

# Sistemas Multiagente Inalámbricos Basados en Agentes CBR-BDI: de la Teoría a la Práctica

Javier Bajo<sup>1</sup>, Ana de Luis<sup>2</sup>, Dante I. Tapia<sup>2</sup>, Juan Manuel Corchado<sup>2</sup>

<sup>1</sup> Universidad Pontificia de Salamanca  
Compañía 5, 37002, Salamanca, Spain  
jbajope@upsa.es

<sup>2</sup> Departamento Informática y Automática, Universidad de Salamanca  
Plaza de la Merced s/n, 37008, Salamanca, Spain  
{adeluis, dantetapia, corchado}@usal.es

**Abstract.** En este artículo se presenta un sistema multiagente desarrollado para la realización de predicciones sobre la absorción o expulsión de CO<sub>2</sub> en el océano Atlántico de forma autónoma. La arquitectura propuesta se basa en el desarrollo de agentes deliberativos, que utilizan sistemas de razonamiento basados en casos para modelar dicho flujo. En este artículo se presenta arquitectura del sistema, así como los resultados preliminares obtenidos.

## 1. Introducción

Los sistemas multiagente (SMA) son aplicaciones informáticas distribuidas con autonomía y cierto grado de “inteligencia”. Estos grandes sistemas están compuestos por unidades pequeñas llamadas agentes. Un agente es una entidad que debe tener ciertas características, tales como autonomía, situación, reactividad, proactividad, habilidad social, aprendizaje, movilidad u organización (Wooldridge y Jennings, 1995). Estas capacidades pueden modelarse de distintas formas. Por ejemplo la capacidad de proactividad o razonamiento puede modelarse utilizando sistemas de razonamiento basado en casos (CBR) [1]. Hoy en día es un hecho la creciente utilización de los agentes y sistemas multiagente para el desarrollo de aplicaciones en entornos dinámicos y flexibles tales como la web, sistemas de control, robótica, etc. En este documento se presenta una arquitectura distribuida cuya principal característica es la utilización de agentes CBR-BDI [7] para la monitorización y la evaluación del intercambio de CO<sub>2</sub> existente entre la atmósfera y la superficie de las aguas oceánicas, basado en la utilización de los datos obtenidos a través de técnicas de teledetección. Las características del problema de la interacción mar-aire, explicado en detalle en la sección 2 de este artículo, hacen que parezca adecuada la utilización de un SMA para la generación de modelos dinámicos de forma automática.

Denominamos agente CBR-BDI [7] a un agente deliberativo BDI que incorpora un sistema CBR como motor de razonamiento. Estos agentes son capaces de aprender, a partir de un conocimiento inicial, interactuar autónomamente con el entorno y con los usuarios del sistema y se adaptan a las necesidades del entorno.

Existen muchas arquitecturas que permiten construir agentes deliberativos y muchas de ellas se basan en el modelo BDI. En el modelo BDI la estructura interna de un agente y su capacidad de elección se basan en aptitudes mentales. Esto tiene la ventaja de utilizar un modelo natural (humano) y de alto nivel de abstracción. El modelo BDI utiliza Beliefs (Creencias) como aptitudes informacionales, Desires (Deseos) como aptitudes motivacionales e Intentions (Intenciones) como aptitudes deliberativas de los agentes. El método propuesto en [6, 11] facilita la incorporación de sistemas CBR como mecanismo deliberativo en agentes BDI, proporcionando un aprendizaje, adaptación y un mayor grado de autonomía que una arquitectura BDI pura [11]. El CBR es un paradigma que se basa en la idea de que los problemas similares tienen soluciones similares. De esta forma, un problema nuevo se resuelve consultando en la memoria experiencias similares que se hubiesen resuelto en el pasado.

Uno de los mayores problemas para el desarrollo de una arquitectura basada en sistemas multiagente es que, hoy en día, se carece de estándares claros o metodologías totalmente desarrolladas que marquen los pasos para realizar un análisis y diseño. En la actualidad existen varias metodologías ZEUS [16], Gaia [22], AUML [2], MaSE [9], MAS-CommonKADS [12], MESSAGE [10]. El problema de estas metodologías es que en general no se encuentran totalmente desarrolladas presentan ciertas limitaciones. En este artículo se opta por realizar un análisis y diseño de nuestro SMA utilizando una combinación de elementos de las metodologías Gaia y Agent Unified Modeling Language (AUML). Gaia es una metodología sencilla que nos permite realizar un primer análisis y un primer diseño con el que afrontar el problema a un nivel general. La gran ventaja es que se puede realizar un estudio rápido y poco detallado. Por el contrario, el problema aparece cuando al finalizar el diseño nos encontramos con que todavía tenemos un nivel de abstracción demasiado alto. Por su parte AUML proporciona mecanismos que permiten obtener un diseño lo suficientemente preciso como para pasar directamente a la implementación, pero presenta el problema de comenzar el estudio del problema a un nivel demasiado específico y detallado. Nuestra idea consiste en aprovechar las ventajas de ambas metodologías realizando un primer análisis y diseño con Gaia y posteriormente realizar los cambios oportunos para continuar con un diseño detallado AUML.

La utilización de dispositivos inalámbricos, tales como PDAs facilita enormemente el trabajo de los oceanógrafos, permitiendo una gran libertad de movimientos e independencia de la localización. Cada oceanógrafo dispone de un agente de usuario situado en un dispositivo ligero que le permite acceder al SMA.

En la siguiente sección nos ocuparemos de explicar el problema oceánico/atmosférico que motiva la mayor parte de esta investigación. En la sección tres se muestra el sistema multiagente desarrollado, haciendo especial incidencia en los agentes CBR-BDI. En la sección 4 se analiza el agente CBR-BDI. Finalmente se presentan unos resultados preliminares y las conclusiones obtenidas.

## **2. Problema de la Interacción Aire Mar**

Uno de los factores más preocupantes que influyen en el comportamiento climatológico es la cantidad de CO<sub>2</sub> presente en la atmósfera. El CO<sub>2</sub> es uno de los

gases de efecto invernadero, que contribuye a que la Tierra tenga una temperatura habitable, siempre y cuando se mantenga en unas cantidades determinadas [18]. Tradicionalmente se ha considerado que el principal sistema regulador de la cantidad de CO<sub>2</sub> en la atmósfera es la fotosíntesis y respiración realizado por las plantas. Sin embargo, y especialmente gracias a la utilización de técnicas de teledetección se ha descubierto que el papel que juega el océano en la regulación de las cantidades de carbono es muy importante y es una magnitud que permanece indefinida [19]. La tecnología actual permite obtener datos y realizar cálculos impensables hasta hace poco tiempo. Estos datos proporcionan conocimiento sobre las fuentes de origen y disminución de CO<sub>2</sub> así como de sus causas [14], lo que supone la posibilidad de realizar predicciones sobre el comportamiento futuro del CO<sub>2</sub>.

El sistema multiagente que se presenta tiene como objetivo modelar el flujo de CO<sub>2</sub> intercambiado entre la atmósfera y las aguas de la superficie oceánica. Los principales parámetros a tener en cuenta a la hora de modelar el intercambio de CO<sub>2</sub> entre la atmósfera y la superficie oceánica son: la temperatura del agua y del aire, la salinidad del agua, las presiones atmosféricas e hidrostáticas, la presencia de nutrientes y el vector velocidad del viento (en módulo y dirección). Estos parámetros pueden obtenerse a través de mediciones in-situ realizadas desde barcos oceanográficos o bien a través de imágenes de satélite. La información de satélite es vital para la construcción de modelos oceanográficos, y en este caso, para generar estimaciones de los flujos de CO<sub>2</sub> entre el aire y el mar con una alta resolución tanto espacial como temporal, utilizando modelos de inteligencia artificial que pueden ser contrastados con la realidad por medio de muestreos directos in situ sobre el CO<sub>2</sub> de la superficie del océano. Será necesario utilizar sistemas distribuidos automatizados y capaces de incorporar nuevo conocimiento para poder manejar todos los datos útiles y para crear modelos diarios en un tiempo y coste razonables. Nuestra propuesta se presenta en la siguiente sección.

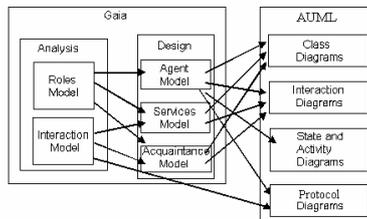


Fig. 2 Metodología seguida en el desarrollo

### 3. Sistema Multiagente para la Interacción Aire Mar

El objetivo final de este proyecto es construir un gran sistema multiagente que se divida en subsistemas, de tal forma que cada subsistema se encargue de modelar el intercambio de CO<sub>2</sub> en una zona del océano con unas determinadas características. Cada subsistema se comunicará con el resto de los subsistemas, obteniendo un intercambio de información que ayude a la creación de modelos. Dado que se trata de

un proyecto de gran magnitud, en este trabajo se presenta el estudio y desarrollo de un subsistema, cuya arquitectura final puede observarse en la Figura 6. Dado que el subsistema debe ser modelado en términos de agentes, se opta por utilizar una metodología de análisis y diseño intermedia entre Gaia [22] y AUML [2], que trata de aprovechar las ventajas de cada una de ellas. A través de Gaia se consigue un rápido y eficaz análisis del problema utilizando criterios de organización y un posterior diseño. El resultado tras aplicar Gaia consiste en un diseño a un nivel de abstracción elevado. En ese momento, tal y como se muestra en la Figura 2, se adapta el diseño Gaia de forma que puedan ser aplicadas técnicas AUML, y sea posible obtener un diseño detallado de bajo nivel.

### 3.1. Análisis y Diseño Gaia

Gaia es una metodología de carácter muy general, con lo que es aplicable a un rango muy amplio de sistemas multiagente, y que además permite tener un amplio conocimiento de los sistemas multiagente tanto a nivel organizativo (social) como a nivel de detalle de cada agente [22]. El análisis Gaia permite obtener de una forma rápida los roles que se juegan en un sistema, así como las relaciones que se establecen entre dichos roles. Para ello propone dos modelos, el modelo de roles y el modelo de interacción. Partiendo de los requisitos del problema planteado se decide utilizar seis roles: El rol STORING se encarga de obtener y almacenar datos de carácter permanente en las bases de datos. El rol PROCESSING transforma las imágenes de satélite en casos. El rol DATACAPTURING obtiene datos de los Vessels. El rol CONSTRUCTAPARTIALCO<sub>2</sub>MODEL se ocupa de generar modelos. El rol OBTAINCO<sub>2</sub>EXCHANGE calcula la tasa de intercambio de CO<sub>2</sub> utilizando los modelos disponibles. El rol AUTOEVALUATION evalúa un modelo contrastando sus resultados con datos reales obtenidos por los sensores de los Vessels. El rol PROCESSINGINFORMATION permite a un usuario interactuar con el sistema.

Role Schema: CONSTRUCTAPARTIALCO <sub>2</sub> MODEL (CPCM)
Description: Constructs a model from the available data. Stores the model information. Allows consults on models stored data.
Protocols and Activities: <u>GenerateModel</u> , RequestConstructData, SendModelData, <u>StoreModelData</u> , <u>ConsultModelData</u>
Permissions: Reads: Model BD  Changes: Model BD  Generates: new model
Responsibilities: Liveness: CONSTRUCTMODEL: [RequestConstructData] ( <u>GenerateModel</u> ) <sup>w</sup> <u>StoreModelData</u> [SendModelData] CONSULTMODEL: <u>ConsultModelData</u> SendModelData Safety: Successful connection with BD established

Fig. 3. Modelo de roles Gaia para el rol CPM.

La Figura 3 muestra el modelo de roles para el rol CPM, con la especificación de sus atributos característicos: Responsabilidades, permisos, actividades y protocolos [22]. Por su parte, el modelo de interacción muestra las dependencias y relaciones entre roles. Por cada interacción entre dos roles se tiene un protocolo. Para nuestro SMA se ha decidido utilizar los protocolos de interacción: ObtainNewModelSuper, ObtainNewModelAuto, ObtainNewModelStoring, ObtainConstructData, ObtainStModel, ObtainStore, ObtainVessel, ObtainEvaluationSuper, ObtainEvaluationDC, Activate/Deactivate Sensors, Delete EPROM, ChangeStore, ChangeCase, ObtainVesselData, ObtainStExchange y ObtainInsituData. Cada interacción se modela de forma que se muestran los roles implicados y los distintos protocolos que utilizar para llevar a cabo la interacción.

En cuanto al diseño Gaia, se definen los agentes del sistema y se reduce el nivel de abstracción para que puedan ser aplicadas técnicas tradicionales. Se consideran tres modelos: modelo de agentes, modelo de servicios y modelo de confianza [22]. La Figura 4 muestra el modelo de agentes para nuestro sistema. Cada agente se encarga de jugar una serie de roles. Por ejemplo el agente Store se va a encargar de jugar los roles STORING y PROCESSING, y en tiempo de ejecución será necesario tener al menos una instancia del rol STORING y una instancia del rol PROCESSING.

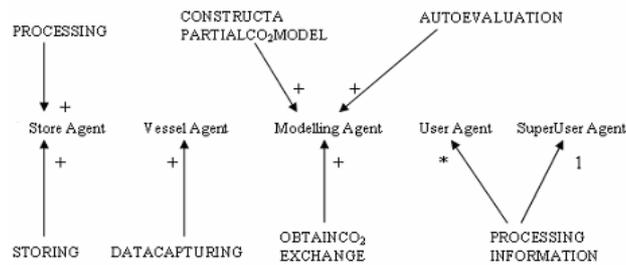


Fig. 4. Modelo de Agentes Gaia para nuestro SMA.

### 3.2. Diseño detallado AUML

AUML es una metodología que trabaja a un nivel de detalle elevado, quizás demasiado elevado en sus etapas iniciales para problemas de gran magnitud, como es el caso concreto en el que nos encontramos. Nuestra propuesta trata de utilizar el análisis de alto nivel obtenido con Gaia y llevar a cabo un diseño AUML a bajo nivel, a un nivel suficiente de detalle como para proceder a la implementación. Hay tres conceptos que varían ligeramente con respecto a su significado de la metodología Gaia a la metodología AUML: rol, servicio y capacidad [2]. Con el diseño AUML obtenemos diagramas de clases para cada agente, diagramas de colaboración o secuencia, diagramas de estado y diagramas de actividad [2].

Por motivos de espacio, en este trabajo se presentan algunos de los diagramas más representativos obtenidos con el diseño de bajo nivel. La Figura 5 muestra el diagrama de clases para el agente CBR-BDI de nuestro sistema. Se trata del agente más importante dentro del sistema distribuido y será estudiado más en detalle en la siguiente sección. El agente posee seis capacidades y ofrece cuatro servicios que pueden utilizar el resto de agentes.

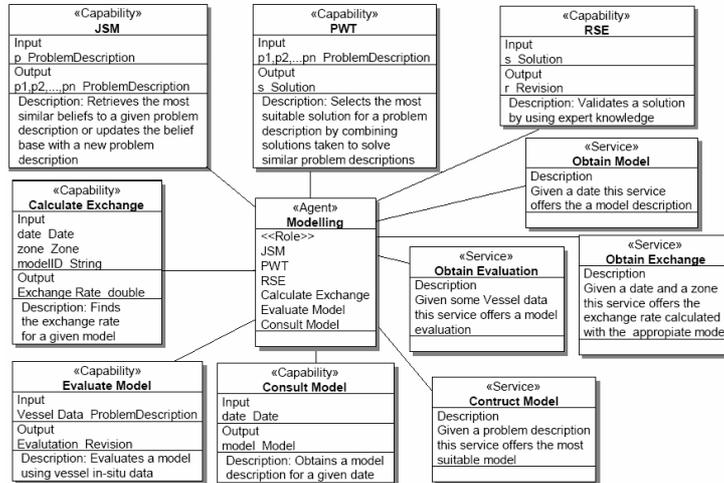


Fig. 5. Diagrama de clases para el agente Modelling.

Una vez finalizado el diseño se procede a la implementación y se utiliza para ello la herramienta Jadex [17], que es una herramienta que incorpora el modelo BDI a los agentes Jade, y la herramienta Jade [5]. De esta forma los agentes Jadex trabajan con conceptos de beliefs (creencias), goals (metas) y plans (planes). Jadex tiene la ventaja de permitir al programador introducir sus propios mecanismos deliberativos. En nuestro caso este mecanismo va a ser un sistema CBR. El agente Modelling es un agente Jadex, mientras que el resto de agentes se construyen con Jade. Los mecanismos de comunicación son los definidos por Jade [5, 17].

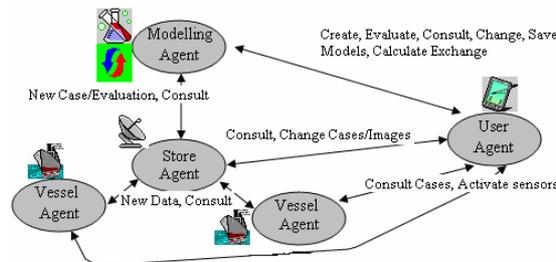


Fig. 6. SMA para una zona concreta del Océano.

Por el momento se ha desarrollado el prototipo correspondiente a un subsistema como el mostrado en la Figura 6, en la que es posible observar como un agente Modelling se encarga de la creación y evaluación de modelos en función de los datos recibidos desde los agentes Store, Vessel o User. Los modelos permiten predecir el intercambio de CO<sub>2</sub> entre la superficie del océano y la atmósfera. La Figura 6 muestra también como el agente Store del subsistema se encarga de procesar las imágenes de satélite y transformarlas para que el sistema pueda utilizarlas (envía al agente Modelling nuevas descripciones de problema de casos). Cada agente Vessel se instala en un navío y recoge información in-situ que permite evaluar los modelos creados por el agente Modelling. El agente usuario puede interactuar con cualquier agente.

#### 4. Agente CBR-BDI

El componente principal de la arquitectura propuesta es el agente encargado de generar modelos. Se trata de un agente deliberativo BDI [21] que utiliza un sistema CBR como motor de razonamiento. De esta forma la generación de un modelo supone la ejecución de un ciclo CBR formado por cuatro etapas secuenciales: retrieval, reuse, revise y retain. Un sistema CBR maneja casos. Un caso es una experiencia pasada, que incluye una descripción del problema, la solución utilizada para resolverlo y el resultado obtenido. Construir un agente CBR-BDI [7, 11] supone por un lado integrar el concepto de caso en la estructura del modelo BDI y por otro lado que el agente BDI ejecute el ciclo CBR. El agente CBR-BDI utiliza sus estructuras internas para representar un caso: creencias, deseos e intenciones. Creencias para representar la descripción del problema, una creencia de tipo ProblemDescription (en nuestro SMA el sistema necesita transformar las imágenes de satélite. Para ello el agente Store utiliza algoritmos [8, 13] obteniendo una descripción de problema como la mostrada en la Tabla 1: fecha, latitud, longitud, salinidad, clorofila, viento, presiones parciales y temperaturas). Deseos para representar las metas que el agente persigue (predecir el flujo de CO<sub>2</sub> intercambiado, calcular los mejores parámetros de predicción para distintos tamaños de ventana, y calcular la mejor ventana de predicción para un determinado % de error máximo permitido). Intenciones o secuencias de acciones para alcanzar una o varias metas en función de las creencias de las que se dispone.

**Table 1.** Atributos de los casos.

Case Field	Measurement
DATE	Date (dd/mm/yyyy)
LAT	Latitude (decimal degrees)
LONG	Longitude (decimal degrees)
SST	Temperature (°C)
S	Salinity (unitless)
WS	Wind strength (m/s)
WD	Wind direction (unitless)
Fluo_calibrated	Fluorescence calibrated with chlorophyll
SW pCO <sub>2</sub>	Surface partial pressure of CO <sub>2</sub> (micro Atmospheres)
Air pCO <sub>2</sub>	Air partial pressure of CO <sub>2</sub> (micro Atmospheres)
Flux of CO <sub>2</sub>	CO <sub>2</sub> exchange flux (Moles/m <sup>2</sup> )

La forma de implantar el ciclo CBR dentro de un agente BDI consiste en que el agente BDI ejecute las etapas del ciclo CBR como capacidades [11]. Un agente BDI posee capacidades (funcionalidades que el agente es capaz de realizar por sí mismo sin necesidad de contactar con otros agentes) y ofrece servicios a otros agentes. El agente Modelling ofrece cuatro servicios (Obtain Exchange, Obtain Model, Obtain Evaluation y Construct Model) y posee seis capacidades (Forecast Exchange Rate, Evaluate Model, Consult Model, Jacobean Sensivity Matriz - JSM, Pondered Weigh Technique - PWT y Revision Simulated Equation - RSE). Las tres últimas capacidades constituyen el ciclo CBR, y su funcionamiento cuando se recibe una nueva descripción de problema es el que a continuación pasa a describirse: En la etapa de recuperación la capacidad JSM permite obtener aquellos casos con una

descripción de problema más similar a la del caso actual [15]. Para ello se utiliza una red neuronal que permite obtener las componentes principales de la memoria de casos, y conocer cuales de las variables de entrada (que forman la descripción del problema) tienen mayor influencia a la hora de crear clusters. Si consideramos JSM una matriz NxM, el elemento  $S_{ki}$  en la matriz representa la sensibilidad (influencia) de la salida  $k$  sobre la entrada  $i$ , como muestra la ecuación 1.

$$S_{ki} = \frac{\partial y_k}{\partial x_i} = \frac{\partial f_k(net_k)}{\partial x_i} = \frac{\partial f_k(net_k)}{\partial net_k} \frac{\partial net_k}{\partial y_j} \frac{\partial y_j}{\partial net_j} \frac{\partial net_j}{\partial x_i} = \frac{\partial f_k(net_k)}{\partial net_k} \left( \sum_{j=1}^H w_{kj} \frac{\partial f_j(net_j)}{\partial net_j} w_{ji} \right) \quad (1)$$

Donde  $w_{ij}$  es el peso de la conexión entre la neurona de entrada  $i$  y la neurona oculta  $j$ .  $w_{kj}$  es el peso de la conexión entre la neurona oculta  $j$  y la neurona de salida  $k$ .  $y_k$  es la salida de la neurona  $k$  de la capa de salida  $y_k = f_k(net_k)$ , y  $y_j$  es la salida de la neurona  $j$  de la capa oculta  $y_j = f_j(net_j)$ .  $f_j$  es la función de activación de la neurona  $h$  y  $net_j$ ,  $net_k$  se obtienen a partir de las ecuaciones 2 y 3 respectivamente. Siendo  $H$  el número de neuronas de la capa oculta, y  $\theta_j$  y  $\theta_k$  los umbrales permitidos. Una vez obtenidas las componentes principales se busca en la memoria de casos aquellos con componentes principales más similares a la descripción del problema actual. Para ello se utiliza un algoritmo de similitud basado en la función coseno.

$$net_j = \sum_{i=1}^N w_{ji} x_i + \theta_j \quad (2)$$

$$net_k = \sum_{j=1}^H w_{kj} y_j + \theta_k \quad (3)$$

En la etapa de adaptación se combinan las soluciones de los casos similares recuperados para obtener una solución adecuada al problema actual. Se efectúa mediante la capacidad PWT, en la que los casos se ponderan [15] y se asigna el mayor peso al caso con una descripción de problema más similar a la del caso actual.

$$p^* = \frac{1}{\sum_{r=1}^Z e^{-|a-r|}} p^r \quad (4)$$

La ecuación 4 muestra el proceso de adaptación, donde  $p^*$  es la predicción,  $Z$  el número de casos similares y  $p^r$  son las predicciones (soluciones) correspondientes a cada caso pasado,  $a$  es la medida de mínima similitud para los casos recuperados y  $r$  es la medida de similitud para el caso  $r$ -ésimo. Se utiliza una red neuronal que se entrena con los casos similares recuperados y sus respectivas soluciones. Una vez entrenada se le proporciona el caso actual y se obtiene su solución.

En la etapa de revisión cada modelo se revisa en la capacidad RSE. Para ello se utiliza la ecuación 5 con los coeficientes de intercambio de Wanninkhof [20].

$$F = kso(pCO_2SW - pCO_2AIR) \quad (5)$$

Finalmente, en la etapa de aprendizaje se compara el valor  $F$  con el valor de la predicción. Si la diferencia es inferior al 10%, el caso se almacena en la base de creencias. La etapa de aprendizaje también se efectúa en la capacidad RSE.

## 5. Resultados y conclusiones

El sistema descrito ha sido probado con datos recogidos en el Océano Atlántico Norte durante 2004. A pesar de que el sistema no se encuentra totalmente operativo y que el objetivo del proyecto es el de construir un prototipo para investigación y no una herramienta comercial, los resultados iniciales han sido muy satisfactorios desde los puntos de vista técnico y científico. La construcción del sistema distribuido ha sido relativamente fácil gracias a la utilización de bibliotecas CBR-BDI previamente desarrolladas [3, 4, 7]. Desde el punto de vista de la ingeniería del software, AUML [2] y Gaia [22] proporcionan un marco adecuado para el análisis y diseño de sistemas distribuidos basados en agentes. El formalismo definido en [11] facilita un paso directo entre la definición de agente y la construcción CBR. La utilización de dispositivos inalámbricos ha sido valorada positivamente por los oceanógrafos.

La interacción con el SMA por parte de desarrolladores y oceanógrafos ha sido continua durante su construcción y periodo de prueba. El sistema ha sido probado bajo condiciones de simulación de tal forma que los agentes Store y Vessel se han construido incluyendo simuladores que generan casos a partir de los datos reales obtenidos en el Océano Atlántico ( $\pm 37N$ ,  $25W$ ). Bajo estas condiciones los modelos propuestos por el sistema multiagente han sido cada vez más precisos. Sin embargo, si el número de casos manejado es muy elevado la eficiencia del sistema disminuye. La Figura 7 muestra una comparativa entre los datos reales y las predicciones realizadas por el SMA trabajando con datos relativos a meses de 2003-2004.

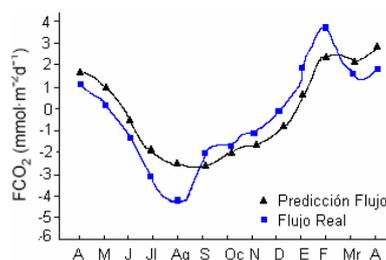


Figura 7. Flujo real de CO<sub>2</sub> y predicción de flujo.

Como puede observarse en la Figura 7 las predicciones realizadas por el SMA son bastante precisas (9 de los 12 modelos fueron aceptados como válidos). El SMA realiza predicciones basándose en su experiencia y en la similitud con situaciones pasadas. Claramente la precisión de las predicciones mejora a medida que el número de casos disponibles aumenta. Por otro lado es necesario controlar que el número de casos manejado no sea excesivo. Para ello se ha utilizado un sistema de prioridades basado en crear una estructura piramidal de eficiencias que permite clasificar los casos en base a la eficiencia obtenida. El sistema facilita la incorporación de nuevos agentes que utilicen diferentes técnicas de modelado y estrategias de aprendizaje de forma que se puedan realizar experimentos adicionales con nuevas técnicas y puedan ser comparados con los resultados iniciales presentados en este documento.

**Agradecimientos.** Este trabajo se ha llevado a cabo gracias a los proyectos MCYT TIC2003-07369-CO2-02 y CAXIS de PML.

## Referencias

1. Aamodt A. and Plaza E.: Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, AICOM. Vol. 7., pp 39-59 (1994)
2. AUML. [www.auml.org](http://www.auml.org) (2006)
3. Bajo J. and Corchado J.M (2005). Evaluation and monitoring of the air-sea interaction using a CBR-Agents approach. LNAI 3620, pp. 50-62. Springer Verlag
4. Bajo J. and Corchado J.M. (2006). Multiagent architecture for monitoring the North-Atlantic carbon dioxide Exchange rate. CAEPIA 2005. LNCS 4177, pp 321-330
5. Bellifime, F. Poggi, A. and Rimasa, G. (2001) JADE: a FIPA2000 compliant agent development environment. Agents 2001 pp. 216-217.
6. Bratman, M.E. (1987). Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A.
7. Corchado J. M. and Laza R. (2003). Constructing Deliberative Agents with Case-based Reasoning Technology, International Journal of Intelligent Systems. Vol 18, No. 12, December. pp.: 1227-1241
8. Dransfeld S., Tatnall A.R., Robinson I. S. and Mobley C.D. (2005). Prioritizing ocean colour channels by neural network input reflectance perturbation. International Journal of Remote Sensing, 26, (5), 043-1048.
9. DeLoach, S. (2001) Analysis and Design using MaSE and AgentTool. Proceedings of the 12<sup>th</sup> Midwest Artificial Intelligence and Cognitive Science Conference (MAICS).
10. EURESCOM (2001) MESSAGE: Methodology for engineering systems of software agents. Technical report P907-T11, EURESCOM.
11. Glez-Bedia M., Corchado J. M., Corchado E. S. and Fyfe C. (2002) Analytical Model for Constructing Deliberative Agents, Engineering Intelligent Systems, Vol 3: pp. 173-185.
12. Iglesias, C., Garijo, M., Gonzalez J.C. and Velasco J. R. (1998) Analysis and Design using MAS-CommonKADS. Intelligent Agents IV LNAI Volume 1365 Springer Verlag.
13. Lavencer S.J., Pinkerton M.H., Froidefond J.M., Morales J., Aiken J. and Moore J.F. (2004) SeaWiFS validation in European coastal waters using optical and bio-geochemical measurements. International Journal of Remote Sensing, Vol. 25, No. 7-8, pp. 1481-1488
14. Lefevre N., Aiken J., Rutllant J., Daneri G., Lavender S. and Smyth T. (2002) Observations of pCO<sub>2</sub> in the coastal upwelling off Chile: Sapatial and temporal extrapolation using satellite data. Journal of Geophysical research. Vol. 107, no. 0
15. Montaña Moreno J.J. and Palmer Pol A. (2002). Artificial Neural Networks, opening the black box. Metodología de las Ciencias del Comportamiento 4(1) 77-93.
16. Nwana H.S., Ndumu, D.T. y Collins J.C. (1999) ZEUS: A Toolkit for Building Distributed Multi-Agent Systems. Applied Artificial Intelligence Journal, vol 1, n°13, pp. 129-185.
17. Pokahr, A., Braubach, L. and Lamersdorf W. (2003) Jadex: Implementing a BDI-Infrastructure for JADE Agents, in: EXP - In Search of Innovation (Special Issue on JADE), Vol 3, Nr. 3, Telecom Italia Lab, Turin, Italy, September 2003, pp. 76-85.
18. Sarmiento J. L. and Dender M. (1994) Carbon biogeochemistry and climate change. Photosynthesis Research, Vol. 39, 209-234.
19. Takahashi T., Olafsson J., Goddard J. G., Chipman D. W. and Sutherland S. C. (1993) Seasonal Variation of CO<sub>2</sub> and nutrients in the High-latitude surface oceans: a comparative study. Global biochemical Cycles. Vol. 7, no. 4. pp 843-878.
20. Wanninkhof, R. (1992). Relationship between wind speed and gas exchange over the ocean. J. Geophys. Res. 97, 7, 373-7, 383.
21. Wooldridge M. and Jennings N.R. (1995) Agent Theories, Architectures, and Languages: a Survey. Wooldridge and Jennings, editors, Intelligent Agents, Springer-Verlag, pp. 1-22.
22. Wooldridge M., Jennings N.R. and Kinny D. (2000) The Gaia Methodology for Agent-Oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems, 3 (3). pp. 285-312.