13. Wang R, Bhanu B, Chen H (2005) An integrated prediction model for biometrics. In: Proc. Audio- and Video-based Biometric Person Authentication, New York, pp. 355–364.
14. Johnson AY, Sun J, Boick AF (2003) Using similarity scores from a small gallery to estimate recognition performance for large galleries. In: Proc. IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures, pp. 100–103.
15. Grother P, Phillips PJ (2004) Models of large population recognition performance. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 68–75.
16. Lindenbaum M (1995) Bounds on shape recognition performance. J Pattern Analysis and Machine Intelligence, vol. 17, no. 7, pp. 666–680.
17. Lindenbaum M (1997) An integrated model for evaluating the amount of data required for reliable recognition. J Pattern Analysis and Machine Intelligence, vol. 19, no. 11, pp. 1251–1264.
18. Boshra M, Bhanu B (2000) Predicting performance of object recognition. J Pattern Analysis and Machine Intelligence, vol. 22, no. 9, pp. 956–969.
19. Boshra M, Bhanu B (2001) Predicting an upper bound on SAR ATR performance. J IEEE Trans. on Aerospace and Electronic Systems, vol. 37, no. 3, pp. 876–888.
20. Wang R, Bhanu B (2005) Learning models for predicting recognition performance. In: Proc. IEEE International Conf. on Computer Vision, vol. 2, pp. 1613–1618.
21. Tan X, Bhanu B (2002) Robust fingerprint identification. In: Proc. IEEE Int. Conf. on Image Processing, vol. 1, pp. 277–280.
22. Mood AM, Graybill FA, Boes DC (1974) Introduction to the Theory of Statistics. McGraw,Hill.
23. Bhanu B, Boshra M, Tan X (2000) Logical templates for feature extraction in fingerprint images. In: Proc. Int. Conf. on Pattern Recognition, vol. 3, pp. 850–854.
24. Figueiredo MAT, Jain AK (2002) Supervised learning of finite mixture models. J IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp. 381–396.

8

A CBR Agent for Monitoring the Carbon Dioxide Exchange Rate from Satellite Images

J.M. Corchado, J. Aiken, and J. Bajo

Departamento Informática y Automática
Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
corchado@usal.es
Centre for Air-Sea Interactions and fluxes
Plymouth Marine Laboratory, Prospect Place, Plymouth, PL1 3 DH, UK
jai@mail.pml.ac.uk
Universidad Pontificia de Salamanca
Compañía 5, 37002, Salamanca, Spain
jbajope@upsa.es

Summary. This work presents a multiagent system for evaluating automatically the interaction that exists between the atmosphere and the ocean surface. monitoring and evaluating within the ocean carbon dioxide exchange process is a function requiring working with a great amount of data: satellite images and in situ Vessel's data. The system presented in this work focuses on Ambient Intelligence (AmI) technologies since the vision of AmI assumes seamless, unobtrusive, and often invisible but also controllable interactions between humans and technology. The work presents the construction of an open multiagent architecture which, based on the use of deliberative agents incorporating Case-Based Reasoning (CBR) systems, offers a distributed model for such an interaction. This work also presents an analysis and design methodology that facilitates the implementation of CBR agent-based distributed artificial intelligent systems. Moreover, the architecture takes into account the fact that the working environment is dynamic and therefore it requires autonomous models that evolve overtime. In order to resolve this problem an intelligent environment has been developed, based on the use of CBR agents, which are capable of handling several goals, constructing plans from the data obtained through satellite images and research Vessels, acquiring knowledge, and of adapting to environmental changes, are incorporated. The artificial intelligence system has been successfully tested in the North Atlantic ocean, and the results obtained will be presented within this work.

8.1 Introduction

Ambient intelligent environments are characterized by their ubiquity, transparency, and intelligence [2]. The agents and multiagent systems (MASs) have become increasingly relevant for developing distributed and dynamic

J.M. Corchado et al.: *A CBR Agent for Monitoring the Carbon Dioxide Exchange Rate from Satellite Images*, Studies in Computational Intelligence (SCI) **73**, 213–246 (2008)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2008

intelligent environments. Agents and MAS have become increasingly relevant for developing applications in the internet, personalized user interfaces, oceanography, control systems, or robotic environments. Agents can be characterized through their capacities in areas such as autonomy, reactivity, proactivity, social abilities, reasoning, learning, and mobility. These capacities which can be modelled in various ways, using different methodologies [47], make the agents and MAS highly suited to intelligent environments. One of the possibilities to model the reasoning capacity is to use Case-Based Reasoning (CBR). In this work we present a distributed architecture whose main characteristic is the use of Case-Based Reasoning–Beliefs Desires Intentions (CBR–BDI) agents [13], which will be named in this chapter as CBR agents. These agents are capable of learning, from their initial knowledge and by interacting autonomously with their environment and with users of the system, adapting themselves accordingly. Both, the developed multiagent architecture and the Modelling agent are described in detail. The development of ambient intelligent systems is normally complicated due it novel connotations, this work presents a practical way for analyzing and designing MAS at the same time of describing the distributed ambient intelligent system developed.

The mission of the intelligent environment presented in this work, is to globally monitor the interaction between the ocean surface and the atmosphere, facilitating the work of oceanographers. Initially, the system is being used in order to evaluate and predict the amount of carbon dioxide (CO_2) absorbed or expelled by the ocean in the North Atlantic [6, 7, 15]. The main purpose of this work is to obtain an architecture that enables the construction of open, distributed, and dynamic systems capable of growing in dimension and of adapting their knowledge according to different changes that take place in their environment. There are many different architectures for constructing deliberative agents and many of them are based on the BDI model. In the BDI model, the internal structure of an agent and its capacity to choose is based on mental aptitudes. This has the advantage that it uses a natural model (human) and a high level of abstraction. The BDI model uses the agent' Beliefs as informational aptitudes, its Desires as motivational aptitudes and its Intentions as deliberative aptitudes. The method proposed in [5, 12, 13] facilitates the incorporation of CBR systems as a deliberative mechanism within BDI agents, allowing them to learn and adapt themselves, and lending them a greater level of autonomy than pure BDI architecture [25]. Moreover, this proposal differs from others [10, 22, 31, 36, 46] in that it proposes direct mapping between the concept of the agents and their implementation. CBR systems are also highly suited to some of the tasks in the study of carbon dioxide exchange between the ocean and the atmosphere, such as the interpretation of satellite images [40].

One of the major problems in the development of an architecture based on MAS is that there are currently no clear standards or well developed methodologies for defining the steps of analysis and design that need to be taken in order to define an intelligent environment. There are at present a number of methodologies: Gaia [48], AUML [8, 34, 35], MAS-CommonKADS [26], MaSE [18], ZEUS [33], MESSAGE [21]. The problem with these methodologies is that they are generally not fully developed and present a number of limitations. For this study, we have decided to opt for a combination of elements from Gaia and Agent Unified Modelling Language (AUML) for our MAS. Gaia is a simple methodology that allows us to carry out a preliminary analysis and design with which to confront the problem at a general level. The great advantage is that we can carry out a rapid, broad study but problems arise when the design is at its completion because there tends to be an overly high level of abstraction. AUML, on the other hand, offers mechanisms which allow us to obtain a design that is sufficiently precise and able to pass directly to the implementation stage, but has the disadvantage of being too precise and detailed for the preliminary stages. Our goal is to take advantage of both methodologies by carrying out a preliminary analysis and design with Gaia and later on to carry out the appropriate changes by using a detailed AUML design. In this way we are able to obtain both a generalized vision of the problem in terms of organization, and a detailed MAS description which helps enormously in the development of such a research project.

In order to implement the BDI agents, various tools are used. One interesting tool is Jadex [41], which incorporates the BDI architecture into Jade agents [9]. In this sense, Jadex agents work with concepts of beliefs, goals, and plans, all of which become objects which can be created and manipulated within the agent. The beliefs represent any type of Java object and are stored in the beliefs' database. The goals represent specific motivations that influence the behaviour of the agent. The plans are procedures written in Java which are executed in order to reach the goals. Jadex has the advantage to allow the programmer to introduce his own deliberative planning mechanisms. In our case, this mechanism will be a CBR system. In addition, it offers all the advantages of Jade and allows the use of Jade and Jadex agents within the same MAS. The MAS incorporates "lightweight" agents that can live in mobile devices, such as phones, personal digital assistants (PDAs), etc. These agents make it possible for a oceanographer to interact with the MAS in a very simple way, downloading and installing a personal agent in his mobile phone or PDA.

In Sect. 8.2, we will explain the various relationships that can be established between CBR and BDI concepts. In Sect. 8.3 we will describe the oceanic/atmospheric problem that has led to most of this research. In Sect. 8.4, the MAS developed will be described, paying special attention to the CBR agents. Finally, some preliminary results and the conclusions will be presented.

8.2 CBR–BDI Agents

Ambient Intelligence has been widely studied and different artificial intelligence techniques have been applied. The application of agents and MAS facilitates taking advantage of the agent capabilities, such as mobility, proactivity, or social abilities, as well as the possibility of solving problems in a distributed way. Agents, in the context of an intelligent environment, must be able to respond to events, take the initiative according to their goals, communicate with other agents, interact with users, and make use of past experiences to find the best ways to achieve goals. There are many architectures for constructing deliberative agents and many of them are based on the BDI model [27,28]. In the BDI model, the internal structure of an agent and its capacity to choose, is based on mental aptitudes: agent behaviour is composed of beliefs, desires, and intentions [42]. The beliefs represent its information state, what the agent knows about itself and its environment. The desires are its motivation state, what the agent is trying to achieve. And the intentions represent the agent's deliberative states. Intentions are sequences of actions; they can be identified as plans. A BDI architecture has the advantage that it is intuitive and relatively simple to identify the process of decision-making and how to perform it. Furthermore, the notions of belief, desire, and intention are easy to understand. On the other hand, its main drawback lies in finding a mechanism that permits its efficient implementation.

CBR is a type of reasoning based on the use of past experiences [28]. The purpose of CBR systems is to solve new problems by adapting solutions that have been used to solve similar problems in the past. The fundamental concept when working with CBR is the concept of case. A case can be defined as a past experience, and is composed of three elements: A problem description which describes the initial problem, a solution which provides the sequence of actions carried out in order to solve the problem, and the final state which describes the state achieved once the solution was applied. A CBR system manages cases (past experiences) to solve new problems. The way in which cases are managed is known as the CBR cycle.

The deliberative agents, proposed in the framework of this investigation, use this concept to gain autonomy and improve their problem-solving capabilities. The method proposed in [13] facilitates the incorporation of CBR systems as a deliberative mechanism within BDI agents, allowing them to learn and adapt themselves, lending them a greater level of autonomy than pure BDI architecture [12]. Accordingly, CBR agents implemented using CBR systems could reason autonomously and therefore adapt themselves to environmental changes. The CBR system is completely integrated within the agents' architecture. The CBR–BDI agents incorporate a "formalism" which is easy to implement, in which the reasoning process is based on the concept of intention. Intentions can be seen as cases, which have to be retrieved, reused, revised, and retained. This makes the model unique in its conception and reasoning capacities. The structure of the CBR system has been designed around the concept of a case. A direct relationship between CBR systems and BDI agents can also be established if the problems are defined in the form of states and actions.

Case: <Problem, Solution, Result>	**BDI agent**
Problem: initial_state	Belief: state
Solution: sequence of <action, [intermediate_state]>	Intention: sequence of <action>
Result: final_state	Desire: set of <final_state>

The relationship between CBR systems and BDI agents can be established by implementing cases as beliefs, intentions, and desires which lead to the resolution of the problem. As described in [7,16], in a CBR–BDI agent, each state is considered as a belief; the objective to be reached may also be a belief. The intentions are plans of actions that the agent has to carry out in order to achieve its objectives [11], so an intention is an ordered set of actions; each change from state to state is made after carrying out an action (the agent remembers the action carried out in the past, when it was in a specified state, and the subsequent result). A desire will be any of the final states reached in the past (if the agent has to deal with a situation, which is similar to a past one, it will try to achieve a similar result to that previously obtained).

8.3 Air–Sea Interaction Problem

One of the factors of greatest concern in climactic behaviour is the quantity of carbon dioxide present in the atmosphere. Carbon dioxide is one of the greenhouse gases that helps to make the earth's temperature habitable, so long it is maintains certain levels [43]. Traditionally, it has been considered that the main system regulating carbon dioxide in the atmosphere is the photosynthesis and respiration of plants. However, thanks to teledetection techniques, it has been shown that the ocean plays a highly important role in the regulation of carbon quantities, the full significance of which still needs to be determined [44]. Current technology allows us to obtain data and make calculations that were unthinkable some time ago. This data gives us an insight into the original source and the decrease in carbon dioxide as well as its causes [30], which allows us to make predictions on the behaviour of carbon dioxide in the future.

The need to quantify the carbon dioxide valence, and the exchange rate between the oceanic water surface and the atmosphere, has motivated us to develop the distributed system, presented here, that incorporates CBR agents capable of estimating such values using accumulated knowledge and updated information. The CBR agents receive data from satellites, oceanographic databases, oceanic, and commercial Vessels. The CBR system incorporated within the BDI agents allows the agents to optimize tasks such as the interpretation of images using various strategies [39]. The information received is composed of satellite images of the ocean surface, wind direction and strength, and other parameters such as water temperature, salinity, and fluorescence as can be

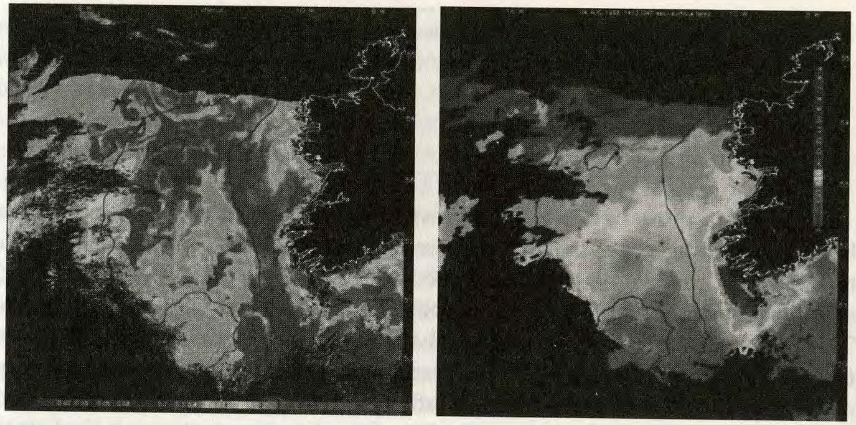


Fig. 8.1. Satellite colour pictures

seen in Fig. 8.1. An improvement over the monitoring and forecasting methods presented in [5,6,15] has been incorporated to the modelling CBR agents presented in this chapter.

The parameters obtained from the satellite images and which have most influence within our models are: temperature of the water and air, salinity of the water, wind strength, wind direction, and biological parameters such as chlorophyll. These parameters allow us to calculate the variables that define our models, such as the velocity of gas transfer, solubility, or the differentiation between partial pressures on the atmosphere and sea surface (a case structure is shown in Table 8.1). The majority of CO2 is dissolved in the sea water because of phytoplankton or accumulates at the bottom of the ocean in the form of organic material. The phytoplankton present in deep areas of the ocean is taken to the surface by surges or surface appearances that are no more than large upwards movements of cold water that bring nutrients to the sea surface. The principal cause of there surges are the winds. The way to detect them through satellites is to study the images captured with sensors that are sensitive to longitudes of thermal infrared waves (capable of detecting the sea surface temperature (SST)) and to identify the cold waters. Another possible way to detect them is to monitor the activity of the chlorophyll through sensors within the spectrum range found between blue and green which are associated with the presence of phytoplankton. In order to obtain the satellite images that contain information about these parameters it is necessary to use different sensors. The Earth Observation satellites that have been used to obtain images in the Northern Atlantic are NOAA, Orbview-2, and above all, the ENVISAT satellite of the European Space Agency. Below we shall briefly describe the sensor used in each one of these and the software for the digital processing of the images.

The thermal sensors allow us to measure the surface temperature of the sea. The NOAA satellites are equipped with the Advanced Very High-Resolution Radiometer (AVHRR) sensor that is capable of detecting electromagnetic energy reflected by objects present on the earth within five spectrum ranges (three bands in the visible and two in the thermal range). It has a receiving cycle of 12 hours with which it is possible to obtain up to six images per day at a resolution of $1\,km^2$. In order to determine the SST, the NOAA uses a multichannel algorithm for the water surface [29]. The ENVISAT satellite has an Avanced Along-Track Scaning Radiometer (AATSR) sensor with which it is capable of exploring the ocean surface at various infrared and visible frequencies in order to measure the exact temperature. Specifically, the temperature of the sea surface can be calculated with an accuracy of 0.3°C. [45]

There are also sensors that allow us to measure the concentration of chlorophyll. The Earth Observation satellite Orbview-2 uses a Sea-Viewing Wide Field-of-view Sensor (SeaWIFS) [29, 49], which is capable of giving images with information on eight bands or ranges of the electromagnetic spectrum. Of these eight bands, four around the blue–green are used for the detection of chlorophyll. In order to calculate these quantities, the Ocean Chlorophyll 4-band OCTS is used, included in the SeaWIFS Data Analysis System (SeaDAS) software developed by NASA [38]. The ENVISAT satellite has a Medium Resolution Image Spectrometer (MERIS) with which it is possible to take images of the planet surface and the clouds, capturing the light of the visible areas, and the infrared of the electromagnetic spectrum. In this way it is capable of knowing the exact colour of the ocean surface and coastal areas, from which it is possible to reflect the biological activity, to monitor cloud cover and to detect the vapour of the invisible water into the atmosphere. [3, 4, 45]

The processing of the images obtained may vary depending on the sensor that has taken them [19, 20]. The processing of the images is carried out at the CAXIS centre at the Plymouth Marine Laboratory (PML). The processing of the thermal images is carried out initially by taking a reading of the images in their original format as they were received. Then a calibration and a radiometric correction is made in order to reduce the atmospheric effects and a reference is made to a known cartography base. The next step is to mask the clouds and the land in order to eliminate distortions. Lastly, the SST is calculated applying a suitable algorithm. In order to process images of the chlorophyll concentration a reading is made of the images and decoded when necessary. Meteorological and ozone files are requested by the software. The clouds and land are masked and the chlorophyll image is calculated. Lastly, reference is made to a known cartographic base and compositions and midpoint images are made, which can take some days.

The MAS presented is aimed at modelling the flux of carbon dioxide exchanged between the atmosphere and the ocean surface. The oceans contain approximately 50 times more carbon dioxide in dissolved forms than the atmosphere, while the land biosphere including the biota and soil carbon contains about three times as much carbon (in carbon dioxide form) as the atmosphere [44]. The carbon dioxide concentration in the atmosphere is governed primarily by the exchange of carbon dioxide with these two dynamic reservoirs. Since the beginning of the industrial era, about 2,000 billion tons of carbon have been released into the atmosphere as carbon dioxide from various industrial sources including fossil fuel combustion and cement production. This amount, which is about 35% of the total amount of carbon in the preindustrial level, corresponds to approximately 590 billion tons as carbon. At present, atmospheric carbon dioxide content is increasing at an annual rate of about 3 billion tons which corresponds to one-half of the annual emission rate of approximately 6 billion tons from fossil fuel combustion. Whether the missing carbon dioxide is mainly absorbed by the oceans or by the land and their ecosystems has been debated extensively over the past decade.

It is important, therefore, to fully understand the nature of the physical, chemical, and biological processes which govern the oceanic sink/source conditions for atmospheric carbon dioxide [30, 44]. Satellite-borne instruments provide high-precision, high-resolution data on atmosphere, ocean boundary layer properties and ocean biogeochemical variables, daily, globally, and in the long term. All these new sources of information have changed our approach to oceanography and the data generated needs to be fully exploited. Wind stress, wave breaking, and the damping of turbulence and ripples by surface slicks, all affect the air–sea exchange of carbon dioxide. These processes are closely linked to the "roughness" of the sea surface, which can be measured by satellite radars and microwave radiometers. Sea surface roughness consists of a hierarchy of smaller waves upon larger waves. Different sensors give subtly different measurements of this roughness.

Our final aim is to model both the open ocean and shelf seas and it is believed that by assimilating Earth Observation (EO) data into artificial intelligence models these problems may be solved. Earth Observation data (both for assimilation and for validation) are vital for the successful development of reliable models that can describe the complex physical and biogeochemical interactions involved in marine carbon cycling. Satellite information is vital for the construction of oceanographic models, and in this case, to produce estimates of air–sea fluxes of carbon dioxide with much higher spatial and temporal resolution, using artificial intelligence models than can be achieved realistically by direct in situ sampling of upper ocean carbon dioxide. To handle all the potentially useful data to create daily models in a reasonable time and with a reasonable cost, it is necessary to use automated distributed systems capable of incorporating new knowledge. Our proposal is presented in Sect. 8.4.

8.4 Air–Sea Interaction Multiagent System

The option chosen to define an appropriate analysis and design methodology for the problem to be resolved is one that combines Gaia [48] and Agent-UML (AUML) [8, 34, 35], in an attempt to take advantage of both. Through Gaia it is possible to make an analysis of the problem using organizational criteria and a later design. After applying Gaia, the result consists of a design at the elevated abstraction level. At this point the Gaia design is transformed so that Agent-UML techniques can be applied. Figure 8.2 illustrates the paths followed in order to obtain the different models used. It shows how Gaia is used initially in order to obtain an analysis and high-level design and then Agent-UML is used in order to obtain a detailed, low-level design.

8.4.1 Gaia Analysis and Design

Gaia is a methodology for analysis and design in agent-based systems. It is very general and therefore applicable to a very wide range of MAS. It also allows the user to have a wide knowledge of the MAS both at an organizational (social) level and at a detailed level for each agent [48]. Through the Gaia analysis, two models are obtained: the role model and the interaction model. We analyse a problem in terms of organization, first by analyzing the different roles that our system could play. Studying the requirements of the problem we have come to the conclusion that we need six roles: a STORING role, for obtaining data that should be permanently available and stored in a database; a PROCESSING role, for transforming the images from the satellite into cases; a DATACAPTURING role for obtaining the data from the Vessel; a CONSTRUCTAPARTIALCO_2MODEL role, for generating a model; an OBTAINCO_2EXCHANGE role for calculating the rate of CO2 exchange

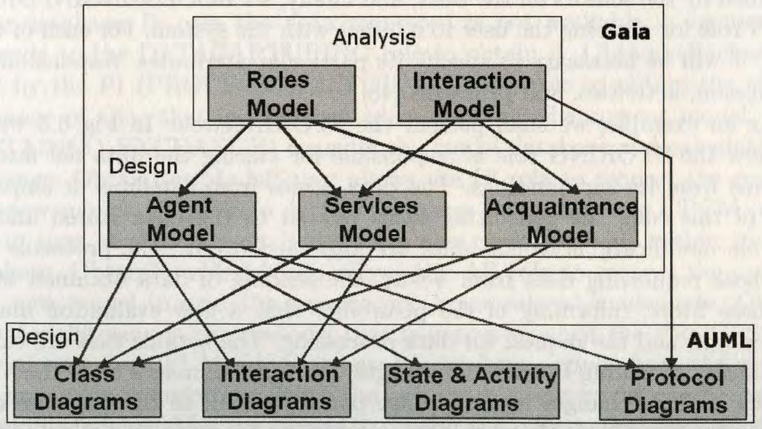


Fig. 8.2. Methodology followed

Role Schema: STORING (S)

Description:
Stores information about the environment i.e. raw data from satellites.
Besides allows consults on stored data.

Protocols and Activities:
StoringSatelliteData, StoringVesselData, RequestInsituData, ConsultCaData, SendCaData, RequestProcessData, ChangeParamsStore, RequestEvaluation, RequestCasesData

---

Permisions:
Reads: On-line Envisal Data, On-line Vessel Data

Changes: CaStore BD

Generates

Responsabilities:
Liveness:
STORESAT: (StoringSatelliteData)^w.RequestProcessData
STOREVESSELDATA: [RequestInsituData].StoringVesselData.[RequestEvaluation]
CONSULTCA: ConsultCaData.[RequestCasesData].SendCaData
MODIFYCA: ChangeParamsStore
Safety: Succesful connection with satellite established.
Succesful connection with Vessel established.
Succesful connection with CaStore BD established.

Fig. 8.3. Gaia role model for the STORING role

using the data from the model; an AUTOEVALUATION role for assessing the model by contrasting the results offered by the model with the real data obtained by the sensors on the boat; and finally, a PROCESSINGINFORMATION role for allowing the user to interact with the system. For each of these roles, it will be necessary to specify its particular attributes: responsibilities, permission, activities, and protocols [48].

As an example, we shall present the STORING role: In Fig. 8.3 we can see how the STORING role is responsible for storing the data fed into the systems from its surroundings. The data comes from satellites or ships. As part of this role, the consulting tasks related to the data stored and the possible modifications to this data are also carried out. The protocols used are those requesting data from Vessel, the sending of data obtained to the database Store, informing of the possibility that a new evaluation may be carried out, and the request for data processing. The actions that are carried out consist of storing the satellite images, storing the images from the Vessel sensors, making changes in the storage parameters, or in the database data, when necessary. The role must have permission to access the data and the Vessel data via satellite. In addition, it must have permission for reading and

writing over the Store database where the data received is stored. Its liveness responsibilities are as follows: STORESAT which continually stores the data received via satellite. When new data is received, it is stored in "raw" format and then a request is made to the PROCESSING role to carry out a data processing action. STOREVESSELDATA is responsible for storing the data that is received from the Vessels. In order for the data to arrive, there exist two possibilities, the data is either requested from the boat, or the boat sends them under its own initiative. In the first case, the sequence is to carry out a data request and store the data. In the second case, the sequence is to store the data, and the appropriate role is informed of the possibility to carry out an autoassessment of the current model with the new data. CONSULTCA carries out consultations concerning the Store database. MODIFYCA makes it possible to carry out modifications both on the parameters of the storage of the database Store, and on the data stored there. Lastly, the safety responsibilities that the STORING role has are those which can establish a valid connection either with the satellite and the Vessel, or with the database.

Once the role model has been obtained, the Gaia analysis is completed with the interaction model. The interaction model shows us the dependences and relationships between the roles. For each interaction between two roles there is a protocol. For our MAS, we have decided to use the following interaction protocols: ObtainVesselData is formed by protocols between the STORING role and the DATACAPTURING role, whereby the first protocol requests the data in situ from the Vessel (for a specific date) and the second protocol ensures that it is given. ObtainConstructData is an interaction through which the CPCM (CONSTRUCTAPARTIALCO_2MODEL) role wishes to construct a new model and in order to do so, requests new cases from the PROCESSING role. The PROCESSING role responds with the requested information. ObtainInsituData allows the AE (AUTOEVALUATION) role to obtain current data in situ aboard the Vessel. To do this, it is necessary to make a request to the STORING role to carry out a consultation of the Store database. In case the data requested is not available, a request will be made to the DATACAPTURING role to obtain it. ObtainStExchange is used by the PI (PROCESSINGINFORMATION) role to obtain the rate of exchange of CO_2 that is produced when applying the current model. OCE (OBTAINCO_2EXCHANGE) consults the model database and calculates the exchange. ObtainNewModelSuper allows the PI role to request the creation of a new model. In order to do this, it makes a request from CPCM, which will in turn have to consult if there are new cases available within the case database. ObtainNewModelAuto allows the AE role to request the creation of a new model in case the current one is considered inadequate. ObtainNewModelStoring is the protocol that is executed when the PROCESSING role informs the CPCM role that new images have arrived from the satellite and have been transformed. With the new data, a new model can be created. ObtainStModel enables the PI role to consult the information associated with a certain model. ObtainStore allows the PI role to consult the data stored in the Store database. In order to do this, it makes a request to the STORING role. ObtainVessel allows the PI role to consult the data stored in the EPROM memory of the Vessel. In order to do this it makes a request to DATACAPTURING role. ObtainEvaluationSuper is the protocol with which PI requests an evaluation of a model. To do this, it makes a request to the AE role. The AE role needs to know the current data in situ in order to make the evaluation. ObtainEvalautionDC enables the DATACAPTURING role to inform the AE role that the Vessel has carried out a new data collection. This implies that AE is able to carry out an evaluation of the current model. If the evaluation is not satisfactory, a request is made to generate a new model. The DATACAPTURING role does not have any direct communication with the AE role so it must make the request to the STORING role, which acts as an intermediary. Activate/Deactivate Sensors allow the PI role to activate or deactivate the Vessel sensors. In order to do this, it is necessary for it to communicate with the DATACAPTURING role. Delete EPROM allows the PI role to delete all the data from the Vessel's EPROM memory. ChangeStore enables the PI role to modify the storage parameters of the Store data. It communicates with the STORING role that carries out the modifications. ChangeCase allows the PI role to modify the storage parameters of the case memory store. To do this it needs to communicate with the PROCESSING role.

Figure 8.4 illustrates the interaction ObtainVesselData. It shows that the interaction uses two different protocols. In each protocol, a textual description indicates the type of interaction (RequestInsituData and SendInsituData), the role that initiates the interaction (STORING in the first one, and DATACAPTURING in the second one), and the role to which it is directed (DATACAPTURING in the first one and STORING in the second one), and a description of the process that is carried out during the interaction. Moreover, the entry information given by the role that initiates the interaction (InsituDataFromAGivenDate in the second protocol) and the exit information given by the role to which the interaction is directed (VesselInsituData in the second protocol) is also shown.

Once the analysis has been finalized, the Gaia design is carried out. Traditional techniques of software engineering are not followed in terms of detailing the analysis to the extent that a direct implementation can be made. Instead, the level of abstraction is reduced so that traditional techniques can be applied. In the design process three models are considered: agent model, services model, and acquaintance model [48]. The agent model shows the types of agents that are going to appear in the system, as well as the number of instances for each agent type that can be executed within the execution time.

Using the role models as a base, we have decided to use five types of agents: Store, Vessel, Modelling, User, and SuperUser. As Fig. 8.5 illustrates, each agent is responsible for carrying out some particular roles. For example, Store agent is responsible for carrying out STORING and PROCESSING

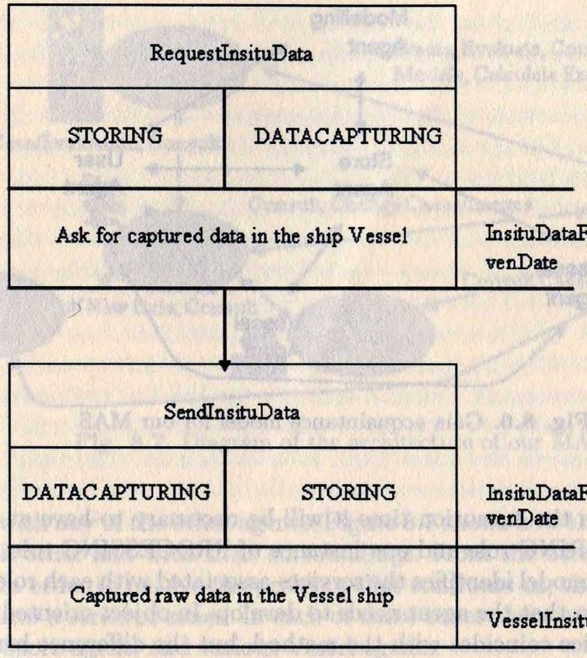


Fig. 8.4. Protocols for the ObtainVesselData interaction

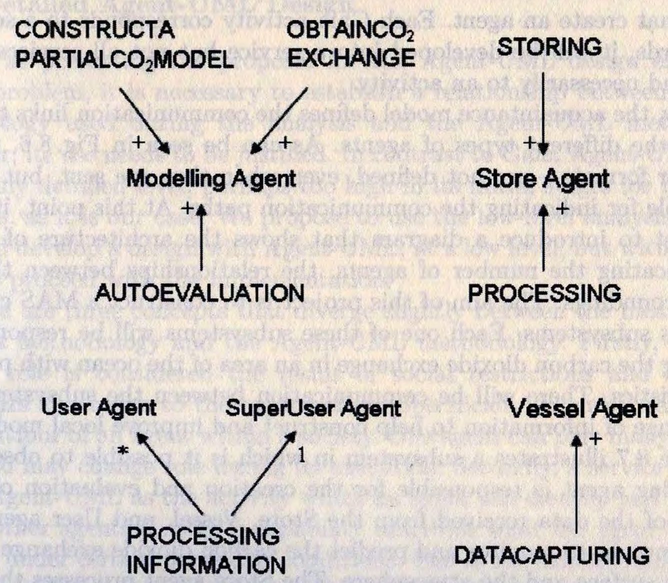


Fig. 8.5. Gaia agents model for our MAS

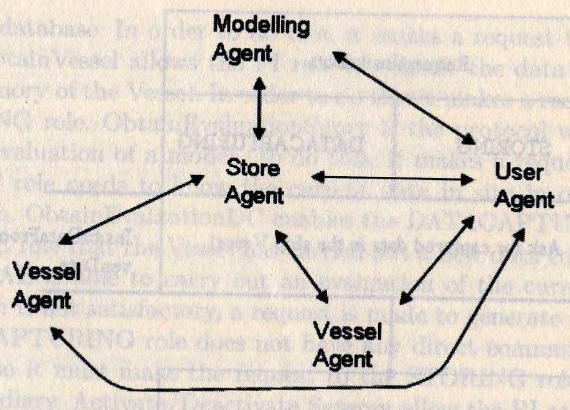


Fig. 8.6. Gaia acquaintance model for our MAS

roles, and within the execution time it will be necessary to have at least one instance of STORING role and one instance of PROCESSING role.

The services model identifies the services associated with each role, being a service a function that the agent needs to develop. In object oriented methodologies, the service coincides with the method, but the difference here is that they would not be available for other agents in the way methods were for other objects. A service will serve as a block of simple, individual, and coherent activities that create an agent. Each Gaia activity corresponds to a service, in other words, it will be developed into a service but not all services need to correspond necessarily to an activity.

Lastly, the acquaintance model defines the communication links that exist between the different types of agents. As can be seen in Fig. 8.6, the messages – or formats – are not defined, even when they are sent, but are only responsible for indicating the communication paths. At this point, it may be of interest to introduce a diagram that shows the architecture of our system, indicating the number of agents, the relationships between them and their surroundings. The aim of this project is to construct a MAS composed of various subsystems. Each one of these subsystems will be responsible for modelling the carbon dioxide exchange in an area of the ocean with particular characteristics. There will be communication between the subsystems, with an exchange of information to help construct and improve local models.

Figure 8.7 illustrates a subsystem in which is it possible to observe how a Modelling agent is responsible for the creation and evaluation of models in terms of the data received from the Store, Vessel, and User agents. This model allows us to monitor and predict the carbon dioxide exchange between the ocean surface and the atmosphere. The Store agent processes the images from the satellite and transforms them for use by the system. Each Vessel agent is installed in a ship and collects information in situ that allows us to evaluate the models created by the Modelling agent. The User agent can

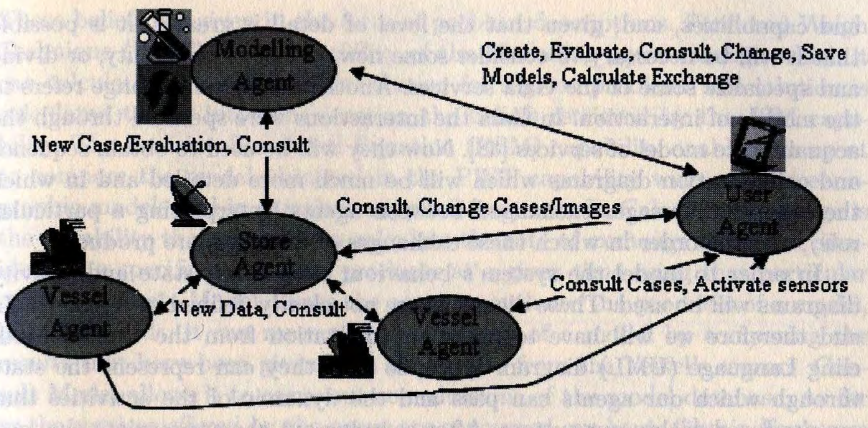


Fig. 8.7. Diagram of the architecture of our MAS

interact with any of the other agents. Figure 8.7 shows how the agents interact with each other and with their surroundings. From the oceanographic point of view, in order to resolve the problem that confronts us, the ocean has been divided into a series of zones. In each of these zones there will be a Modelling agent, a Store agent, and various Vessel agents.

8.4.2 Detailed Agent-UML Design

As well as presenting the proposal for the Agent-UML design established for the problem, it is necessary to establish a relationship between the Gaia methodology used during the analysis and the Agent-UML methodology. Moreover, its use needs to be justified. In contrast to Gaia, Agent-UML works at a highly detailed level, perhaps too high in its initial stages for large scale problems, as it is our case. We propose to use the low-level analysis made by Gaia and develop a design with Agent-UML, at a low level, but with sufficient detail to proceed with the implementation.

There are three concepts that diverge slightly between the meaning from the Gaia methodology and the Agent-UML methodology. Firstly, in Agent-UML a role is considered the result of social restrictions and individual behaviours and refers to the organization. Specifically, it makes reference to the behaviour of an agent within a society. One agent can play many roles in a MAS and may change role during its execution. Secondly, a service is defined within Agent-UML as the activity which an agent can develop and distribute among other agents. Lastly, a capability describes what the agent is capable of doing under certain particular conditions. Due to the existing differences in the definition that Gaia and Agent-UML provide of roles, services, and capacities, it is necessary to adapt the Gaia design to the Agent-UML standard. As far as the roles are concerned, we have divided those of Gaia into more specific Agent-UML roles. The services of Gaia are divided into Agent-UML services and capabilities, and, given that the level of detail is greater, it is possible that it will be necessary to consider some new service or capability, or divide and specialize some of the Gaia services. Another important change refers to the models of interaction. In Gaia the interactions were specified through the acquaintance model of services [48]. Now they will be used to obtain sequence and collaboration diagrams which will be much more detailed and in which there appear messages exchanged between agents (interpreting a particular role), and the order in which these exchanges of messages are produced.

In order to model the system's behaviour even more, state and activity diagrams will be used. These diagrams are not clearly defined in Agent-UML, and therefore we will have to make an adaptation from the Unified Modeling Language (UML) diagram types, so that they can represent the state through which our agents can pass and the dynamic of the activities that are produced within our system. After carrying out the appropriate changes, we begin the design of the Agent-UML by obtaining the class diagrams. The specifications that will be followed are those of the Foundation for Intelligent Physical Agents (FIPA) for the modelling of class diagrams for agents using Agent-UML [8]. We obtain a class diagram for one of the most prominent agents, the Modelling agent, the CBR agent.

Figure 8.8 shows the class diagram for the Modelling agent. The Modelling agent develops six capabilities and offers four services to other agents. The Jacobean Sensitivity Matrix (JSM) capability offers a mechanism to retrieve the beliefs that can be used to solve a given problem in a given situation.

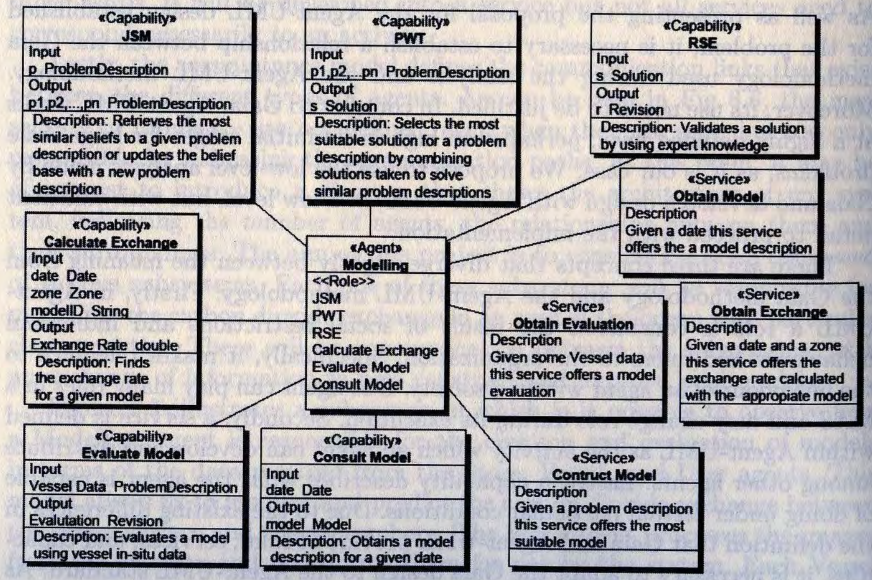


Fig. 8.8. Class diagram for the Modelling agent

These beliefs are given in the form of problem descriptions. Pondered Weigh Technique (PWT) is the capability of the Modelling agent through which it can calculate the most suitable solution for a given problem. The solution is calculated taking into consideration the beliefs retrieved by the JSM capability. The Revision Simulated Equation (RSM) capability enables the agent to compare the model obtained in the PWT capability with other oceanographic models and in situ data supplied from vessels. Calculate Exchange is the capability that allows it to calculate the rate of exchange of carbon dioxide that a specific ocean zone produces at a given time using that particular model. Evaluate Model allows the agent to evaluate the goodness of a model, in other words, it can measure the efficiency of a model by comparing the results that have been given with the results from the Vessel's sensors. Consult Model allows it to carry out consultations of the model database. As far as the services offered by the agent are concerned, we have: Obtain Exchange through which an agent can request a calculation of the rate of exchange of carbon dioxide that is produced in a given ocean zone at a given date, by using the specific model indicated. Obtain Model allows an agent to request the Modelling agent for data from a model that was being used at a given date. Construct Model offers the possibility to attend to construction requests from the models. Lastly, Obtain Evaluation offers any agent the possibility to request an evaluation for a particular model in a particular ocean zone at a given date based on the real data obtained from the Vessel sensors.

The Agent-UML design is completed by offering interaction diagrams which show the interaction between the MAS agents as well as the different roles that can be taken up by the different agents and the interactions between these roles. It is habitual to use a collaboration diagram, although a sequence diagram, which would be equivalent to the collaboration diagram, can also be used, [34]. We can differentiate ten different interactions.

Figure 8.9 shows the interactions between the Modelling and Store agents when a new problem descriptor or case is stored. When new satellite data is received, the Store Sat Data role of the Store agent is in charged of storing the data in the right format with the help of the Transform Im-Cases role. The image is digitally processed in order to obtain the corresponding problem description data. Finally, the Store agent moves into the Store Cases role to store the new problem description data. The Store agent establishes a communication process with the Modelling agent and transfers the new problem description information. The Modelling agent processes the new problem description, and may, creates a new model. The Modelling agent executes the Jacobean Sensitivity Matrix role, in which the agent consults the beliefs base in order to obtain the most similar beliefs (problem descriptions) to the initial problem sent by the Store agent. Once the Modelling agent has retrieved the beliefs, it changes to execute the Pondered Weigh Technique role. Now, the agent calculates the most suitable solution for the initial problem case provided by the Store agent. The most suitable solution is calculated using the cases retrieved by the Jacobean Sensitivity Matrix role as it is shown in

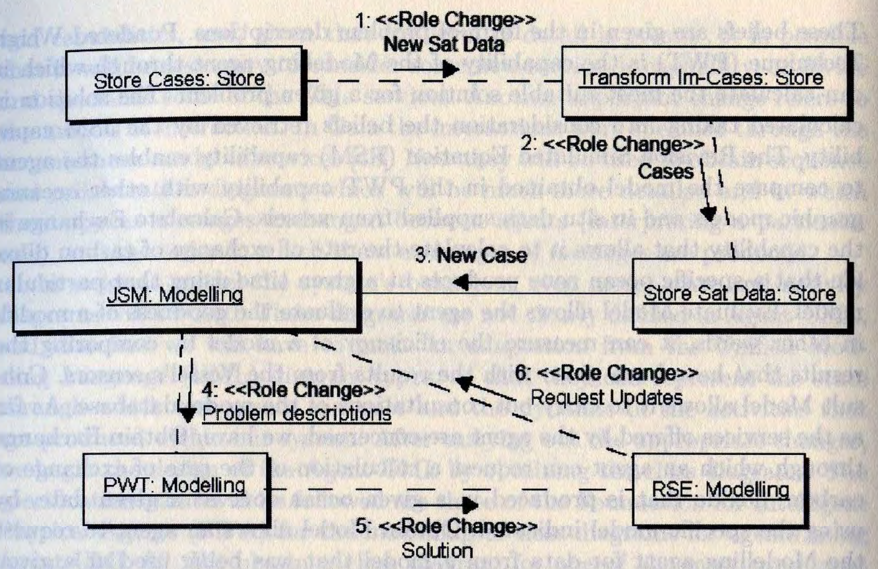


Fig. 8.9. Collaboration diagram corresponding to the interactions that occurs because of the arrival of a new satellite image to the Store agent.

Sect. 8.4.3. A model is created using the most suitable solution, and the RSE role is executed, which is in charge of the revision of the model. Finally, the RSE role transmits the result of the revision by changing back to the role Jacobean Sensitivity Matrix. Now the Modelling agent (with the Jacobean Sensitivity Matrix role) retains the problem description and the knowledge obtained after all this process. A new model may have been created or modified.

To finish the Agent-UML design, state, and activity diagrams are created to model the behaviour of the agents. We use UML state diagrams [37], which we have designed for the Store and Modelling agents. Figure 8.10 shows the state diagram for the Store agent. We believe that the Store agent can be found in three possible states: A state in which the agent is awaiting requests; a state in which the agent is modifying stored data; and thirdly, a state in which the agent carries out operations to store data. Some of these operations include obtaining particular parameters that characterize an image. An additional state allows it to be ready to receive new requests when it has no tasks to be carried out. Finally, to finish with the Agent-UML design, in terms of the activity diagrams, we focus on the activities that will be developed within the CBR cycle.

Once the design is complete, we go on to the implementation, using the Jadex tool, a tool that incorporates the BDI model within Jade agents and tool. With Jadex, the Modelling agents are built while the rest of the agents will be Jade. The communication mechanisms are the same as in Jade (Agent Communication

Language (ACL) is used) [9, 41]. The use of Jadex means that it is necessary to use Object Query Language (OQL) consulting language.

8.4.3 The CBR–BDI Modelling Agent

Once the architecture proposed has been studied, it would seem a good idea to deepen the Modelling agents – in the form of a deliberative agent that uses a CBR mechanism. This agent will have two principal functions. The first one is to generate models which are capable of predicting the atmospheric/oceanic interaction in a particular area of the ocean in advance. The second one is to permit the use of such models. In Fig. 8.10, we can see that a Modelling agent possesses two principal states: one to generate the forecasting models and the other to permit the use of the models. Moreover, the reasoning cycle is one of the activities carried out by the Modelling agent. We can see how the reasoning cycle of a CBR system is included among the activities, composed of stages of retrieval, reuse, revise, and retain. Also, an additional stage that introduces expert's knowledge is used. This reasoning cycle must correspond to the sequential execution of some of the agent roles.

The Modelling agent presents a deliberative architecture, based on the BDI model [12]. In this model, the internal structure and capabilities of the agents are based on mental aptitudes, using beliefs, desires, and intentions. This method facilitates the incorporation of CBR systems [1] as a deliberative mechanism within BDI agents, facilitating learning, and adaptation

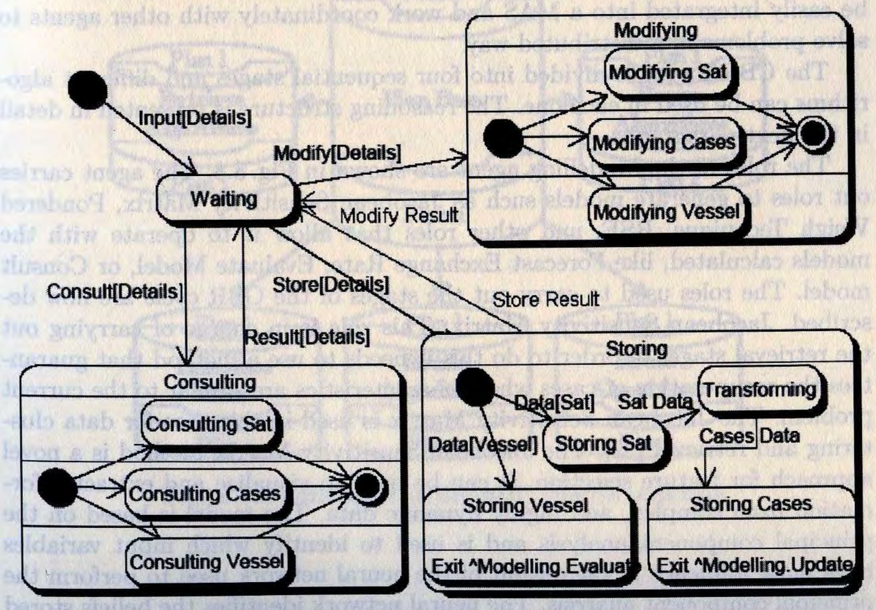


Fig. 8.10. State diagram for the Store agent

and providing a greater degree of autonomy than pure BDI architecture. To introduce a CBR motor into a BDI agent it is necessary to represent the cases used in a CBR system by means of beliefs, desires, and intentions, and implement a CBR cycle. A case is a past experience composed of three elements: an initial state or problem description that is represented as a belief; a final state that is represented as a set of goals; and the sequence of actions that makes it possible to evolve from an initial state to a final state. This sequence of actions is represented as intentions or plans. CBR consists of four sequential stages: retrieve stage to recover the most similar past experiences to the current one; reuse stage to combine the retrieved solutions in order to obtain a new optimal solution; revise stage to evaluate the obtained solution; and retain stage to learn from the new experience.

Figure 8.11 shows the internal structure of a CBR agent. Problem description (initial state) and solution (situation when final state is achieved) are represented as beliefs, the final state as a goal (or set of goals), and the sequences of actions as plans. The CBR cycle is implemented through goals and plans. When the goal corresponding to one of the stages is triggered, different plans (algorithms) can be executed concurrently to achieve the goal. Each plan can trigger new subgoals and, consequently, cause the execution of new plans.

Deliberative CBR agents, like Modelling agent, are able to incorporate other reasoning mechanisms that can coexist together with the CBR. Modelling is an autonomous agent that can survive in dynamic environment. However, is possible to incorporate communication mechanisms that allow it to be easily integrated into a MAS and work coordinately with other agents to solve problems in a distributed way.

The CBR motor is divided into four sequential stages and different algorithms can be used in each one. The reasoning structure is presented in detail in the next paragraphs.

The roles of the Modelling agent are shown in Fig. 8.8. The agent carries out roles to generate models such as Jacobean Sensitivity Matrix, Pondered Weigh Technique, RSE, and other roles that allow it to operate with the models calculated, like Forecast Exchange Rate, Evaluate Model, or Consult model. The roles used to carry out the stages of the CBR cycle are now described. Jacobean Sensitivity Matrix: This role is in charge of carrying out the retrieval stage. In order to do this it needs to use a method that guarantees the recuperation of cases whose characteristics are similar to the current problem. The Jacobean Sensitivity Matrix is used in this case for data clustering and retrieval [32]. The Jacobean Sensitivity Matrix method is a novel approach for feature selection. It can be used to visualize and extract information from complex, and highly dynamic data. The model is based on the principal component analysis and is used to identify which input variables have more influence in the output of the neural network used to perform the principal component analysis. The neural network identifies the beliefs stored

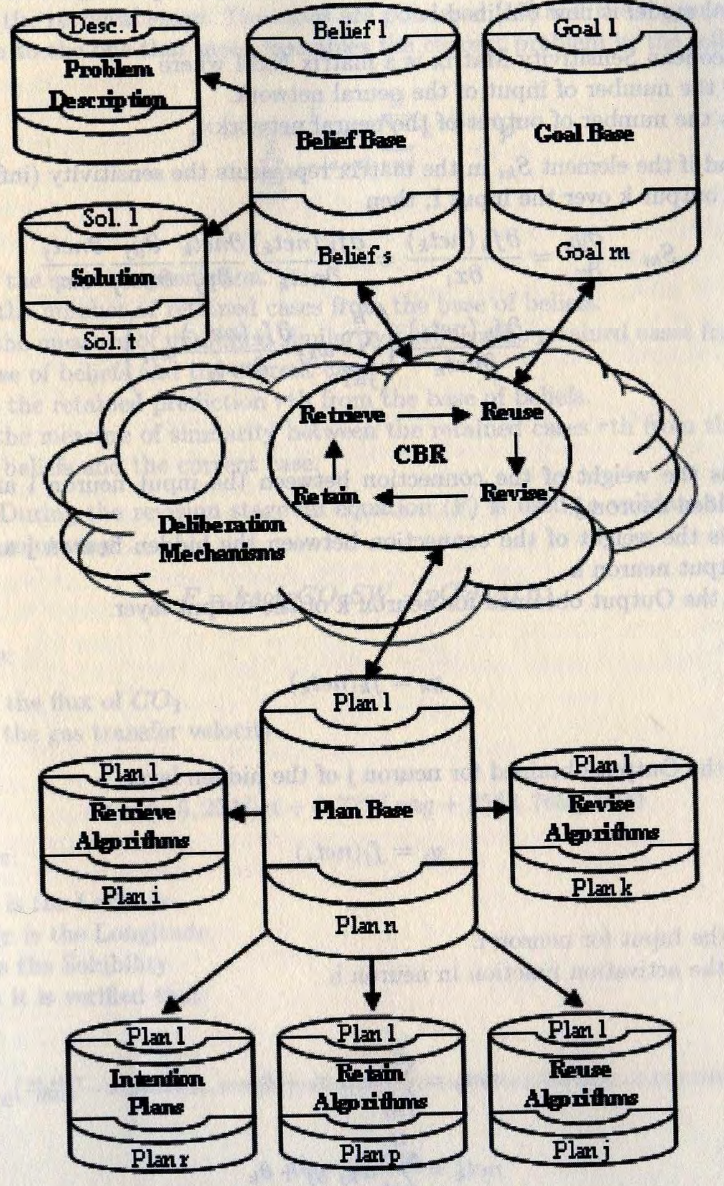


Fig. 8.11. CBR-agent (CBR–BDI) internal structure

by the agent that can be more useful to solve a given problem. The mathematical model is now outlined.

If Jacobean Sensitivity Matrix is a matrix NxM where
N: is the number of input of the neural network.
M: is the number of output of the neural network.

And if the element S_{ki} in the matrix represents the sensitivity (influence) of the output k over the input I, then

$$S_{ki} = \frac{\partial y_k}{\partial x_i} = \frac{\partial f_k(net_k)}{\partial x_i} = \frac{\partial f_k(net_k)}{\partial net_k}\frac{\partial net_k}{\partial y_j}\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial x_i}$$

$$= \frac{\partial f_k(net_k)}{\partial net_k}\left(\sum_{j=1}^{H} w_{kj}\frac{\partial f_j(net_j)}{\partial net_j} w_{ji}\right) \quad (8.1)$$

where

w_{ji}: is the weight of the connection between the input neuron i and the hidden neuron j.
w_{kj}: is the weight of the connection between the hidden neuron j and the output neuron k.
y_k: is the Output obtained for neuron k of the output layer.

Then

$$y_k = f_k(net_k) \quad (8.2)$$

where

y_j: is the Output obtained for neuron j of the hidden layer.

Then

$$y_j = f_j(net_j) \quad (8.3)$$

where

x_i: is the Input for neuron i.
f_h: is the activation function in neuron h.

then

$$net_j = \sum_{i=1}^{N} w_{ji}\, x_i + \theta_j \quad (8.4)$$

$$net_k = \sum_{j=1}^{H} w_{kj}\, y_j + \theta_k \quad (8.5)$$

where

H: is the number of neurons in the hidden layer.
θ_j: is the value of threshold of neuron j of the hidden layer.
θ_k: is the value of threshold of neuron k of the output layer.

Pondered Weigh Technique: The reuse is carried out using the cases selected during the retrieval stage. The cases are pondered [17] and the bigger weight is given to the one that more resembles the current problem in the following way:

$$p^* = \frac{1}{\sum_{r=1}^{Z} e^{-|a-r|}} \sum_{r=1}^{Z} e^{-|a-r|} p^r \quad (8.6)$$

where:

p^*: is the solution prediction.
Z: is the number of retained cases from the base of beliefs.
a: is the measure of minimum similarity between the retained cases from the base of beliefs and the current case.
p^r: is the retained prediction rth from the base of beliefs.
r: is the measure of similarity between the retained cases rth from the base of beliefs and the current case.

RSE: During the revision stage an equation (F) is used to validate the proposed solution p^*.

$$F = kso(pCO_2SW - pCO_2AIR) \quad (8.7)$$

Where:

F: is the flux of CO_2.
k: is the gas transfer velocity.

Then

$$k = (-5,204Lat + 0,729Long + 2562,765)/3600 \quad (8.8)$$

Where:

Lat: is the Latitude.
Long: is the Longitude.
so: is the Solubility.
then it is verified that:

$$so = e^{\left(\frac{93,4517}{100tk} - 60,2409 + 23,3585\log(100tk) + s(0,023517 - 0,023656\bullet 100tk + 0,0047036\bullet 1002tk)\right)} \quad (8.9)$$

$$\text{tk} = 273,15 + \text{t} \quad (8.10)$$

Where:

t: is the Temperature.
s: is the Salinity.

$$pCO_2 = A + BLong + CLat + DSST + EYear \quad (8.11)$$

Table 8.1. Case attributes

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 ($m\,s^{-1}$)
WD	Wind direction (unitless)
Fluo_calibrated	Fluorescence calibrated with chlorophyll
SW pCO_2	Surface partial pressure of CO_2 (microatmospheres)
Air pCO_2	Air partial pressure of CO_2 (microatmospheres)
Flux of CO_2	CO_2 exchange flux ($Moles\,m^{-2}$)

Table 8.2. Months\Coefficients values

Months\Coefficients	A	B	C	D	E
Feb	−2488	−0,42	4,98	−12,23	1,38
May	−7642	−0,9	−1,74	−20,77	4,14
Jun	−4873	−0,85	1,3	−15,64	2,66
Jul	−7013	−0,025	3,66	−7,07	3,64
Aug	−3160	−0,69	0,84	−11,31	1,8
Sep	−1297	0,43	−4,19	−17,06	1,05
Oct	83	−0,81	4,81	−10,92	0,076
Nov	747	0,2	−0,73	−17,3	−0,062
Dec	−4306	0,38	−0,22	−17,13	2,45

Where SST is the temperature of the marine surface or air as it corresponds to pCO_2SW or pCO_2AIR. The coefficients of the equation depend on the month, as shown in Table 8.2.

During the revision, the agent compares the obtained F value with the predicted one, and if the prediction differs in less than 10%, the case is stored on the base of beliefs. As it has been shown the CBR agents use a CBR system, at a low level of implementation, which is the reason for using cases. One case for the CBR consists of a problem (initial situation and a number of goals) and the plans to resolve it. For oceanic/atmospheric interaction, we define the problem in terms of the attributes shown in Table 8.1:

Table 8.1 shows the description of a case: DATE, LAT, LONG, SST, S, WS, WD, Fluo_calibrated, SW pCO_2, and Air pCO_2. Flux of CO_2 is the value to be identified.

As mentioned in Sect. 8.2 there is a correspondence between cases and BDI agents. To use a deliberative BDI model that utilizes a CBR mechanism, it is necessary to transform the case representation by the CBR system into a BDI formalisms. The BDI model deals with:

1. Beliefs, which represent the state of the problem, with certain knowledge about the surroundings and the agent itself. In our problem we shall use as belief the attributes DATE, LAT, LONG, SST, S, WS, WD, Fluo_calibrated, SW pCO_2, and Air pCO_2. A beliefs base will be used in which each belief is a ProblemDescription type and contains all the attributes mentioned in Table 8.1.
2. Desires, that represent those final states to which the agent wishes to arrive or reach. In this case, it deals with three goals:
   - Predict the flux of carbon dioxide exchanged between the sea surface and the atmosphere, using a window of two or three weeks.
   - Calculate the best parameters to use in order to improve the prediction for different window sizes.
   - Calculate the most suitable prediction window in relation to a maximum % error allowed.

   An agent stores all the goals in a similar way to the beliefs.
3. Intention, that represents the sequence of actions that should be followed in order to reach the final state or goal. This new attribute is introduced into the case description. The sequence of actions to be carried out is generally formed by the stages of the reasoning cycle and the different algorithms executed in each one of those stages. In general an agent will have available various predefined plans or intentions that could be called up and modified at the execution time. The selection of plans is made through the CBR agent, Jacobean Sensitivity Matrix, Pondered Weigh Technique, and Revise Simulated Equation mechanisms.

The tools offered by the Jadex platform [41] have been used for the storing and use of beliefs, desires, and intentions or plans. In this way, we have been able to construct a deliberative BDI agent capable of reasoning through the use of a CBR mechanism. The agent manages cases and carries out CBR cycles.

8.4.4 Communication Agents

To complete the proposal for the MAS, we outline the types of communication that the system agents employ. The Jadex tool has been used to carry out the implementation of the system. This tool is an extension of Jade and, among other features, which uses a standard for communications in accordance with the FIPA [23]. In this way, both ontologies and languages used are those proposed by the FIPA. Jade uses the ACL defined by the FIPA. The agents send and receive java objects that represent ACL messages in accordance with a series of protocols. The majority of protocols appear in the libraries offered by Jade [9]. Furthermore, the FIPA – Semantic Language (FIPA-SL) contents language is used.

The messages used have a syntax whereby the agent is instructed to send the message to the receiver of that message. It also indicates the content

```
{Request
        :sender User
        :receiver Modelling
        :content
                (RequestDataModel 10 100 10 100 27-11-
2005)
        :in-reply-to
        :reply-with
        :language FIPA-SL0
        :ontology RequestDataModel LONMin LONMax LATMin
LATMax DATE
}
```

Fig. 8.12. Example of the message used in the multiagent system. The User agent makes a request to the Modelling agent asking about the models generated for the maximum and minimum longitude and latitude coordinates the 27 November 2005.

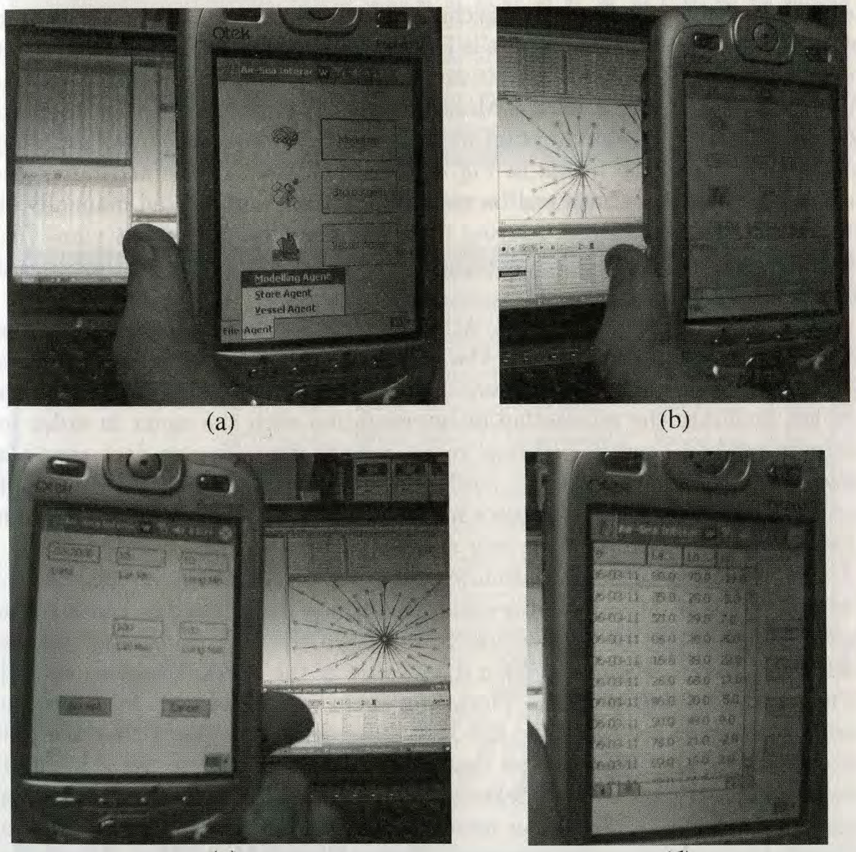
of the message, information to identify the message, name of the language in which the content of the message is written, and the ontologies that define the meaning of the vocabulary used. Figure 8.12 shows an example of a message used by the Modelling agent to request the Storing agent for information on the cases stored in the case base. The language of contents used is FIPA Semantic Language. The ontology defines the vocabulary that is used within the message and the content makes reference to the request for a search to be carried out of the cases and the specific parameters of the search.

The types of messages used in the MAS proposed in this work: request, agree, refuse, cancel, inform, query-if, subscribe, propose, reject-proposal, accept-proposal, failure, and not-understood. As far as the protocols are concerned, the three used are defined by the FIPA: FIPA-request protocol, FIPA-query protocol, and FIPA-ContractNet protocol [23].

8.5 Results and Conclusions

The application of Artificial Intelligence techniques [1] is extremely useful in a field like oceanography and specifically in the study of the carbon dioxide exchange between the ocean surface and the atmosphere. The intelligent environment that has been developed allows oceanographers to maintain a seamless, unobtrusive, and often invisible but also controllable interaction with the available technology. The use of agents and MAS as the fundamental base for creating an intelligent environment is highly suited because of the characteristics of the agents themselves. Their mobility, proactivity, autonomy, social capabilities, reasoning, and capacity for learning all makes the MAS and transparent intermediate layer between the user of the system and the underlying technology. A user can access the system rapidly and efficiently

using their personal agent. This agent is able to sit within a "light" device and can communicate through wireless technology with the other agents of the system. This allows oceanographers to be independent and unbound by location. Figure 8.13 shows the interface of a User agent accessed via a PDA. It is possible to see how the user can access the Modelling, Store, or Vessel agent. The appropriate Store or Vessel can be selected through a simple interface that only presents the necessary information and avoids showing too many elements on the screen. The oceanographers themselves can decide that amount of elements that they wish to see. Figure 8.13b presents the options that can be executed by the Modelling agent: To request the creation of a new model, for which it will be necessary to enter the appropriate parameters; predict the level of exchange in a particular zone of the ocean; make an



(a) (b) (c) (d)

Fig. 8.13. User agent screen shot. **(a)** Menu accessing subsystem agents. **(b)** User options for interacting with the Modelling agent from a light device **(c)** Inquiry about cases from a Store agent. **(d)** Result obtained from the inquiry made in **(c)**

inquiry about the models stored; evaluate a model by entering real data; or saving the corresponding data to the models that are being currently used.

The User agent offers similar menus to allow the user to interact with the Vessel and Store agents. Figure 8.13c shows the parameters that an oceanographer needs to enter when he wishes to make and inquiry the cases stored by the Store agent for the zone of the Atlantic Ocean situated between 10° and 100° latitude and 10° and 100° longitude on the 11 March 2006. After the inquiry is made, the cases are shown to the user in a table as can be seen in Fig. 8.13d. The user can select each one of these cases and modify the parameters.

However, the agents do not represent a mere software that interacts between the user and the technology, but also has the capacity to make decisions and act for themselves in a distributed way, in order to respond and adapt to the changes that are produced within the environment and within its own internal knowledge structure. In this chapter we have described the construction of a MAS whose main component is the Modelling agent, a deliberative agent based on the BDI model [12] that uses CBR [1] as its reasoning mechanism. As can be seen in Fig. 8.14, the Modelling agent handles beliefs, desires, and intention from a conceptual point of view and cases from an implementation point of view. A case (a file in Fig. 8.14) is composed of the attributes described in Table 8.1. Cases can be viewed, modified, and deleted manually or automatically by the agent (during its revision stage). The agent plans (intentions) can be generated using different strategies since the agent integrates different algorithms.

Figure 8.14 shows the North Atlantic exchange rate calculating by the Modelling agent, during the 11 March 2006. The screen shot also presents the algorithms used in the different stages of the CBR cycle. The menus on the left facilitate the interaction or interrogation with the agent in order to evaluate models, predict exchange rates, consult data, change data create a new model or save the current model data. Figure 8.14 presents a view of the Modelling agent. These agents have their own interface and can also be accessed via the User or Super user agents.

The Modelling agent is fully integrated within the MAS. As can be observed in Fig. 8.15, the Modelling agent creates new goals based on the changes in its internal state or in response to messages received from other agents within the system. Figure 8.15 is a screen shot of the Jadex Tracer agent [41] in which the behaviour of the Modelling agent is represented. In it we can observe the goals generated by the Modelling agent, the plans that are put into operation, and the messages that it receives. For example, for a Request message inquiring about a model received from the Gui agent, the Modelling agent executes the update_plan model, from which a consult_model goal is created, that subsequently launches the plan ConsultModelPlan. As can be seen, the agent is capable of handling various goals and plans at the same time.

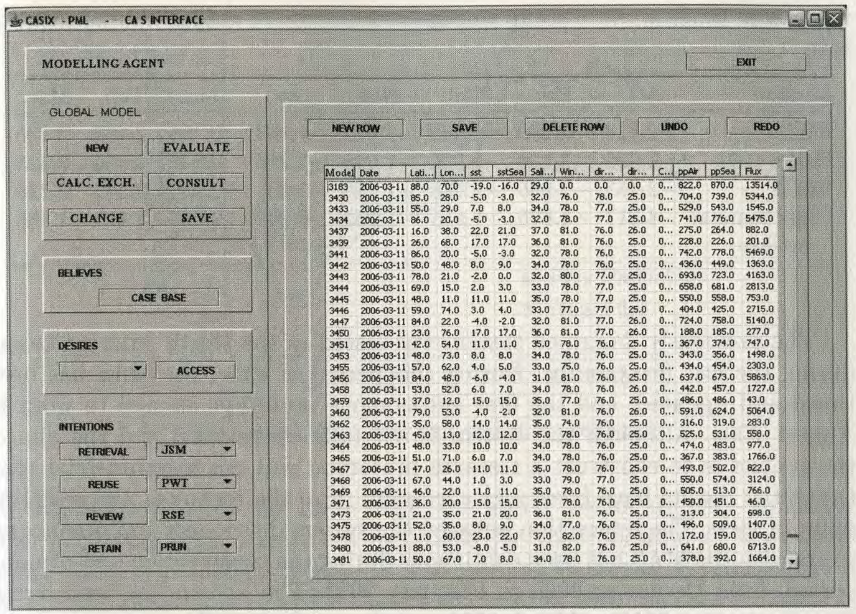


Fig. 8.14. Modelling agent User interface

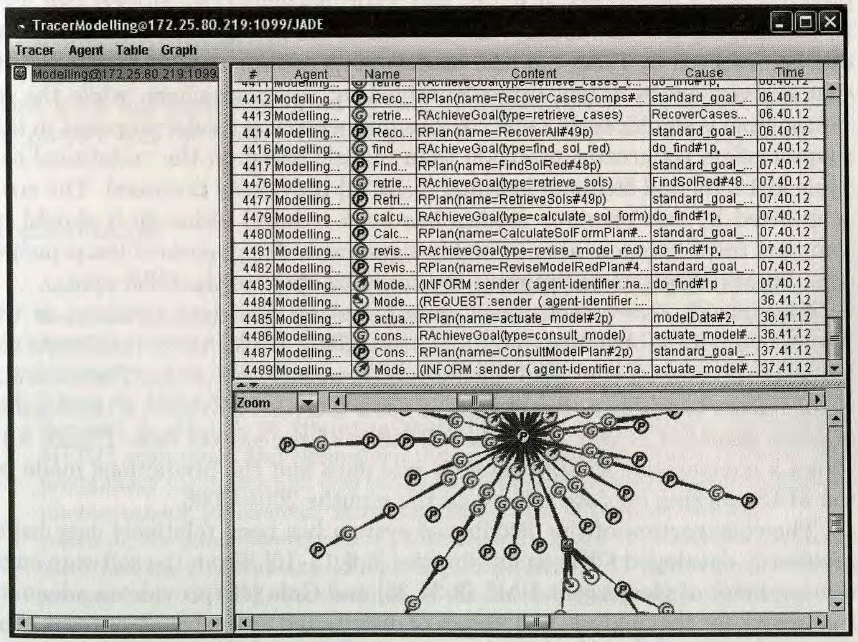


Fig. 8.15. Behaviour of the Modelling agent observed through a Jadex Tracer agent

Table 8.3. Mmol $m^{-2}d^{-1}$ of CO_2 exchanged in the North Atlantic during 2005 and 2006

Method	Aug.	Sep.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.
PWT	−2.119	−2.230	−1.885	−1.622	−1.164	0.495	2.435	2.235
CoHel IBR	−2.325	−2.126	−1.926	−1.625	−1.210	0.845	2.634	2.325
VCBP	−2.453	−2.965	−2.036	−1.155	0.965	−0.235	2.555	2.725
Casix manual models	−4.317	−1.875	−1.655	−1.233	0.205	2.035	3.978	1.955

The previously described system was tested in the North Atlantic Ocean during the last few months. During this period of time, the MAS has been tuned and updated, and the first autonomous prototype started to work In August 2004. Although the system is not fully operational and the aim of the project is to construct a research prototype and not a commercial tool, the initial results have been very successful from the technical and scientific point of view. The interaction between the system developers and oceanographers with the MAS has been continuous during the construction and pruning period, from December 2003 to February 2005. The system has been tested from September 2005 to March 2006 and the results have been very accurate. Table 8.3 presents the results obtained with the MAS proposed in this work, previous MAS developed [5, 6, 15], and with mathematical Models [30] used by oceanographers to identify the amount of carbon dioxide exchanged. As can be observed in Table 8.3, the models proposed for the MAS offer results that are very close to real values obtained by oceanographers, while the response time is significantly reduced. The mathematical model proposed in this chapter offers far greater precision than models based on the variational calculus [6,14,16] and Hebbian learning [5,15,24] previously proposed. The error committed by two previous models has been reduced although it should be said that the differences are not highly significant. An analysis of the principal components allows us to optimize the recovery stage of the CBR cycle.

The models have constructed cases based on real data obtained in the Azores zone of the Atlantic Ocean (± 37N, 25W). Under these conditions the models proposed for the MAS have been increasingly accurate. The accuracy of the results increases as the number of cases increase. However, if the number of cases managed is very high, the efficiency of the system falls. Figure 8.16 shows a comparative sample between real data and the predictions made by the MAS working on data related to the months 2005–2006.

The construction of the distributed system has been relatively easy using previously developed CBR-agent libraries [5,6,13–16]. From the software engineering point of view Agent-UML [8,34,35] and Gaia [48] provide an adequate framework for the analysis and design of distributed agent-based systems. The formalism defined in [25] facilitates the straight mapping between the agent definition and the CBR construction. Although the proposed system requires

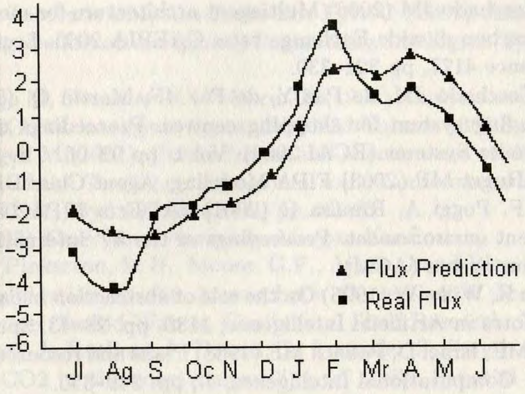


Fig. 8.16. CO_2 real flux and flux prediction

further improvements and more work, the initial results are very promising. The generated open framework facilitates the incorporation of new agents using different modelling techniques and learning strategies so further experiments will allow us to compare these initial results with the ones obtained by other techniques.

Acknowledgements

This work has been supported by the Spanish MCYT TIC2003-07369-C02-02 project and the PML's CASIX project.

References

1. Aamodt A, Plaza E (1994) Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, AICOM. Vol. 7., March, pp 39–59.
2. Aarts E (2004) Ambient intelligence, Third International Conference, AH 2004, Eindhoven, The Netherlands. LNCS Springer Verlag. ISBN: 3-540-22895-0, p. 1.
3. Aiken J., Fishwick J.R., Lavender S.J., Barlow R., Moore G.F., Sessions H., Bernard S., Ras J. & Hardman-Mountford N.J. (Submitted). Validation of MERIS reflectance and chlorophyll during the BENCAL cruise October, 2002: preliminary validation and new products for phytoplankton functional types and photosynthetic parameters. International Journal of Remote Sensing.
4. Aiken J., Hardman-Mountford N., Ufermann S., Woolf D., Challenor P., Robinson I. and the CASIX team, (2005). Exploiting ENVISAT-ERS data for deriving air-sea fluxes of CO2, Proceedings of the Envisat Symposium, Salzburg, ESA publication, in press.
5. Bajo J, Corchado JM (2005) Evaluation and monitoring of the air-sea interaction using a CBR-Agents approach. ICCBR 2005. Lecture Notes in Computer Science 3620, pp 50–62.

6. Bajo J, Corchado JM (2006) Multiagent architecture for monitoring the North-Atlantic carbon dioxide Exchange rate. CAEPIA 2005. Lecture Notes in Computer Science 4177, pp 321–330.
7. Bajo J, Corchado JM, de Paz Y, de Paz JF, Martín Q. (2006) A multiagent recommending system for shopping centres. Proceedings of the workshop on Recommender Systems (ECAI 2006). Vol 1, pp 92–06.
8. Bauer B, Huget MP (2003) FIPA Modeling: Agent Class Diagrams.
9. Bellifime F, Poggi A, Rimasa G (2001) JADE: a FIPA2000 compliant agent development environement. Proceedings of the 5th international conference on autonomous agents (ACM).
10. Bergmann R, Wilke W (1996) On the role of abstraction in case-based reasoning. Lecture Notes in Artificial Intelligence, 1186, pp. 28–43. Springer Verlag.
11. Bratman ME, Israel D, Pollack ME (1988) Plans and resource-bounded practical reasoning. Computational Intelligence, 4., pp. 349–355.
12. Bratman ME (1987) Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A.
13. Corchado JM, Laza R (2003) Constructing Deliberative Agents with Case-based Reasoning Technology, International Journal of Intelligent Systems. Vol 18, No. 12, December. pp. 1227–1241.
14. Corchado JM, Lees B (2001) A Hybrid Case-based Model for Forecasting. Applied Artificial Intelligence. Vol 15, no. 2, pp. 105–127.
15. Corchado JM, Aiken J, Corchado E, Lefevre N, Smyth T (2004) Quantifying the Ocean's CO2 Budget with a CoHeL-IBR System. 7th European Conference on Case-based Reasoning, Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence 3155, Springer Verlag. pp. 533–546.
16. Corchado J M, Pavón J, Corchado E, Castillo LF (2005) Development of CBR-BDI Agents: A Tourist Guide Application. 7th European Conference on Case-based Reasoning 2004. Lecture Notes in Artificial Intelligence 3155, Springer Verlag. pp. 547–559.
17. De Paz Santana Y (2005) Mixture Weibull distribution using artificial neural networks with csnsurated data PHD thesis, Chapter 3.
18. DeLoach S (2001) Anlysis and Design using MaSE and AgentTool. Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS).
19. Dransfeld S., Tatnall A.R., Robinson I.S. and Mobley C.D. (2004). A comparison of Multi-layer Perceptron and multilinear regression algorithms for the inversion of synthetic ocean colour spectra. Int. J. Remote Sens. 25 (21): 4829–4834.
20. Dransfeld S., Tatnall A.R., Robinson I.S. and Mobley C.D. (2005). Prioritizing ocean colour channels by neural network input reflectance perturbation. Int. J. Remote Sens. 26 (5): 1043–1048.
21. EURESCOM (2001) MESSAGE: Methodology for engineering systems of software agents. Technical report P907-TI1, EURESCOM.
22. Feret MP, Glasgow JI (1994) Explanation-Aided Diagnosis for Complex Devices, Proceedings of the 12th National Conference an Artificial Intelligence, (AAAI-94), Seattle, USA, August 94.
23. FIPA Foundation for Intelligent Physical Agents. http://www.fipa.org.
24. Fyfe C, Corchado ES (2002) Maximum Likelihood Hebbian Rules. European Symposium on Artificial Neural Networks. 2002.

25. Glez-Bedia M, Corchado JM, Corchado ES, Fyfe C (2002) Analytical Model for Constructing Deliberative Agents, Engineering Intelligent Systems, Vol 3, pp. 173–185.
26. Iglesias C, Garijo M, Gonzalez JC, Velasco JR (1998) Analysis and Design using MAS-CommonKADS. Inteligent Agents IV LNAI Volume 1365 Springer Verlag.
27. Kinny D, Georgeff M (1991). Commitment and effectiveness of situated agents. In: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI'91), Sydney, Australia, pp. 82–88.
28. Kolodner J (1993) *Case-based reasoning.* Morgan Kaufmann.
29. Lavender, S.J., Pinkerton, M.H., Moore, G.F., Aiken, J and Blondeau-Patissier, D. (2005). Modification to the Atmospheric Correction of SeaWiFS Ocean Colour Images over Turbid Waters. Continental Shelf Research, 25, 539–555.
30. Lefevre N, Aiken J, Rutllant J, Daneri G, Lavender S, Smyth T (2002) Observations of pCO2 in the coastal upwelling off Chile: Sapatial and temporal extrapolation using satellite data. Journal of Geophysical research. Vol. 107, no. 0.
31. Martín FJ, Plaza E, Arcos JL (1999) Knowledge and experience reuse through communications among competent (peer) agents. International Journal of Software Engineering and Knowledge Engineering, Vol. 9, No. 3, pp. 319–341.
32. Montaño Moreno JJ, Palmer Pol A (2002) Artificial Neural Networks, opening the black box. Metodología de las Ciencias del Comportamiento 4(1) 77–93.
33. Nwana HS, Ndumu DT, Lee LC, Collins JC (1999) ZEUS: A Toolkit for Building Distributed Multi-Agent Systems. Applied Artificial Intelligence Jounal, vol 1, n°13, pp. 129–185.
34. Odell J, Levy R, Nodine M (2004) FIPA Modeling TC: Agent Class Superstructure Metamodel. FIPA meeting and interim work.
35. Odell J, Huget MP (2003) FIPA Modeling: Interaction Diagrams.
36. Olivia C, Chang CF, Enguix CF, Ghose AK (1999) Case-Based BDI Agents: An Effective Approach for Intelligent Search on the World Wide Web, AAAI Spring Symposium on Intelligent Agents, 22–24.
37. OMG Unified Modelling Language Specification. Version 1.3. http://www.omg.org.
38. O'Reilly J.E., S. Maritorena, B.G. Mitchell, D.A. Siegel, K.L. Carder, S.A. Garver, M. Kahru, C. McClain, 1998. Ocean color chlorophyll algorithms for SeaWiFS. Journal of Geophysical Research-Oceans, **103**(11), 24937–24953
39. Perner P (1998) Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation. Lecture Notes in Computer Science 1488, pp 251–261.
40. Perner P (2001) Why Case-Based Reasoning Is Attractive for Image Interpretation. Lecture Notes in Computer Science 2080, pp 27–43.
41. Pokahr A, Braubach L, 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. Jadex
42. Rao AS, Georgeff MP (1995) BDI Agents: From Theory to Practice. First International Conference on Multi-Agent Systems (ICMAS-95). San Franciso, USA.
43. Sarmiento JL, Dender M (1994) Carbon biogeochemistry and climate change. Photosynthesis Research, Vol. 39, pp. 209–234.
44. Takahashi T, Olafsson J, Goddard JG, Chipman DW, Sutherland SC (1993) Seasonal Variation of CO2 and nutrients in the High-latitude surface oceans: a comparative study. Global biochemical Cycles. Vol. 7, No. 4. pp. 843–878.

45. The Beam Project. http://www.brockmann-consult.de/beam/.
46. Wendler J, Lenz M (1998) CBR for Dynamic Situation Assessment in an Agent-Oriented Setting. Proc. AAAI-98 Workshop on CBR Integrations. Madison (USA).
47. Wooldridge M, Jennings NR (1995) Agent Theories, Architectures, and Languages: a Survey. In: Wooldridge and Jennings, editors, Intelligent Agents, Springer-Verlag, pp. 1–22.
48. Wooldridge M, Jennings NR, 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.
49. Zeichen M.M., Robinson I.S. (2004). Detection and monitoring of algal blooms using SeaWiFS imagery (vol 25, pg 1389, 2004). Int. J. Remote Sens. 25 (9): 1797–1798.

9

Extracting Knowledge from Sensor Signals for Case-Based Reasoning with Longitudinal Time Series Data

P. Funk and N. Xiong

Department of Computer Science and Electronics
Mälardalen University
SE-72123 Västerås, Sweden
{peter.funk, ning.xiong@mdh.se}

Summary. In many industrial and medical diagnosis problems it is essential to investigate time series measurements collected to recognize existing or potential faults/diseases. Today this is usually done manually by humans. However the lengthy and complex nature of signals in practice often makes it a tedious and hard task to analyze and interpret available data properly even by experts with rich experiences. The incorporation of intelligent data analysis method such as case-based reasoning is showing strong benefit in offering decision support to technicians and clinicians for more reliable and efficient judgments.

This chapter addresses a general framework enabling more compact and efficient representation of practical time series cases capturing the most important characteristics while ignoring irrelevant trivialities. Our aim is to extract a set of qualitative, interpretable features from original, and usually real-valued time series data. These features should on one hand convey significant information to human experts enabling potential discoveries/findings and on the other hand facilitate much simplified case indexing and similarity matching in case-based reasoning. The road map to achieve this goal consists of two subsequent stages. In the first stage it is tasked to transform the time series of real numbers into a symbolic series by temporal abstraction or symbolic approximation. A few different methods are available at this stage and they are introduced in this chapter. Then in the second stage we use knowledge discovery method to identify key sequences from the transformed symbolic series in terms of their cooccurrences with certain classes. Such key sequences are valuable in providing concise and important features to characterize dynamic properties of the original time series signals. Four alternative ways to index time series cases using discovered key sequences are discussed in this chapter.

9.1 Introduction

Case-based reasoning (CBR) [1] has been widely recognized as a powerful learning methodology for circumstances where generalized domain knowledge is not available or hard to obtain. Based on the tenet that similar problems