

A NOVEL ARCHITECTURE FOR RECOGNIZING SYMBOLS IN A CAD ENVIRONMENT

Fernández-Pacheco, D.G.¹; Conesa, J.¹; Albert, F.²; Cavas-Martínez, F.¹; Aleixos, N.²

¹ Universidad Politécnica de Cartagena, ² Universitat Politècnica de València

The use of sketch based interfaces is not so extended due to the problems they present at the moment, like the lack of robustness, repeatability and reliability. The current recognizers used on line in some interactive applications do not offer ideal solutions, they use to be rigid and the success ratio in the classification decreases when the number of symbols admitted by the system increases. This kind of interfaces could, for instance, support the initial stages of design, in which the traditional pen and paper are still necessary, achieving the link of these phases with the parametric model of the product. In this paper a reliable and robust agent-based architecture to support the user in the first conceptual design stages by recognizing hand-sketched symbols is presented and evaluated.

Keywords: *Sketches; Natural interface; CAD; Multi-agent system*

UNA NUEVA ARQUITECTURA PARA EL RECONOCIMIENTO DE SÍMBOLOS EN UN ENTORNO CAD

El empleo de los interfaces basados en bocetos no está muy extendido debido a los problemas que presentan en la actualidad, tales como el de la robustez, repetibilidad o fiabilidad. Los reconocedores actuales que se pueden utilizar en línea sobre aplicaciones interactivas no ofrecen soluciones idóneas, suelen ser poco flexibles y su porcentaje de acierto en la clasificación decrece según aumenta el número de símbolos que admite el sistema. Este tipo de interfaces permitiría, por ejemplo, soportar las fases iniciales del diseño que actualmente se realizan íntegramente con lápiz y papel, consiguiendo enlazar dichas fases con el modelo paramétrico del producto. En este trabajo se presenta y evalúa una arquitectura fiable y robusta basada en multi-agentes que puede dar soporte al usuario en esas primeras etapas del diseño conceptual mediante el reconocimiento de símbolos realizados a mano.

Palabras clave: *Bocetos; Interfaz natural; CAD; Sistema multi-agente*

1. Introducción

Estudios llevados a cabo por la Engineering Design Graphics Division de la American Society for Engineering Education (Barr, 2004) y la American Society of Mechanical Engineers ASME (Rose, 2005) demuestran que las dos destrezas más valoradas como competencia de los estudiantes de ingeniería en el ámbito de la comunicación gráfica son la “habilidad para bocetar objetos de ingeniería a mano alzada” y la “habilidad para crear modelos sólidos 3D por ordenador”. Otros estudios, como los realizados por Plimmer y Apperley (2002) o Tversky (2002), analizan el papel tan importante que desempeña el uso de bocetos en la industria durante el proceso de desarrollo de nuevos productos.

La llegada del CAD (Computer Aided Design) supuso un cambio importante en las últimas fases del proceso de diseño, pero no alteró prácticamente a la fase de diseño conceptual, en la que se ha mantenido el uso de los bocetos realizados con lápiz y papel. En la práctica el diseñador sólo comienza a crear un modelo CAD 3D desde cero una vez ha obtenido un boceto final. El motivo de esto se debe principalmente a las carencias que tienen las herramientas CAD disponibles tanto a nivel comercial como académico, si bien cuando se han desarrollado herramientas con una funcionalidad adecuada, éstas han demostrado ser efectivas. Por ejemplo, se ha demostrado que el uso de herramientas CAS (Computer Aided Sketching) es muy útil para desarrollar las habilidades de visión espacial en estudiantes de ingeniería (Contero et al., 2005). Sin embargo, la razón por la que las herramientas CAS no se han desarrollado de forma adecuada aún se debe, en primer lugar, a que el hardware necesario para su implementación no ha estado disponible hasta la reciente aparición en el mercado de los Tablets y otros dispositivos portátiles táctiles, y en segundo lugar, a la limitada funcionalidad de dichos dispositivos, que no ha mejorado las prestaciones del bocetado tradicional sobre papel.

Estos nuevos dispositivos han surgido como instrumentos de dibujo estándares y han supuesto un importante avance en el desarrollo de nuevos interfaces de usuario, conocidos también como interfaces caligráficos. Muestra de ello es el sistema CIGRO propuesto por Contero et al. (2003), donde se aporta un interfaz caligráfico de instrucciones reducidas para la creación de objetos poliédricos mediante bocetos.

Las mejoras que ofrece el empleo de estos interfaces posibilita el desarrollo de aplicaciones muy diversas, como por ejemplo la aplicación propuesta por Wu et al. (2007) para definir formas en 3D a partir de una única vista empleando “paletas de forma”, esto es, se dibuja una primitiva 2D en una vista bi-dimensional y se especifica posteriormente su orientación 3D dibujando la primitiva correspondiente en la paleta de forma, o el sistema interactivo Plushie desarrollado por Mori e Igarashi (2007) que permite a usuarios inexpertos diseñar sus propios muñecos de trapo mediante bocetos, implementando comandos gestuales que permiten recortar o alargar partes del modelo existentes, o el sistema FiberMesh presentado por Nealen et al. (2007) para el diseño de superficies de forma libre mediante curvas 3D bocetadas.

Sin embargo, uno de los usos más importantes de los que podemos dotar a un interfaz caligráfico es el de proveer a un sistema CAD de la capacidad de reconocer de forma automática ciertos símbolos bocetados a mano para utilizarlos como comandos de un software concreto, en lugar de tener que recurrir al uso de menús complejos y poco intuitivos.

El reconocimiento de símbolos resulta una tarea muy compleja si tenemos en cuenta que un mismo símbolo se puede dibujar de forma muy diferente según el usuario que lo bocete, variando incluso la orientación y el tamaño. Además, el orden seguido por cada usuario durante la fase de bocetado puede influenciar en el resultado final del reconocimiento.

Puntos finales de línea que no coincidan con los puntos iniciales de otra línea, o líneas que terminan interceptando con otras líneas pueden ser algunos de los problemas principales que podemos encontrarlos. Debido a estos inconvenientes muchos algoritmos de reconocimiento resultan poco robustos o presentan ambigüedades en sus decisiones, siendo necesaria en varias ocasiones la intervención del usuario (Ghorbel et al., 2013; Ghorbel, Lemaitre & Anquetil, 2012; Wang, Wang & Qin, 2012). Es por esta razón que se hace imprescindible un cambio en la metodología para que estos reconocedores diferencien y clasifiquen los dibujos como lo haría un ser humano, esto es, utilizando información del contexto, siendo flexible en las decisiones intermedias y finales, pudiendo llegar a contradicciones que obliguen a cambiar la decisión a través de negociaciones consensuadas. Este tipo de comportamiento se le puede atribuir al reconocedor mediante el uso de sistemas expertos, de sistemas basados en objetos, y de sistemas basados en agentes, entre otros. De entre estos, la tecnología basada en agentes se ha empleado con éxito en muchas ocasiones para simulación y control de procesos, organización, control de tráfico, navegación, etc. aunque poco a poco su uso se está empezando a utilizar cada vez más en procesos que intervienen en entornos de visión artificial, análisis de imagen y tareas de reconocimiento, como es el caso de la interpretación de bocetos para soportar interfaces caligráficas. Por ejemplo, Juchmes, Leclercq y Azar (2005) recurren al empleo de los sistemas multi-agente para desarrollar una herramienta de interpretación de bocetos en la etapa conceptual del diseño arquitectónico, siendo capaz de capturar líneas y componer de forma progresiva los modelos tecnológico y funcional del edificio que se está diseñando. Una estructura similar es presentada por Azar et al. (2006), que proponen e implementan un interfaz multimodal para la interpretación de bocetos arquitectónicos basado en una arquitectura multi-agente, la cual está compuesta por varios agentes gráficos encargados del reconocimiento de elementos gráficos básicos (como líneas y círculos) y de otros más complejos (como puertas, escaleras, etc.), característicos del diseño arquitectónico. Otros autores como Mackenzie y Alechina (2003) optan por aplicar una arquitectura multi-agente a un clasificador de bocetos de animales dibujados por niños.

A la hora de obtener un buen reconocimiento resulta crucial extraer la información del contexto, pues en muchas ocasiones nos permite además resolver ambigüedades. Este es el caso precisamente del reconocedor de diagramas UML bocetados propuesto por Casella et al. (2008), basado en una arquitectura multi-agente. El proceso de reconocimiento es llevado a cabo por una serie de "agentes inteligentes" que se encargan de reconocer los símbolos bocetados y de coordinarse entre sí con el fin de obtener una interpretación eficiente y precisa del boceto introducido.

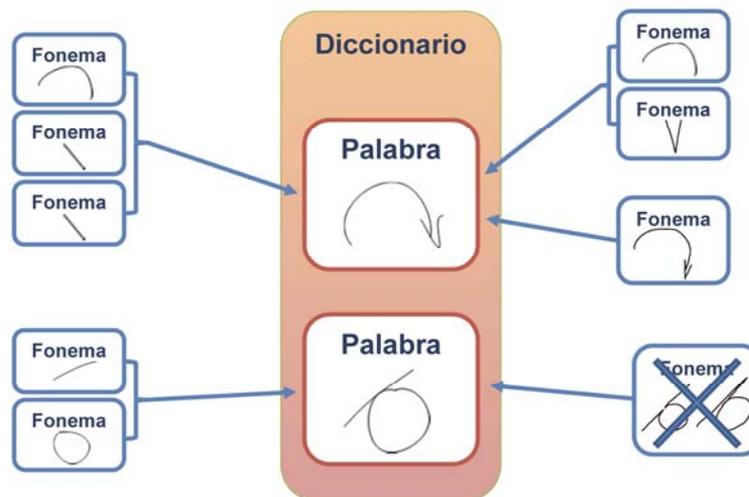
Sin embargo, los sistemas descritos hasta ahora no tienen en cuenta otros símbolos esenciales que permitan crear secciones 2D o geometría 3D (como por ejemplo restricciones geométricas o dimensionales). Por este motivo, en este artículo se define una arquitectura multi-agente que permite el reconocimiento de bocetos dibujados a mano, como parte de un software de más alto nivel, detectando tanto entidades (líneas, arcos, círculos, curvas, elipses, etc.) como comandos gestuales para su tratamiento (extrusión, revolución, simetría y borrado) y restricciones geométricas/dimensionales (acotación, concetricidad, paralelismo, perpendicularidad, etc.). El empleo de agentes permite a su vez ampliar el campo de aplicación de forma sencilla mediante la creación de nuevos agentes, que se pueden incorporar siguiendo una estructura modular y dotando así al sistema de cierta flexibilidad.

2. Materiales y métodos

2.1 Paradigma para el sistema de reconocimiento

Previamente al desarrollo de la arquitectura multi-agente, se definió un paradigma que permitiera llevar a cabo el reconocimiento de los bocetos. En este paradigma se hizo un símil con el lenguaje hablado, haciendo corresponder a cada símbolo, ya sea primitivo o combinado, con una palabra, de modo que un símbolo estaría compuesto por uno o más trazos, considerando a cada uno de estos trazos como los fonemas que forman una palabra, las cuales a su vez conforman un diccionario. Así pues, por ejemplo, para el símbolo correspondiente al comando gestual de revolución (ver figura 1), pueden existir 3 trazos o fonemas que conformen el símbolo (2 líneas y un arco), o dos fonemas (un arco y un ángulo), e incluso un único fonema (en el caso de dibujar el símbolo de una sola vez). En el caso del símbolo correspondiente a la restricción geométrica de tangencia (ver figura 1), no tendría sentido dibujarlo de una sola vez mediante un único trazo, por lo que esta posibilidad no se contempla, y tan sólo podrá estar compuesto por dos fonemas (una línea y un círculo).

Figura 1: Paradigma para el proceso de reconocimiento



Para el diseño del paradigma se han elegido como fonemas (o símbolos primitivos) todos aquellos trazos dibujados de una sola vez y que pueden formar parte de los símbolos o palabras que se han contemplado en un diccionario (ver figura 2). A su vez, se han contemplado como palabras (o símbolos combinados) todos aquellos símbolos que forman un conjunto mínimo para poder crear, editar y parametrizar geometría 2D básica (figura 2).

2.2 Arquitectura del sistema multi-agente

Una estructura multi-agente ofrece, por su propia naturaleza, la posibilidad de centrar las tareas de reconocimiento de un símbolo en función del conjunto de entidades que interesa detectar. De esta manera, es posible agrupar los símbolos en áreas temáticas definidas en ingeniería. A modo de ejemplo podemos bocetar símbolos empleados en el dibujo artístico (línea y arco artísticos), símbolos que permitan ejecutar operaciones de modelado (extrusión, revolución, sección o eje de simetría/revolución), símbolos relacionados con el análisis estructural (uniones fijas, uniones articuladas o uniones deslizantes), símbolos relacionados con el diseño industrial (roscas, rodamientos o ruedas dentadas), símbolos asociados a restricciones del diseño (horizontalidad, verticalidad, paralelismo, perpendicularidad, igualdad, concetricidad, tangencia), o símbolos de acotación (cota radial, cota diametral, cota lineal).

Figura 2: Símbolos primitivos y combinados del diccionario implementado

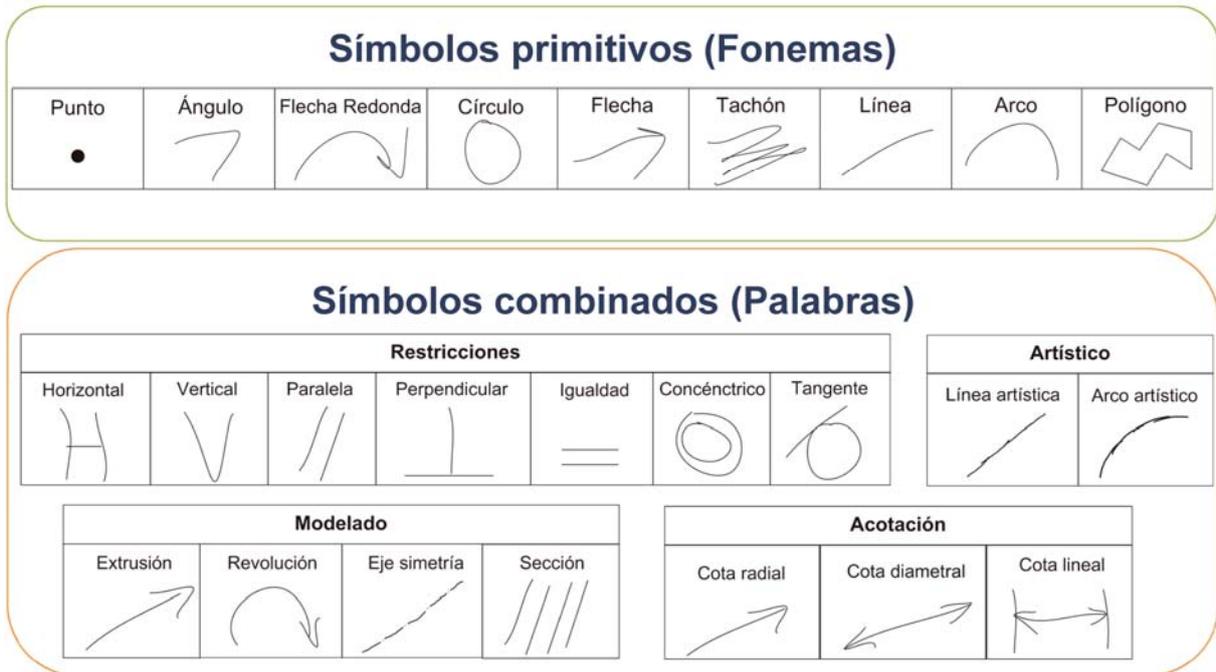
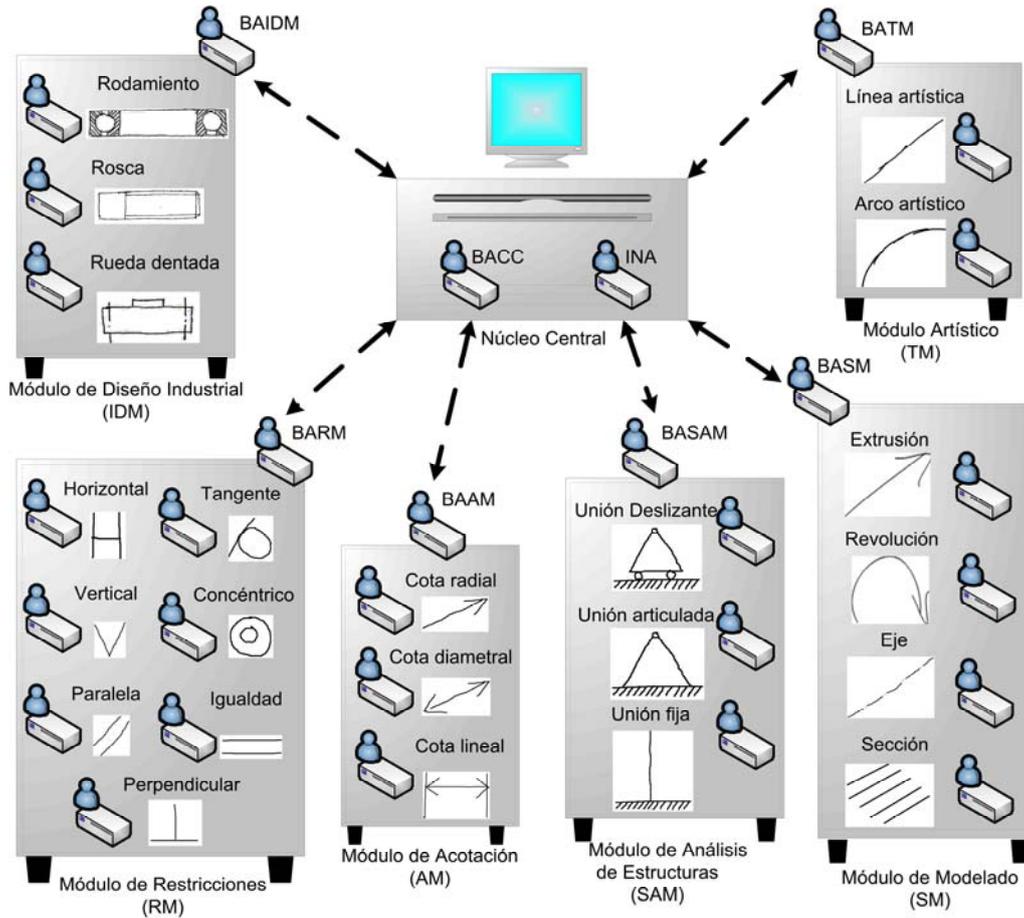


Figura 3: Esquema general de la plataforma de bocetado basado en agentes



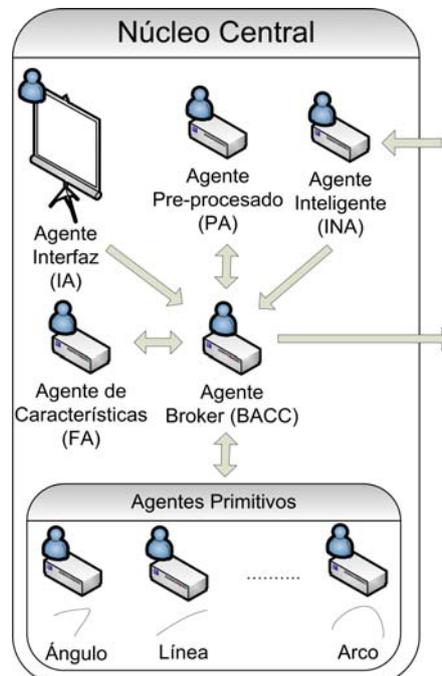
La plataforma que se propone presenta una estructura integrada, por un lado, de un núcleo central que actúa de interfaz con el usuario y en el que se definen los agentes encargados de realizar las tareas básicas de reconocimiento, coordinados por un agente broker (BACC); y por otro lado, de una serie de módulos gestionados cada uno de ellos por otro agente broker (BAM), que reparte las tareas a los agentes encargados del reconocimiento de cada uno de los símbolos definidos en el módulo y que informa a un núcleo central del resultado de sus agentes para que tome la decisión final (ver figura 3). Cabe mencionar que en esta figura (y en sucesivas) cada icono representa a un agente del sistema.

2.3 El núcleo central

En el núcleo central de la aplicación es donde se realiza el reconocimiento de todos los trazos introducidos por el usuario (ver figura 4). En este trabajo se han diferenciado hasta un total de 9 trazos a los que se les ha denominado símbolos primitivos y a partir de los cuales es posible componer todos los símbolos definidos en los distintos módulos, conformando el diccionario de primitivos. Para identificar cada símbolo primitivo se ha definido un agente, y al conjunto de éstos se les ha denominado agentes primitivos (ver figura 2).

Cuando un usuario dibuja un trazo, el agente interfaz IA lo muestra y envía los puntos digitalizados a través de una estructura de datos al agente Broker del núcleo central (BACC). El agente Broker BACC recibe la notificación y envía los puntos del trazado al agente de pre-procesado PA o Pre-processing Agent. De forma ideal, estos puntos deberían estar distribuidos de forma uniforme. Sin embargo, cuanto más rápido se dibuja el trazo, menos puntos se digitalizan, lo que provoca una variación en la concentración de puntos, además de que el trazado pueda ser más o menos tembloroso o titubeante. El agente PA debe filtrar y eliminar este ruido para conseguir un trazo óptimo listo para el reconocimiento.

Figura 4: Estructura del Núcleo Central



Cuando un usuario dibuja un trazo, el agente interfaz IA lo muestra y envía los puntos digitalizados a través de una estructura de datos al agente Broker del núcleo central (BACC). El agente Broker BACC recibe la notificación y envía los puntos del trazado al agente de pre-procesado PA o Pre-processing Agent. De forma ideal, estos puntos deberían

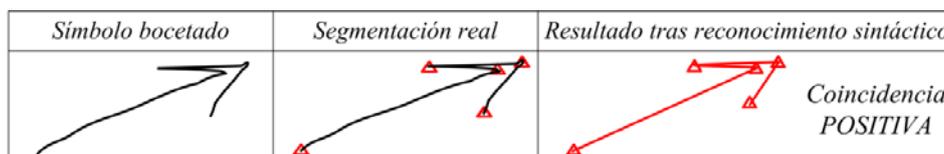
estar distribuidos de forma uniforme. Sin embargo, cuanto más rápido se dibuja el trazo, menos puntos se digitalizan, lo que provoca una variación en la concentración de puntos, además de que el trazado pueda ser más o menos tembloroso o titubeante. El agente PA debe filtrar y eliminar este ruido para conseguir un trazo óptimo listo para el reconocimiento.

Una vez se ha completado el pre-procesado, el agente BACC solicita la extracción de características al agente FA o Feature Agent. El agente de extracción de características FA posee un funcionamiento secuencial, distinguiéndose dos partes claramente diferenciadas. Por un lado se realiza la “segmentación” del trazo, extrayendo las primitivas geométricas que lo integran y aportando información de las mismas que posteriormente se empleará en las etapas de decisión. Por otro lado se obtienen una serie de características o “cues” del trazo introducido. Una vez el agente FA ha extraído ambas informaciones, devuelve dichas características al agente BACC, el cual reenviará la información a los respectivos agentes primitivos.

2.4 Los agentes primitivos

Cada uno de los agentes primitivos empleará únicamente la información que le resulte significativa con el fin de encontrar pistas relevantes que le permitan reconocer el símbolo primitivo del cual se encarga. El agente primitivo llevará a cabo un reconocimiento basado en la geometría del trazo, devolviendo un resultado de coincidencia o no-coincidencia tras un análisis sintáctico. Dicho análisis sintáctico se realiza sobre los vértices y las primitivas aproximadas a partir de la segmentación obtenida por el agente FA (ver figura 5).

Figura 5: Coincidencia de los vértices y primitivas aproximadas



Los agentes primitivos evalúan y cuantifican la correspondencia o no de un trazo con su correspondiente símbolo primitivo pero no toman una decisión final, devolviendo así al agente BACC los resultados de la primera etapa de reconocimiento. El agente BACC se sirve de la información suministrada por los agentes primitivos para tomar la decisión sobre qué tipo de símbolo primitivo se trata e informar a los distintos agentes BAM de todos los módulos y al agente INA, iniciándose así la segunda etapa de reconocimiento.

La información facilitada por los agentes primitivos y el análisis realizado por el agente BACC será el punto de partida para intentar intuir qué símbolo está siendo esbozado por el usuario, o si por el contrario, el trazo forma parte de la geometría del boceto. Un agente inteligente (INA) será el encargado de tomar la decisión final, pero ésta dependerá, tanto de la información facilitada por el agente BACC como de la información que cada módulo de la aplicación envíe al agente INA.

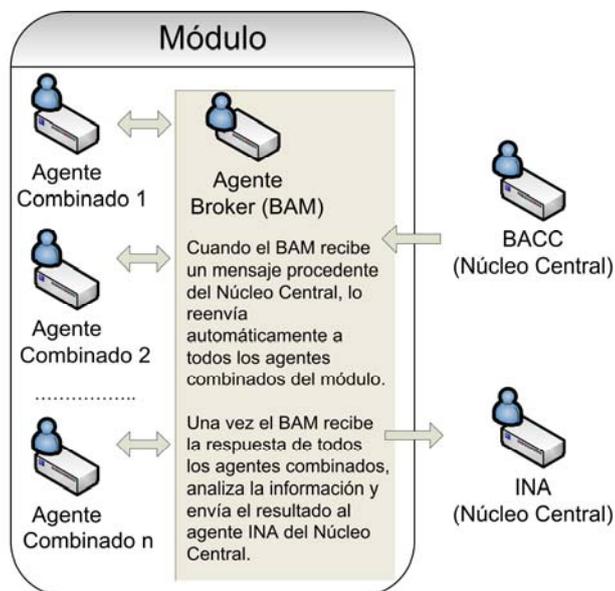
2.5 Los módulos del sistema

Todos los módulos que formarán parte de la plataforma deben tener una estructura similar (ver figura 6). En cada módulo se define un agente por cada símbolo con significado dentro del módulo. A estos símbolos se les denominará “símbolo combinado” y a cada agente encargado de su reconocimiento “Agente combinado”. Los agentes combinados compiten entre sí para reconocer su símbolo combinado a partir de los símbolos primitivos que lo forman.

Cada módulo está gestionado por un agente broker (BAM), el cual tiene una doble función. Por un lado, envía a los agentes combinados de su módulo la información de cada trazo

generada por los agentes primitivos y que recibe del agente broker del módulo central (BACC). Por otro lado, y una vez procesada esta información por los agentes combinados de su módulo, envía al agente inteligente del núcleo central (INA) los resultados del módulo que gestiona.

Figura 6: Coincidencia de los vértices y primitivas aproximadas



2.6 Los agentes combinados

A partir de la información que reciben del análisis realizado por los agentes primitivos, los agentes combinados buscan el reconocimiento de su símbolo. Esta búsqueda se realiza a tres niveles:

1. Estudio contextual: Este análisis está basado en la posición relativa entre los distintos símbolos primitivos reconocidos y las proporciones entre los mismos.
2. Premisas para que el símbolo tenga significado: Es importante distinguir las distintas acciones que vienen asignadas a cada símbolo combinado. Así pues, podemos clasificar en tres grupos las tareas que suponen el reconocimiento de un símbolo:
 - Agentes de modelado (extrusión y revolución): son aquellos cuya ejecución precisa de la selección previa de una superficie. El reconocimiento del símbolo por parte de estos agentes combinados se realiza únicamente si existe tal selección. El agente BACC será el encargado de informar a cada módulo si existe o no la selección previa de dicha superficie.
 - Agentes con referencias (paralelismo, perpendicularidad, igualdad, tangencia, concetricidad): son agentes que deben establecer una relación entre dos entidades geométricas, por lo que el reconocimiento del símbolo pasa por la premisa de que debe haber sido introducido dos veces consecutivas.
 - Agentes independientes (verticalidad, horizontalidad, tachado, cota radial, cota lineal, cota diametral, eje de revolución/simetría, sección, línea y arco artísticos): son agentes que suponen un reconocimiento inmediato, independientemente de cualquier otro símbolo introducido previamente y no precisan de selección alguna.
3. Interpretación semántica de los trazos: Es preciso tener presente que un trazo puede representar por sí solo un símbolo o formar parte de un símbolo más complejo, y que un

mismo símbolo puede estar formado por un mismo conjunto de trazos pero introducidos en distintos orden. A modo de ejemplo, el símbolo de cota diametral podría ser reconocido a partir de cualquiera de las combinaciones de símbolos primitivos que se muestran en la figura 7.

Figura 7: Posibilidades de bocetar el comando gestual de cota diametral

Símbolo que indica cota diametral	Generación del símbolo combinado mediante la combinación de símbolos primitivos
	Unión de cinco símbolos primitivos de línea 
	Unión de tres símbolos primitivos de línea y un ángulo 
	Unión de un símbolo primitivo de línea y dos ángulos 
	Unión de dos símbolos primitivos de línea y un símbolo primitivo de flecha 
	Unión de un símbolo primitivo de ángulo y un símbolo primitivo de flecha 

A partir de la información que reciben del análisis realizado por los agentes primitivos, los agentes combinados buscan el reconocimiento de su símbolo combinado realizando un estudio contextual que haga al símbolo combinado independiente del orden en el que son introducidos sus símbolos primitivos. De esta forma, los símbolos primitivos enviados al agente combinado son añadidos a una lista. A continuación, el agente combinado analiza si el conjunto de símbolos primitivos en la lista constituyen o forman parte del símbolo combinado que busca. De forma paralela, y una vez realizados los distintos análisis mencionados, los agentes combinados actualizan su estado, pudiendo tomar tres valores:

- Rechazado: si el símbolo primitivo no forma parte del símbolo combinado.
- En proceso: si el símbolo primitivo por sí solo, o junto con los símbolos primitivos enviados anteriormente son susceptibles de formar parte del símbolo combinado.
- Aceptado: si el conjunto de símbolos primitivos existentes en la lista identifican el símbolo combinado.

Una vez actualizado su estado, el agente combinado informa al agente broker de su módulo.

2.7 Los agentes broker

Los agentes BAM se encargan de llevar a cabo dos funciones. La primera de ellas es la de ejercer de mero puente para enviar a los agentes combinados la información suministrada por los agentes primitivos y recibida a través del BACC, con la que poder comenzar la etapa de reconocimiento correspondiente. La segunda función, más compleja, es la de gestionar los resultados que arrojan cada uno de los agentes combinados e informar al agente inteligente de las conclusiones de su módulo. La gestión se realiza a partir de la comparación de los estados de los agentes combinados.

Cuando un agente broker recibe un nuevo símbolo primitivo, éste es enviado a los agentes combinados cuyo estado se encuentre en el estado "en proceso". Una vez recibidos los informes de actualización de estado de todos los agentes combinados del módulo, el agente broker emite un informe al agente inteligente que puede contener tres estados:

- Rechazado: si todos los agentes combinados indican el estado "rechazado".
- En proceso: si al menos uno de los agentes combinados indica el estado "en proceso".

- Aceptado: si uno de los agentes combinados indica el estado “aceptado” y todos los demás el estado “rechazado”.

2.8 El agente inteligente INA

El agente inteligente del núcleo central es el encargado de gestionar los informes procedentes de los agentes BAM de los distintos módulos instalados en la plataforma y tomar la decisión final sobre el reconocimiento del símbolo introducido, estableciéndose así una especie de ‘negociación’ entre el agente INA y los agentes Brokers de los distintos módulos. La gestión realizada por el agente inteligente está basada en la información dada por los agentes primitivos y los estados de los agentes BAM de los distintos módulos. En su estado inicial, los agentes BAM se inicializan al estado “en proceso”.

Para el correcto funcionamiento del algoritmo se precisa definir una lista, en la que se van introduciendo todos los trazos realizados por el usuario. Cada vez que se introduce un nuevo trazo, y tras ser éste pre-procesado, el agente BACC envía sus características a los agentes broker de los módulos que se encuentren en el estado “en proceso”. Cada agente BAM establece comunicación con sus respectivos agentes combinados y actualizan sus estados, notificándolos posteriormente al agente INA.

Una vez el agente INA ha recibido los mensajes de actualización de todos los agentes BAM, toma su decisión, basándose en el siguiente comportamiento:

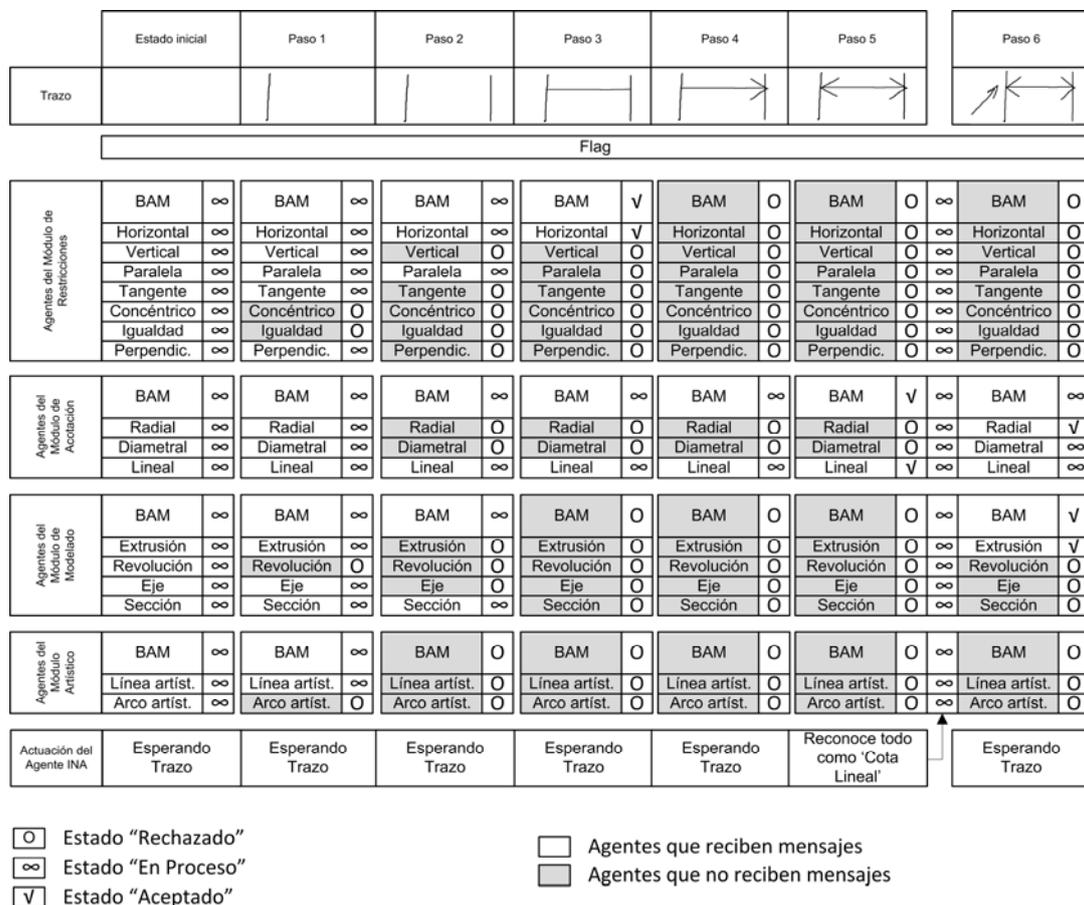
1. Si el trazo no es reconocido por ningún agente primitivo:
 - a. Si algún agente BAM “en proceso” contiene un agente combinado cuyo estado es “aceptado”, se reconoce su símbolo combinado y el nuevo trazo se considera geometría.
 - b. En caso contrario la lista y el nuevo trazo se consideran geometría.Finalizada la comprobación se borra la lista.
2. Si el trazo es reconocido por algún agente primitivo:
 - a. Si el trazo es reconocido por el agente primitivo de tachado:
 - i. Si algún agente BAM tiene un agente combinado cuyo estado es “aceptado” se reconoce su símbolo combinado.
 - ii. En caso contrario, la lista se considera geometría.En cualquiera de los casos anteriores y a posteriori, se reconoce el símbolo primitivo de tachado y se borra la lista.
 - b. En caso contrario el trazo se añade a la lista y se espera la actualización de los agentes BAM:
 - i. Si el estado de todos los agentes BAM es actualizado a “rechazado”:
 1. Si previo a añadir el último trazo algún agente BAM tenía un agente combinado cuyo estado era “aceptado” se reconoce su símbolo combinado, se borra la lista y vuelve a introducirse el último trazo para iniciar el proceso de reconocimiento, ordenando previamente a todos los agentes BAM y combinados a actualizar su estado a “en proceso” (inicialización).
 2. En caso contrario, la lista se considera geometría y se borra su contenido.
 - ii. Si existe algún agente BAM cuyo estado se encuentre “en proceso”, se espera al siguiente trazo.

iii. Si todos los agentes BAM tienen su estado como “rechazado” y sólo uno mantiene su estado como “aceptado”, se asume que la lista es reconocida como símbolo. El agente combinado del módulo en cuestión cuyo estado se mantenga en “aceptado” indica de qué tipo de símbolo se trata. Finalizada la comprobación se borra la lista.

Cada vez que se borra la lista, el agente INA envía un mensaje de inicialización al agente broker BACC, para que éste a su vez les comunique a los agentes BAM de cada módulo que deben pasar al estado “en proceso”. Éstos a su vez deberán informar a los agentes combinados de su módulo que también deben pasar al estado “en proceso”.

Para ilustrar mejor el funcionamiento de la plataforma, en la figura 8 se analiza el reconocimiento de una cota lineal, mostrando los estados de todos los agentes combinados y agentes broker de los módulos de restricciones, acotación, modelado y artístico.

Figura 8: Estado de los agentes durante el reconocimiento de una cota lineal



3. Resultados

La evaluación de la arquitectura propuesta se ha realizado en dos fases. En la primera fase se ha evaluado el reconocimiento llevado a cabo por los Agentes Primitivos, y en la segunda fase se ha evaluado el resultado del reconocimiento final en el que intervienen los Agentes Combinados y el Agente Inteligente INA.

En los test realizados para la primera fase de evaluación han intervenido 10 usuarios de aplicaciones CAD, introduciendo cada uno de ellos varias ocurrencias de todos los símbolos primitivos del alfabeto. En total se han evaluado unos 2200 símbolos primitivos con

diferentes orientaciones y tamaños. Los resultados obtenidos de este reconocimiento se muestran en la figura 9, donde los datos que aparecen enmarcados en negrita en la diagonal de la matriz de confusión se corresponden con el porcentaje de los datos de acierto, apareciendo en el resto de celdas las clasificaciones fallidas para cada símbolo esbozado y su correspondiente porcentaje de confusión. A partir de estos resultados, el promedio final del reconocimiento llevado a cabo por los Agentes Primitivos ha resultado en un porcentaje de éxito del 96.41%.

Figura 9: Resultados del reconocimiento efectuado por los Agentes Primitivos (en %)

Símbolos reconocidos / Símbolos esbozados	Ángulo	Arco	Círculo	Línea	Flecha	Flecha redonda	Punto	Tachón	Polígono
Ángulo	98.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.4
Arco	0.0	94.9	0.0	2.5	0.0	0.0	2.5	0.0	0.0
Círculo	0.0	7.0	93.0	0.0	0.0	0.0	0.0	0.0	0.0
Línea	0.0	0.0	0.0	99.3	0.0	0.0	0.7	0.0	0.0
Flecha	0.0	0.0	0.0	0.0	94.2	5.8	0.0	0.0	0.0
Flecha redonda	0.0	0.0	0.0	0.0	5.7	94.3	0.0	0.0	2.8
Punto	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
Tachón	0.0	0.0	0.0	1.8	0.0	0.0	0.0	95.4	2.8
Polígono	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	97.9

Los test realizados para la segunda fase de evaluación, o evaluación final del interfaz, fueron llevados a cabo con 9 usuarios de aplicaciones CAD. Esto símbolos fueron almacenados hasta un total de 2500 con diferentes orientaciones y tamaños. Una vez introducidos, el interfaz los clasificó según los resultados que muestra la matriz de confusión de la figura 10.

Figura 10: Resultados del reconocimiento efectuado por los Agentes Combinados (en %)

Símbolo Reconocido / Símbolo Bocetado	Módulo de Restricciones geométricas							Módulo de Modelado			Módulo de Acotación		Módulo de Dibujo Artístico		Núcleo Central		
	Concéntrico	Tangente	Vertical	Horizontal	Paralela	Perpendicular	Igualdad	Sección	Eje	Revolución	Extrusión / Cota Radial	Cota Lineal	Cota Diametral	Línea Artística	Arco Artístico	Geometría	Tachón
Concéntrico	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.3	0.0
Tangente	0.0	99.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Vertical	0.0	0.0	98.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0
Horizontal	0.0	0.0	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.3	0.0
Paralela	0.0	0.0	0.0	0.0	100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Perpendicular	0.0	0.0	0.0	0.0	0.0	97.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0
Igualdad	0.0	0.0	0.0	0.0	0.0	0.0	100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sección	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Eje	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	93.0	0.0	0.0	0.0	0.0	0.0	0.0	7.0	0.0
Revolución	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0
Extrusión/Cota Rad	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.0	0.0	0.0	0.0	0.0	1.0	0.0
Cota Lineal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	90.0	0.0	0.0	0.0	10.0	0.0
Cota Diametral	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	97.0	0.0	0.0	3.0	0.0
Línea Artística	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	95.0	0.0	5.0	0.0
Arco Artístico	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	90.0	10.0	0.0
Geometría	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100	0.0
Tachón	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	96.0

4. Conclusiones

El artículo presenta una arquitectura basada en multi-agentes que permite el reconocimiento de símbolos primitivos y compuestos para su implementación en un entorno CAD. El proceso de reconocimiento se lleva a cabo mediante dos niveles de agentes: un primer nivel donde residen los “agentes primitivos” encargados del reconocimiento sintáctico, y un segundo nivel donde se encuentran los agentes combinados encargados del reconocimiento semántico con la ayuda de información contextual. La arquitectura ha sido diseñada para ser fácilmente ampliable mediante la inclusión de nuevos agentes primitivos o combinados.

5. Referencias

- Azar, S., Couvreur, L., Delfosse, V., Jaspartz, B., & Boulanger, C. (2006). An agent-based multimodal interface for sketch interpretation. In *International Workshop on Multimedia Signal Processing (MMSP-06)*, British Columbia, Canada.
- Barr, R.E. (2004). The Current Status of Graphical Communication in Engineering Education. In *34th ASEE/IEEE Frontiers in Education Conference* (pp S1D8-13). Savannah, GA.
- Casella, G., Deufemia, V., Mascardi, V., Costagliola, G., & Martelli, M. (2008). An agent-based framework for sketched symbol interpretation. *Journal of Visual Languages and Computing*, 19, 225–257.
- Contero, M., Naya, F., Jorge, J., & Conesa, J. (2003). CIGRO: A Minimal Instruction Set Calligraphic Interface for Sketch-Based Modeling. *Lecture Notes in Computer Science*, 2669, 549-558.
- Contero, M., Naya, F., Company, P., Saorin, J.L., & Conesa, J. (2005). Improving visualization skills in engineering education. *IEEE Computer Graphics and Applications*, 25(5), 24-31.
- Juchmes, R., Leclercq, P., & Azar, S. (2005). A freehand-sketch environment for architectural design supported by a multi-agent system. *Computers & Graphics*, 29 (6), 905–915.
- Mackenzie, G., & Alechina, N. (2003). Classifying sketches of animals using an agent-based system. In *10th International Conference on Computer Analysis of Images and Patterns (CAIP'03)* (vol 2756, pp. 521-529), Berlin.
- Mori, Y., & Igarashi, T. (2007). Plushie: An Interactive Design System for Plush Toys. In *ACM Transactions on Graphics (SIGGRAPH 2007)* (vol 26(3), Article 45).
- Nealen, A., Igarashi, T., Sorkine, O., & Alexa, M. (2007). FiberMesh: designing freeform surfaces with 3D curves. In *ACM Transactions on Graphics (SIGGRAPH 2007)* (vol 26(3), Article 41).
- Plimmer, B., & Apperley, M. (2002). Computer-Aided Sketching to Capture Preliminary Design. In *Third Australasian conference on User interfaces* (Vol 7, pp. 9-12).
- Rose, A.T. (2005). Graphical Communication Using Hand-Drawn Sketches in Civil Engineering. *Journal of Professional Issues in Engineering Education and Practice*. 131(4), 238-247.
- Tversky, B. (2002). What do Sketches say about Thinking?. In *AAAI Spring Symposium Series - Sketch Understanding* (pp. 148-152).
- Wu, T-P., Tang, C-K., Brown, M.S., & Shum, H-Y. (2007). ShapePalettes: interactive normal transfer via sketching. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2007)* (Article 44).