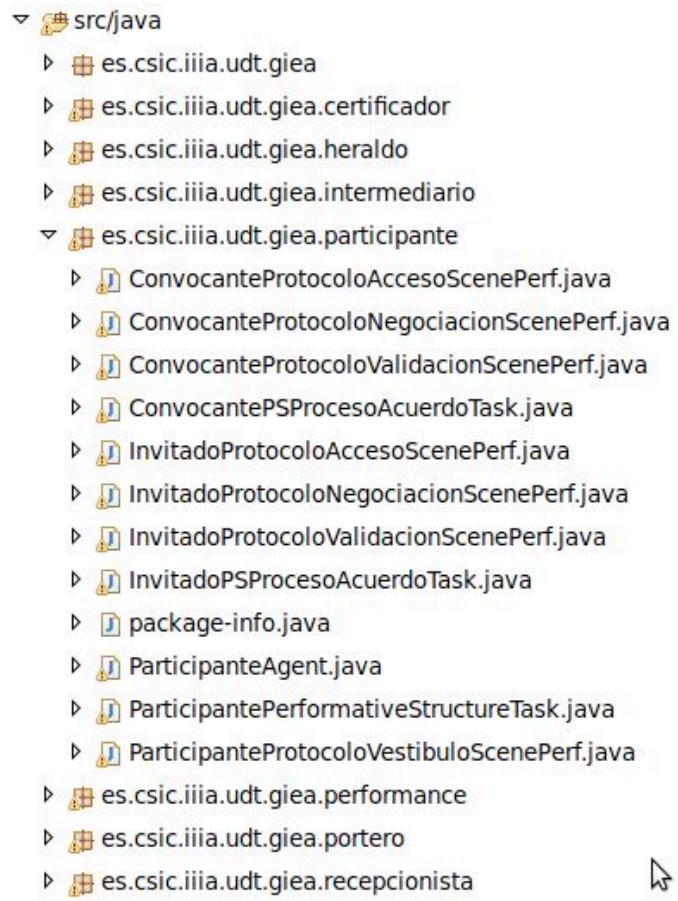
i. Generación de código de los agentes con eBUILDER. Éste, usando como entrada el archivo XML que construye ISLANDER, genera un proyecto en java con el código

que corresponde al esqueleto básico de comportamiento (movimientos y mensajes) de todos los agentes dentro de la institución.

El proyecto en java que genera el aBUILDER tiene una estructura como la siguiente:

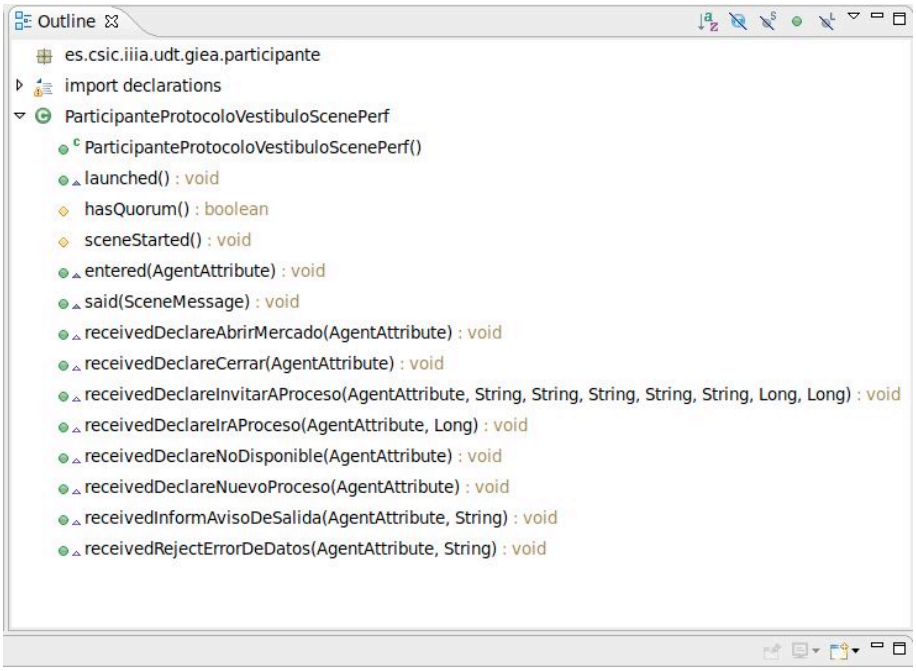
Como puede observarse, en el proyecto "GI" cada uno de los agentes que actúan dentro de la IE tiene un paquete asociado donde reside el software que corresponde a sus movimientos en la estructura performativa de la IE y sus acciones dialógicas dentro de cada protocolo.

Por ejemplo, si abrimos el paquete correspondiente al agente "participante" podemos encontrar que en su interior contiene todas las clases que permiten programar las características del agente propiamente dicho (ParticianteAgent.java) así como las acciones para cada rol (Convocante o Invitado) que desempeña en la estructura performativa y en cada uno de de los protocolos (Vestíbulo, Acceso, Negociación, Validación) donde participa:



Cada una de las clases contiene los métodos que permiten insertar el código necesario para programar las acciones dialógicas especificadas para los agentes en cada protocolo. Por ejemplo, si detallamos la clase:

"ParticipanteProtocoloVestibuloScenePerf.java":



Podemos comprobar que existe un método para responder a cada una de las acciones de habla que han sido especificadas entre el agente "participante" (con rol de participante) y otros agentes que actúan en el protocolo. Por ejemplo, el método "receivedDeclareAbrirMercado" deberá contener el código correspondiente a las acciones que debe llevar a cabo el agente "participante" cuando reciba el mensaje "abrirMercado" emitido por el agente "recepcionista" en el protocolo Vestíbulo (Ver figura 5 y tabla 2). También podemos ver otros métodos que son propios de las librerías de EIDE, como por ejemplo, "launched", "entered", "said" que corresponden a determinadas acciones que llevan a cabo los agentes dentro de la IE.

Otro paquete que merece ser comentado es el "es.csic.iiia.udt.giea.performance". Éste es generado por el aBUILDER y, salvo casos excepcionales, no habrá que modificar el código generado. Aquí están contenidas un conjunto de clases abstractas que estructuran y facilitan la programación, así como también las clases que corresponden a especificaciones dadas en el diseño. En particular podemos destacar: tipos de datos, marco dialógico y ontología. Veamos a continuación el contenido del paquete "performance":

ii. Rellenar los esqueletos de agentes generados por aBUILDER. En esta fase partimos de un código que nos da la posibilidad de programar para cada agente: la interpretación que cada uno hace de los mensajes que recibe en cada protocolo, cuál debe ser el contenido de los mensajes que emite en cada protocolo y qué caminos deben seguir en de acuerdo al flujo de trabajo especificado. En resumen, esta fase nos permite agregar el código necesario para dar "inteligencia" a los agentes (tanto internos como externos) en el marco normativo que establece la institución electrónica.

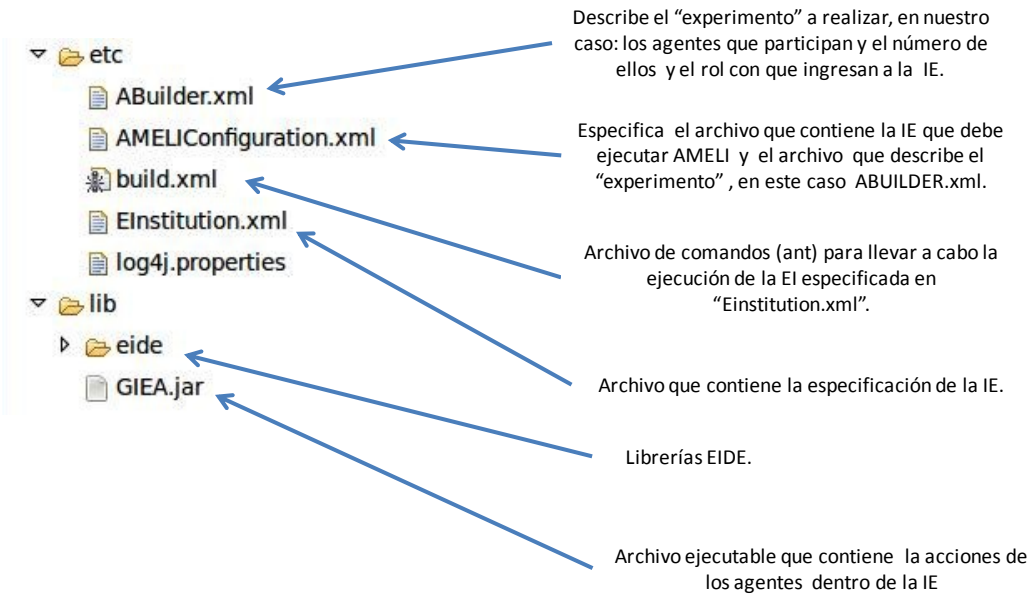
Para llevar a cabo esta tarea será necesario recorrer cada uno de los paquetes (salvo el de "performance") y rellenar con el código correspondientes a la información interna del agente, todas las acciones dialógicas de cada protocolo y los movimientos dentro de la(s) estructura(s) performativa(s). Por ejemplo, para el agente "participante" será necesario rellenar:

- La información y los métodos propios del al agente: "ParticipanteAgent.java".
- Las acciones en el protocolo vestíbulo, único donde actúa con el rol de "participante":
    - ParticipanteProtocoloVestibuloScenePerf.java
- Los movimientos en la estructura performativa (en la sub-estructura performativa ProcesoAcuerdo interviene ejerciendo los roles de convocante o invitado):
    - ParticipantePerformativeStructureTask.java
- Las acciones en los protocolos Acceso, Negociación y validación con el rol de convocante:
    - ConvocanteProtocoloAccesoScenePerf.java
    - ConvocanteProtocoloNegociacionScenePerf.java
    - ConvocanteProtocoloValidacionScenePerf.java
- Los movimientos en la sub-estructura performativa ProcesoAcuerdo con el rol de convocante:
    - ConvocantePSProcesoAcuerdoTask.java
- Las acciones en los protocolos Acceso, Negociación y validación con el rol de Invitado:
    - InvitadoProtocoloAccesoScenePerf.java
    - InvitadoProtocoloNegociacionScenePerf.java
    - InvitadoProtocoloValidacionScenePerf.java
- Los movimientos en la sub-estructura performativa ProcesoAcuerdo con el rol de Invitado:
    - InvitadoPSProcesoAcuerdoTask.java

iii. Ejecución. La ejecución de la IE comienza con la inicialización de *AMELI* una vez que se carga la especificación. En ese momento, los agentes (a través del software

programado en ii.) pueden acceder a la institución para interactuar con los otros agentes a través de *AMELI*.

El proyecto java que crea aBUILDER también crea el contexto de ejecución de la IE en los paquetes: "etc" y "lib". En nuestro caso son los paquetes GI/etc y GI/lib respectivamente. En la primera se almacenan un conjunto de archivos que permiten la coherencia en el contexto de EIDE y en la segunda la librería de EIDE propiamente dicha. Vemos ambos paquetes:



Si se desea ver la documentación completa del software desarrollado para la implantación de la IE ver: Documentación del software .

8 Cómo agregar un nuevo protocolo de negociación.

Para este primer prototipo se modeló un solo protocolo de negociación (el protocolo a sobre cerrado con precio de reserva, ver sección 7.2: Escena Negociacion). No obstante, el entorno de desarrollo permite la inclusión de nuevos protocolos de diversas maneras. En este trabajo explicaremos una de ellas que, sin ser sofisticada en términos de las posibilidades que ofrece el EIDE y el java, es relativamente sencilla.

Para agregar nuevos protocolos de negociación será necesario:

- Modificar el archivo de especificación de la IE. Usando ISLANDER como herramienta, se modificará la IE para modelar un nuevo protocolo de negociación. Dicho protocolo debe sumarse a los otros protocolos de la escena "Negociacion" (en este momento sólo existe el protocolo sobre cerrado con precio de reserva).

- Generar el software para el nuevo protocolo. Esto se realiza creando un nuevo proyecto java con aBUILDER, ahora tomando como entrada la nueva especificación (la que ya incluye el nuevo protocolo). El aBUILDER, siguiendo su patrón de generación de código, agregará en cada uno de los paquetes de los agentes que intervienen en el nuevo protocolo las clases que corresponden a las acciones especificadas para cada rol ejercido por el agente en dicho protocolo. En nuestro caso, los paquetes que tendrán agregaciones serán los que corresponden a los agentes participante ("es.csic.iiia.udt.giea.participante") e intermediario ("es.csic.iiia.udt.giea.intermediario"). Como el agente participante puede ejercer los roles de convocante e invitado, entonces, dentro del paquete "participante", habrá dos nuevas clases que se agregan. En el caso de agente intermediario, se agregará solo una clase que contiene todas las acciones especificadas para él en el nuevo protocolo. En el nuevo paquete "performance" ("es.csic.iiia.udt.giea.performance") se agregarán las nuevas clase abstractas que corresponden a la adición de un nuevo protocolo y, dependiendo de las características del mismo, también pueden producirse se variaciones en: ontología ("ontology"), marco dialógico ("DialogicalFramework"), y tipos de dato ("DatosDelproceso", "Acuerdo").

- Agregar y sustituir software a la aplicación original.

  Una vez identificadas las clases que deben ser agregadas, estás deben ser incluidas en los paquetes correspondientes de la aplicación original y rellenadas con el código de las acciones que los agentes deben realizar en el protocolo que se está incluyendo.

  También será necesario reemplazar el paquete "performance" de la aplicación original por el nuevo "performance" generado (esto nos asegura que se reflejen los cambios en: ontología, marco dialógico, tipos de dato y clases abstractas).

  Finalmente, reemplazar en la aplicación original el paquete "etc" el archivo "EInstitution.xml" con el archivo XML (generado por ISLANDER) que contiene la nueva especificación de la IE.

9 Parametrización de agentes de software – Ejemplo.

A continuación se propone una estructura XML para agregar información que les sirva como entrada a los agentes de software que acceden a la plataforma.

- El nodo raíz lo denominamos <Agentes>… </Agentes>. Entre estas dos etiquetas estará contenida toda la información de cada uno de los agentes.
- Cada agente estará etiquetado como <Agente> … </Agente> y contendrá los siguientes elementos:

  - <Agente nombre="nombre_de_agente">. El atributo nombre indica el identificador del agente.

- <Convocante>valor_booleano</Convocante>. Se señala si el agente va a proponer la negociación (true) o si va actuar como invitado (false) en una negociación de su interés.
- <Protocolo nombre="nombre_del_protocolo" at1=”x” at2=”x1”… atn=”xn”> </Protocolo>. Se Indica el nombre del protocolo que usará en la negociación. Y una lista de atributos que corresponde a las condiciones impuestas para dicho protocolo. Por ahora, solo hemos definido el protocolo "Subasta a sobre cerrado con precio de reserva (SCPR)". Para este último protocolo usamos los atributos “MinInv”, que establece un número mínimo de participantes en la puja y “PcioRes” que impone un precio de reserva inicial.

  En caso de que se trate de un agente que desee como actuar como invitado se estaría indicando que está dispuesto a aceptar una negociación, sólo en caso de que se trate del protocolo especificado. En esta situación, el agente no podría imponer condiciones (que son propuestas por el que convoca), por lo tanto, no haría falta rellenar los atributos.

- <Rol>rol_en_negociación</Rol>. Se indica si el agente actuará como comprador o vendedor.
- <Producto>product_que negocia</Producto>. Especifica el producto que el agente está interesado en vender o comprar.
- <Precio>precio_de_negociación</Precio>. Indica el precio: si se trata de un agente comprador, el máximo precio que está dispuesto a pagar; si se trata de un vendedor, el mínimo precio por el que está dispuesto vender. En caso de que se trate del protocolo de subasta a sobre cerrado, el precio de reserva ya ha sido asignado en el atributo correspondiente en la etiqueta “Protocolo”, por lo tanto, esta etiqueta contendría el precio límite aceptable si se efectúan varias rondas de este protocolo o de otro con similares características.
- <Criterio>criterio_establecido</Criterio>. Se establece el criterio de invitación del convocante. En este caso habría que definir con precisión las diferentes modalidades para invitar a un proceso de negociación. Por ahora, la aplicación solo trabaja con el criterio “lista”, es decir, una lista que contiene los nombres de los invitados. En el ejemplo se una el criterio: “compradorHabitual” que define un grupo de negociantes que habitualmente negocian con el convocante. De igual manera se pueden establecer otros muchos criterios que satisfagan las exigencias del mercado.
- <Contrapartes>…</Contrapartes>. Estas etiquetas delimitan la lista de potenciales negociantes. En el caso de que el agente sea convocante de la negociación, serán sus invitados, en caso contrario, serán aquellos de los cuales puede aceptar invitaciones.
- Cada uno de los potenciales negociantes estarán delimitados las etiquetas <Contraparte>nombre_contraparte</Contraparte>. Nombre_contraparte

es la identificación de cada uno de los potenciales invitados o convocantes, según sea el caso.

- <TiempoDeEspera>tiempo</TiempoDeEspera>. Indican el tiempo máximo que está dispuesto a esperar un convocante después de haber realizado una invitación. En caso de que un agente no actúe como convocante "tiempo" no es una variable a considerar. En los ejemplo "tiempo" se mide en milisegundos.
- <PerfilNegociador>perfil</PerfilNegociador>. Dentro de estas etiquetas el perfil nos proporciona información con respecto al carácter del negociador. Por el momento solo hemos definido dos: "audaz" y "conservador". Esto serviría para personalizar determinadas decisiones en los diferentes procesos de negociación.

Ejemplo del xml que describe una lista de agentes y su información para negociar:

```
<Agentes>
    <Agente nombre="SwAgent1-1TradSell">
        <Convocante>true</Convocante>
        <Protocolo nombre="SCPR" MinInv="3" PcioRes="100.0"</Protocolo>
        <Rol>vendedor</Rol>
        <Producto>plastico</Producto>
        <Precio>90.0</Precio>
        <Criterio>compradorHabitual</Criterio>
        <Contrapartes>
            <Contraparte></Contraparte>
        </Contrapartes>
        <TiempoDeEspera>5000</TiempoDeEspera>
        <PerfilNegociador>audaz</Perfilnegociador>
    </Agent>
    <Agente nombre="SwAgentSBNoTradBuy">
        <Convocante>false</Convocante>
        <Protocolo nombre="SCPR" </Protocolo>
        <Rol>comprador</Rol>
```

```
                <Producto>chatarra</Producto>
                <Precio>100.0</Precio>
                <Criterio>lista</Criterio>
                <Contrapartes>
                        <Contraparte>empresa1</Contraparte>
                        <Contraparte>empresa2</Contraparte>
                        <Contraparte> empresa3</Contraparte>
                        <Contraparte> empresa4</Contraparte>
                        <Contraparte> empresa5</Contraparte>
                </Contrapartes>
                <TiempoDeEspera>2500</TiempoDeEspera>
                <PerfilNegociador>conservador</PerfilNegociador>
        </Agente>
</Agentes>
```

10 Trabajo futuro.

Para versiones posteriores, se puede plantear el prototipo como una aplicación proveedora de servicios web. Para esto, habrá que considerar factores tales como: la organización y distribución de datos del mercado (repositorios centralizados, datos locales, interconexión entre ellos), tipo y condiciones de determinadas negociaciones, cómo se desea articular el servicio con las aplicaciones cliente, etc.
En el caso de que la aplicación sea un proveedor de servicios web, para la interconexión con el espacio GreenIDI sólo bastaría con intercambiar URLs y trasmitir un conjunto de parámetros. Ahora bien, como en cualquier servicio web, para que la parametrización cobre sentido habrá que definir, previamente, un lenguaje de interoperabilidad (expresado en formato xml) entre la plataforma de acuerdos y las aplicaciones cliente que nos permita transmitir la información de entrada para el comportamiento de los agentes de software, así como la necesaria para que grabemos la información generada en los proceso de negociación.

Cabe destacar, que en la medida en que se incorporen parámetros que permita a los agentes tomar decisiones, se podrán definir más y mejores servicios web. Por ello, el aspecto clave del lenguaje de interoperabilidad es cómo estructurar la data que incorpore “inteligencia” a los agentes que fungirán como usuarios humanos.

11 Bibliografía.

[1] Ingeniería de Sistemas Multiagente vía Instituciones Electrónicas. Carles Sierra, Juan A. Rodríguez-Aguilar, Pablo Noriega, Josep Ll. Arcos, Marc Esteva : NOVÁTICA NO. 170. Julio-agosto (2004). pp20-25

http://www.iiia.csic.es/~jar/papers/2004/novatica-spanish.pdf [29-06-2012]

[2] EIDE Tutorial. Introduction to Electronic Institutions and the Electronic Institutions Development Environment (EIDE). http://e-institutions.iiia.csic.es/ [29-06-2012]

[3] EIDE Tutorial. A gentle tutorial on Islander, the graphical specification tool of EIDE.

http://e-institutions.iiia.csic.es/ [29-06-2012]

[4] An Integrated Development Environment for Electronic Institutions. Josep Arcos, Marc Esteva, Pablo Noriega, Juan Rodríguez and Carles Sierra . In Rainer Unland, Matthias Klusch, and Monique Calisti, editors, Software Agent-Based Applications, Platforms and Development Kits, pages 121-142. Birkhauser, (2005).

http://www.iiia.csic.es/files/pdfs/967.pdf [29-06-2012]