

AN INTELLIGENT DISTRIBUTED SYSTEM FOR ENVIRONMENTAL MANAGEMENT*

BENEDITA MALHEIRO⁽¹⁾, EUGÉNIO OLIVEIRA⁽²⁾

*(1) Dept. of Electrical Engineering
Institute of Engineering of Porto
Rua de S. Tomé
4200 Porto, Portugal*

*(2) Dept. of Electrical Engineering and Computers
Faculty of Engineering,
University of Porto
Rua dos Bragas - 4099 Porto CODEX, Portugal*

ABSTRACT. Environmental management is a complex task. The amount and heterogeneity of the data needed for an environmental decision making tool is overwhelming without adequate database systems and innovative methodologies. As far as data management, data interaction and data processing is concerned we here propose the use of a Geographical Information System (GIS) whilst for the decision making we suggest a Multi-Agent System (MAS) architecture.

With the adoption of a GIS we hope to provide a complementary coexistence between heterogeneous data sets, a correct data structure, a good storage capacity and a friendly user's interface. By choosing a distributed architecture such as a Multi-Agent System, where each agent is a semi-autonomous Expert System with the necessary skills to cooperate with the others in order to solve a given task, we hope to ensure a dynamic problem decomposition and to achieve a better performance compared with standard monolithic architectures.

Finally, and in view of the partial, imprecise, and ever changing character of information available for decision making, Belief Revision capabilities are added to the system.

Our aim is to present and discuss an intelligent environmental management system capable of suggesting the more appropriate land-use actions based on the existing spatial and non-spatial constraints.

1. Introduction

This chapter begins by providing the reader with the necessary framework for the understanding of the involved concepts and then presents and discusses the proposed environmental decision tool. The Geographical Information Systems overview (section 2) is followed by a Belief Revision short presentation (section 3). Section 4 is dedicated to Distributed Artificial Intelligence, in general, and Multi-Agent Systems, in particular. Those familiar with these themes should proceed to section 5, where the implementation of the Intelligent Environmental System, as a whole, is discussed. Finally, on section 6 we summarise our conclusions.

1.1. MOTIVATION

The idea of interconnecting both Multi-Agent Systems and Geographical Information Systems emerged from the analysis of both systems' features. On one hand, in a Geographical Information

* In Environmental Informatics, Methodology and Applications of Environmental Information Processing, Eds. N. Avouris and B. Page, Computer and Information Science, Vol. 6, Kluwer Academic Publishers, 1995.

System, the data is distributed over different layers of information which, whenever integrated, provides new knowledge dimensions, on the other hand, the knowledge in a Multi-Agent Systems is distributed over different agents which cooperate among themselves, solving complex problems that they would never be able to do individually.

Through the Multi-Agent System - Geographical Information System combination the Intelligent Environmental System will include a diversity of competencies on different subjects - geology, hydrology, landscape, climate, historical and social factors, etc. - working together in an integrated and cooperative way, aiming at producing the best advice for the use of the environment.

In systems in which several agents cooperate with one another within a decentralised control regime, the information management problem is exacerbated still further - each agent has to contend with deficiencies and changes in the information supplied by its contemporaries as well as in its own local information [Malheiro, 1994]. Our intention is to cope with these difficult problems through the adoption of suitable cooperation policies: task and result sharing, belief revision and conflict resolution.

This means that we see the distribution problem not just at lower level, implying the quest for higher computational performances, but also at more abstract and semantically significant higher level, implying true cooperation among different experts, acting as semi-autonomous agents.

By providing automatic, sound, credible suggestions in such a sensible and complex domain, this decision making tool is particularly well suited for regional planning authorities, private developers and central administration.

2. Geographical Information Systems

2.1. INTRODUCTION

We will briefly present and explain the principles of geographical information systems. The description of its major components, including raster and vector data structures, methodologies for classification, modelling and spatial analysis are only generically introduced, since our aim is to provide the reader with an overall idea of the main features of the system rather than a detailed and exhaustive one.

Until computers were applied to mapping, every spatial database was either a drawing on a piece of paper or a film. The information was encoded as a set of points, lines or areas. These different graphic entities were displayed using a variety of colours, symbolisms or text codes whose meaning was explained in an accompanying legend [Burrough 1992].

On one hand, the database support (the paper map) conditioned the amount and complexity of the data to be displayed, and on the other hand, the map production was an expensive and time consuming activity.

New data acquisition methods were adopted, namely remote sensing (satellite and aerial photography imagery), to map large areas with great accuracy. These products of airborne and space sensors are no longer maps, in the original sense of the word, but digital data. This data is not in the familiar form of points, lines or areas, usually used to describe classified features of the earth's surface, but encoded picture elements (pixels) in a two-dimensional matrix containing the intensity value of the reflected electromagnetic radiation in a given band. New tools were needed to process this information and convert it into a more meaningful representation. The Geographical Information Systems (GIS) emerged as the mapping integration tool for remote

sensing, earthbound survey and cartography, providing the user with an extensive set of functions to process and analyse the large complex spatial database.

2.2. GEOGRAPHICAL INFORMATION SYSTEMS OVERVIEW

The use of computers for spatial analysis and mapping was accompanied by several developments in related areas like automated data capture, data analysis and cadastral, topographical mapping, thematic cartography, mathematical studies of spatial variation, soil science, surveying and photogrammetry, remote sensing and image analysis. Essentially, all these fields needed a powerful set of tools for collecting, storing, retrieving, transforming and displaying spatial data from the real world. This set of tools constitutes the Geographical Information System.

In these systems, real world entities are represented by their position in a coordinate system, a set of attributes (name, area, colour, etc.) and the topological relations with other elements (e.g. this point is part of a line that defines an area).

Some people may think that they are just another computer graphics tool. This simply reveals a complete misunderstanding of the whole GIS concept. Although both systems represent, manipulate and relate graphic entities (points, lines and polygons) to the same coordinate system, handle non-graphic attributes and describe topological relations, in the GIS case, the heterogeneity and size of the data sets require specialised data analysis methods.

The GIS is much more than just an adequate tool for encoding, storing and retrieving data about aspects of the earth's surface. It provides the user with "what if?" analysis (simulations) for studying environmental processes, analysing results of trends or anticipating the outcome of planning decisions, enabling planners and decision makers to explore a wide range of possible scenarios.

Existing geographical information systems draw up maps faster and cheaply, produce user specific maps, allow simulation, easy map update and facilitate data analysis.

2.3. THE COMPONENTS OF A GEOGRAPHICAL INFORMATION SYSTEM

Geographical information systems have three main components: computer hardware, application software modules and proper organisational context.

Typically, the computer hardware required for a geographical information system includes a central processing unit with a disk drive storage unit and some peripheral units like digitising tables, scanners, plotters and printers.

Data input covers every digitising activity from the transformation of data captured in the form of maps, to field observations and sensors (satellite and aerial imagery).

Data storage and database management are concerned with the structure and organisation of the topological data and the geographical entities attributes. Geographical information systems rely on Database Management Systems (DBMS) to organise the database.

Data output and presentation are left to the user's choice. A variety of maps, charts, tables ranging from CRT visualisation to plotter or printer outputs are available.

Data transformation embrace transformations needed to remove errors, update or match with other data sets, and the large array analysis methods that have to be applied in order to answer the queries posed by the user. Transformations can occur both on spatial and non-spatial data, either separately or in combination. They include scale changing, data fitting to new projections,

logical retrieve of data, calculation of areas, perimeters and volumes, as well as other domain specific transformations [Burrough 1992].

A geographical information system is expected to answer to a wide range of user questions that will require a certain amount of data transformations and calculations. Some of these questions may be the following:

- Where is object A?
- Where is A in relation to B?
- How many occurrences of type A are there within distance D of B?
- What is the value of function F at position P?
- How large is B (area, perimeter)?
- What is the result of intersecting various types of spatial data?
- What is at positions P1, P2, ...?
- What objects are next to ...?
- Reclassify the objects according to a specific set of attributes.
- Using the database as a model of the real world, simulate the effect of process P over time T for a given scenario S.

2.3.1. *Data Structures.* Geographical data has specific data structure requisites since it includes information about position, possible topological connections, and attributes of the represented objects.

There are two different ways of representing spatial data in a computer: either the spatial entities are described in terms of a grid, where the grid elements that include the object are assigned a numeric value, or they are represented as a set of vectors that describe the form of the object. The first spatial representation is called raster while the second is called vector. The two fundamental ways of representing topological data can be summarised as follows:

- Raster representation - set of cells (pixels) located by coordinates; each cell is independently addressed and contains a value that represents an attribute.
- Vector representation - contains three main geographical entities, points, lines and areas: points are similar to cells, except they do not cover areas; lines and areas are sets of interconnected coordinates that can be linked to given attributes. The attributes of the graphic objects are stored and handled through a database management system (DBMS).

Today's geographical information systems allow the coexistence of both raster and vector data: a relational database structure for points, lines and polygons can be established that treats the raster and vector approaches to modelling the topology as equivalent alternatives.

The underlying principle is that although attribute data and topological data should be kept separate from each other, there must always exist an unique link for their interconnection. The attribute data that describe what the entity represents may include data about its real world attributes, and also about how the entity must be managed within the database. This last can also include information that specifies the type of possible relationships among entities: a point should not have a field indicating its area, a polygon should be defined in terms of the sets of lines or the set of point entities that define it spatially.

Raster data is organised by lines and columns where each cell or pixel can be addressed either by its coordinates (line, column) or by the value stored in that pixel. In vector entities the basic element is the point or vertex that can be either bi-dimensional (X,Y) or three-dimensional (X,Y,Z); points associated in oriented lists with an initial vertex and a final vertex define a line; a set of lines closed over the same point define an area or polygon. To each graphic entity there is

an unique key or label associated. It is through this unique key that alphanumeric attributes are linked to vector entities.

2.3.2. *Methods of Data Analysis and Spatial Modelling.* Transformations such as data cleaning or updating, and scale or projection system changing are trivial to geographical information systems. The major difference between geographical information systems and computer-assisted cartography systems reside in the set of functionalities available for spatial data transformation in order to answer particular queries. These specific capabilities range from simple methods of retrieving database subsets, through statistical analysis methods (clustering and others), to spatial analysis (neighbourhood, likelihood functions, or interpolation, decimation methods).

Specific sets of command sequences or function invocations that tackle particular analytical problems are referred to as models. Many of these models are combined and applied for simulation purposes providing the user with new methods for evaluation and planning. The ease with which a number of options or scenaria can be compared, linked with the possibilities for the simulation of processes over time allows "what if ?" analysis that would otherwise be impossible, make GIS the essential geographical data modelling tool.

2.3.3. *Classification Methods.* Classification may be considered as an essential activity for human understanding. Without the generalisation that occurs during classification, our brain would be overwhelmed with small, uninteresting details.

To classify spatial geographical data into a set of previously defined patterns (forestry, crops, urban areas, etc.) is an essential activity. Typically, the classification procedures operate over the raster data and produce as a result a set of polygons (vector entities). Many of the current geographical information systems provide these classification facilities.

2.4. THE OVERLAY CONCEPT

The overlay concept is natural to cartographers and designers. To model a particular aspect of the real world the user needs to represent in the geographical information system every relevant spatial data available. Usually, a wide range of thematic maps and imagery data are available. The user must select those that are more relevant to his specific needs, so that the combined information will provide him with an integrated overview, reclassification or generalisation.

In a geographical information system the different types of spatial data are stored and organised by layers. These diverse, yet complementary, spatial data sets represent the actual world model: from geological layers to demography layers, passing by satellite imagery layers, road layers, cadastral layers, etc..

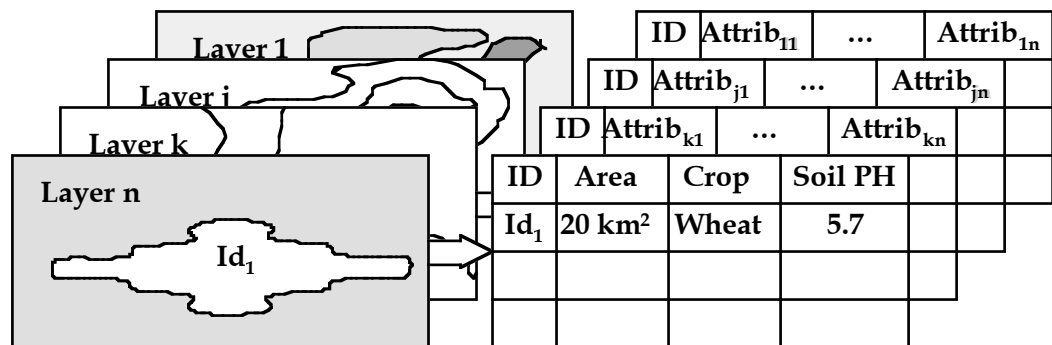


Figure 1 - The GIS overlay concept.

Raster, vector and alphanumeric data are interconnected allowing the simultaneous consultation of the whole information set, constituting an unique geo-relational structure. In principle, the number of layers are unlimited, restrictions being imposed only by storage space limitations.

3. Belief Revision

In real world applications problem solving entities often have to make decisions based on partial, imprecise, and ever changing information.

To keep track of an agent's changing beliefs, researchers have devised a number of different types of Truth Maintenance System (TMS) [Martins, 1990]. Such systems portray as their main features the maintenance of the consistency between their beliefs, the reason for their beliefs and the identification of contradictions.

In a TMS "belief" is taken to mean justified belief - either it is an assumption or it has been deduced from other beliefs. While an assumed fact is believed, an ordinary fact may be believed or disbelieved; and a contradiction is always false. There are a number of different ways in which the dependencies between beliefs can be registered: in justification based TMSs (JTMSs) each belief is associated with the beliefs that immediately caused it [Doyle, 1979]; whereas in an assumption based TMS (ATMS) each belief is associated with the smallest set of environments from which it can be deduced (*the belief's label*) [de Kleer, 1986]. This work concentrates on the ATMS approach because, when compared with the JTMS, it:

(a) enhances the system's efficiency and improves its real time operation because multiple contexts are kept. This makes it faster to move from one valid context to another because the new one does not have to be calculated from scratch.

(b) improves the system's transparency because the belief revision is not contingent on dependency oriented backtracking and hence it will not stop on unresolved circularities or leave nodes unlabeled.

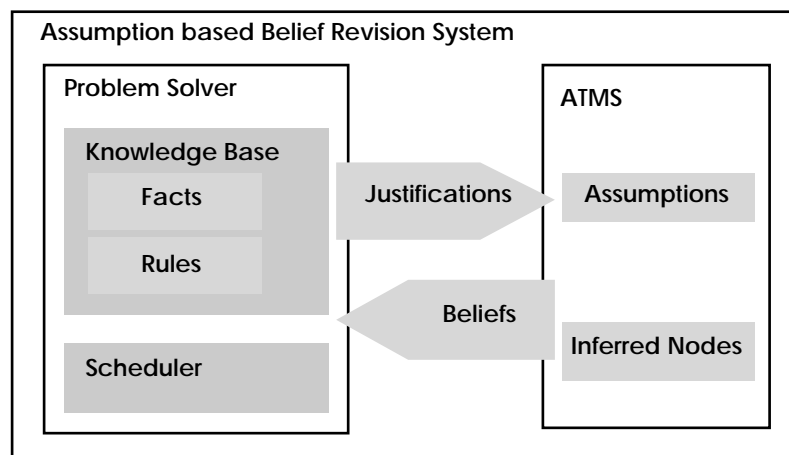


Figure 2 - Assumption based Belief Revision System

ATMSs are composed of three units: (i) the truth maintenance system itself; (ii) the problem solver; and (iii) the interface unit. The TMS guarantees that the conclusions reached by the problem solver are kept updated and consistent. However, it only deals with propositions (usually substituted by arbitrary identifiers called nodes) and their dependencies. For each

proposition there will be a node and for each dependency a justification which describes how the node was deduced from other nodes. The TMS allows four operations: (i) creation of new *assumption nodes*, whenever new propositions are assumed to be true; (ii) creation of new *ordinary nodes*, when a new proposition is deduced by the problem solver; (iii) addition of new justifications to existing nodes, whenever the problem solver finds new ways of deducing them; and (iv) the querying about existing beliefs.

The main purpose of de Kleer's ATMS is to find, as efficiently as possible, the most general environment of a node and the most general version of a contradiction. Adopting this strategy avoids unnecessary work since: (i) by finding the most general version of a contradiction it avoids various solving steps which would only lead to inconsistencies; (ii) by finding the most simple node label it avoids superfluous label updating. This approach guarantees that the least dependent nodes are always deduced before their consequent nodes.

4. Distributed Artificial Intelligence

Artificial Intelligence (AI) researchers have long been concerned with both theoretical and implementational issues related with single agents intelligent behaviour. By agents, we mean independent stand-alone computational processes that are committed to problem solving in a specific domain. A wide spectrum of themes has been explored over the years, ranging from natural language to problem solving, learning, knowledge representation, planning, etc..

Recent technological developments and the continuous increasing complexity of problem solving domains laid ground for a new approach within AI. Powerful concurrent computers, the widespread and generalisation of data networks, together with the result of sociological studies that revealed the inherent task decomposition as well as the cooperative interactions that occur within human teams during problem solving activities, were some of the factors that strongly contributed to the emergence of a new Distributed Artificial Intelligence (DAI) field.

Within DAI there are three main sub-fields: Distributed Problem Solving, Multi-Agent Systems, and Parallel AI. The Distributed Problem Solving (DPS) research is concerned with solving a specific problem through its decomposition in an independent number of modules. These modules cooperate through the division and the sharing of knowledge about the problem and about the developing solution [Lesser 87]. Multi-Agent Systems (MAS) research tackles problem solving via the coordination of the intelligent behaviour among a set of semi-autonomous intelligent agents. Typical Multi-Agent Systems issues are the inter-agent coordination of knowledge, goals, skills, and plans, in order to jointly solve the submitted problems. Agents in a Multi-Agent System can either work toward a single global goal or toward private local goals that interact. Knowledge sharing and process coordination among the agents are essential features. Parallel AI is focused on the development of parallel computer architectures, languages, and algorithms for AI [Bond, 1988].

According to Bond and Gasser, some typical motivations for selecting DAI systems are:

- (a) Adaptability - DAI systems are well suited in face of spatial, logical, temporal or semantic problem distribution.
- (b) Cost - A large number of low cost computer units may be an interesting solution provided communication costs are not relevant.
- (c) Development and Management - The inherent modularity allows different system parts to be developed independently guaranteeing a continuous system extendibility.

- (d) Efficiency or Speed - Concurrency and process distribution over different computer units can increase the speed of computation and reasoning, provided that the coordination overhead is kept within reasonable limits.
- (e) History - The need to integrate existing distributed resources, such as workstation networks or different domain experts.
- (f) Isolation/Autonomy - Local process control of isolated or separated system parts may be regarded as a safety or protection must.
- (g) Naturalness - Some problems are better described in distributed terms, e.g., they are more adequately and naturally characterised as a collection of separate agents.
- (h) Reliability - Distributed systems can exhibit a higher degree of reliability than centralised systems because they provide redundancy, cross-checking, by-pass, and triangulation of results.
- (i) Resource Limitations - Individual computational agents have bounded resources for problem solving, rationality and possibly bounded influence, necessitating cooperation and coordination to solve large complex problems.
- (j) Specialisation - Knowledge or action may be collected in specialised, bounded domains, for purposes of control, extendibility or comprehensiveness.

From the 1986 Workshop on Distributed Artificial Intelligence, a classification for DAI systems emerged [Sridharan 1987]. Systems were characterised according to their degree of heterogeneity, autonomy, granularity, distribution and interaction, as well as control strategy, scale, and resources.

4.1. MULTI-AGENT SYSTEMS

In DPS systems, agents generally share an overall global goal, have a common language and semantics, and problem solving is an activity that can only be achieved by the community as a whole, since single agents are not capable of solving a given problem by themselves.

In Multi-Agent Systems, agents generally display a relative degree of autonomy, share the same environment, compete for limited resources (e.g. time, space, tools), coordinate their activities for efficiency reasons (benefiting from other agents results, assisting other agents) and try to avoid conflicts. In some extreme situations, agents are able to solve some given problems individually. They are not necessarily bounded to a common language or semantics, provided they are able to communicate through a common protocol. Multi-Agent Systems characteristics may be summarised as exhibiting a higher degree of heterogeneity and autonomy than DPS systems.

Large systems are built in a distributed fashion in order to master complexity. Ideally, this should mean a separation of control and execution to make the control part more explicit and maintainable, reducing thereby the complexity. The cooperative systems approach is to split the overall system into several smaller and dedicated systems where both the execution and the control are decentralised. Each dedicated system, not only, has to control its local activity, but also, has to control the coordination with the other systems. These dedicated cooperative systems are referred to as agents.

4.1.1. *Multi-Agent System Architecture.* A variety of system and agent architectures have been studied and implemented for Distributed Intelligent systems. The proposed Multi-Agent System architecture is based on the ARCHON (ARchitectures for Cooperative Heterogeneous ON Line systems) project architecture [Wittig 1992].

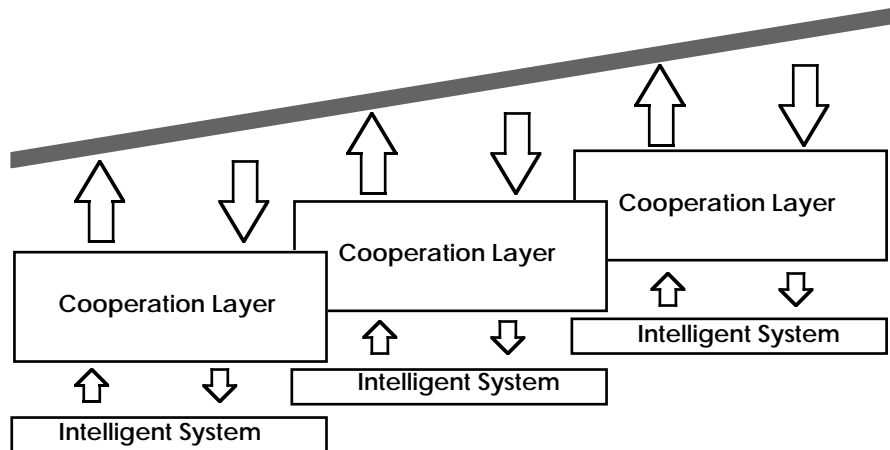


Figure 3 - System Architecture

The agents consist of two layers: the intelligent system or problem solver layer, and the cooperation layer. The intelligent system layer is the domain system and is simply called Intelligent System (IS). Although, the Intelligent Systems are mainly Knowledge Based Systems, it does not mean that every Intelligent System has to be an Expert System. The Cooperation Layer (CL) has a dual function: inter-agent and intra-agent coordination.

4.1.1.1. *Cooperation Layer*. The functionality related to cooperation is represented as a distinct problem solving layer which sits above the Intelligent System layer (Figure 4). The cooperation layer has the following components [Wittig, 1992]: (i) a cooperation module; (ii) a communication module which sends/receives asynchronous messages between the agents; (iii) a self model which represents information about the underlying domain level system; and (iv) a set of acquaintance models which represent the relevant information about the other community members with which the agent can be expected to interact.

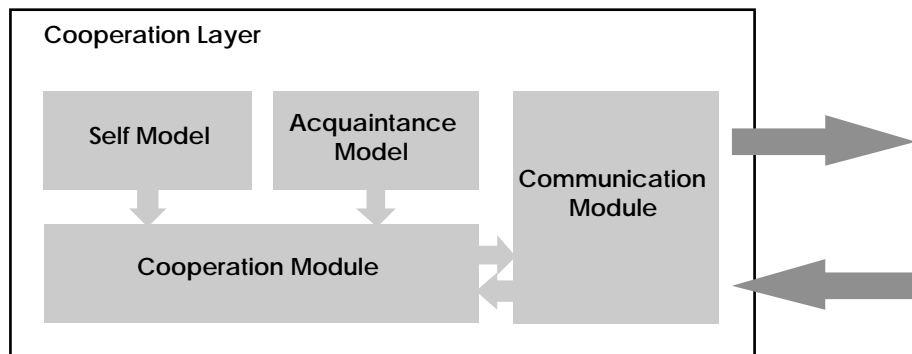


Figure 4 - Cooperation Layer Architecture

The Cooperation Module is responsible for determining when cooperation is necessary and for deciding what strategy should be employed for each social interaction. Cooperation can be viewed from two perspectives: from the organiser's point of view and from the respondent's point of view [Oliveira, 1993]. The organiser starts the cooperation based on its needs and views. This includes asking for assistance (*task sharing*) and supplying voluntary help (*result sharing*). Task sharing is initiated when the organiser has an activity that it cannot accomplish alone and so it looks for help within the community - the agent which accepts the task is the respondent. Result

sharing is initiated when the organiser generates information which it believes will be useful to others, based on its acquaintance model.

The Communications Module provides the necessary physical channels and the high-level protocol needed to support cooperative problem solving among heterogeneous agents.

In order to participate effectively in a multi-agent system, an agent must know which of its tasks can be carried out without assistance (*independent tasks*) and which of its tasks depend on other agents in some way (*dependent tasks*). This information is represented in the agent's self model: `who_am_i` gives the agent's identification; `my_conclusions` gives the tasks the agent is capable of performing; `i_know_about` gives the facts the agent knows about; and `my_goal` gives the requirements and results of each task.

The acquaintance models provide agents with information about who can assist them with their dependent tasks (`who_knows_about`) and who they can assist by volunteering useful information (`is_interesting_to`).

4.1.1.1. *Intelligent System Layer*. The IS is typically a domain problem solver and constitutes the agent problem solving contribute to the overall community. During the execution of independent tasks only local data processing occurs, while during dependent tasks external data processing is also involved. The IS function is to provide the agent with relevant results in order to accomplish its tasks.

4.2. COORDINATION POLICIES

Agents within a community must have some coordination policies or strategies. The simplest request for help may generate a conflict that will require some type of negotiation. Every time an agent receives none or different results concerning a specific item of knowledge, a conflict exists.

4.2.1 *Conflict Resolution*. Conflicts are always possible to happen in a cooperative multi-agent community [Oliveira 1993]. Conflicts may appear either during task sharing or result sharing operations.

Nevertheless, the most important case happens during result sharing when several agents, that are performing the same task, produce either inconsistent or antagonistic results. Here we are referring to different alternative answers to the same request (they can also be seen as different values for the same receiver's task input).

The ability to master these situations (resolving the conflicts) rely mostly on the availability of a functionality for the measurement of the information quality.

There are positive and negative conflicts that occur both during task and result sharing:

- Positive conflicts during task sharing occur when more than one agent is capable of executing the specific requested task. This situation has to be solved via negotiation.
- Negative conflicts during task sharing occur when there are no agents capable of executing the requested task.
- Positive conflicts during result sharing occur when several agents produce different but complementary results for the same request (or similar results with different credibilities associated).
- Negative conflicts during result sharing occur when several agents, performing the same task, produce either inconsistent or antagonistic results. Such situations have to be solved via negotiation.

4.2.2. *Negotiation*. Negotiation is a very important feature agents must exhibit in order to resolve conflicts. Cooperation, mainly task sharing, may lead to the need for negotiation. Once an agent wants to formulate a request, in order to get an answer from other agents, it needs to establish a protocol with the possible respondents for that specific request. This protocol will permit the exchange of information between the agent interested in the answer and the possible respondents. During this protocol the Organiser informs the other agents about the constraints associated to that request (mainly a deadline) and the potential contracted agents will respond informing about the expected time needed for having the task executed and the expected quality of the answer (bid message). It is up to the organiser to evaluate the bid messages and to contract one (or possibly more) respondent [Oliveira 1993].

4.3. LOCALLY CONSISTENT MULTI-AGENT BELIEF REVISION SYSTEMS

Whilst the Belief Revision systems discussed in section 3 are generally sufficient for maintaining beliefs in an asocial context, they need to be extended if they are to be used in a social context. For example, as well as beliefs that an individual has generated for itself, there will be beliefs about which he has been informed by other community members (either because an acquaintance has answered a query or because it has volunteered a piece of relevant information).

Some previous distributed belief revision systems have been implemented, namely by Mason and Johnson (a Distributed ATMS where the interchange between the agents includes not only the shared data, but also the shared data labels and the invalid assumptions sets)[Mason, 1989], and by Huhns and Bridgeland (a distributed JTMS that provides local and shared consistency, that may leave some nodes unlabelled) [Huhns, 1991].

Global consistency in a distributed belief revision system means complete consistency between every agents' conclusions all the time. However attainment of this level of consistency depends not only on the system's architecture and design but also on the amount of inter-agent communication which is acceptable. Whenever a centralised architecture is appropriate it is reasonable to build a global ATMS which incorporates all of the system's facts and justifications; whereas in the case of a distributed architecture a pragmatic compromise between the achieved consistency level and the information redundancy among the agents has to be reached. These two architectural options give rise to two fundamental approaches to belief revision in a multi-agent system: *global consistency* and *local consistency*, respectively. In the first case, two or more agents cannot assign a different belief status to the same fact. In the second case, different agents may have different perspectives over the same fact if conveniently justified.

In multi-agent systems the semi-autonomous agents each have their own repositories where they record local propositions and justifications. Only when cooperation occurs do non-local facts have to be represented. In such an environment global consistency is unattainable, unless the system broadcasts every relevant activity to all the pertinent agents, therefore we settled for local consistency.

Given this stance, a new crucial issue arises: the question of how to include external propositions in an agent's local dependency network. Depending on the scheme chosen for attaining local consistency, an agent that receives an external fact may or may not receive its label: if the label is sent, it is possible to guarantee the consistency between the foundations of the external fact and the local facts and assumptions; if no label is sent, it is impossible to cross check the external fact's foundations with the local ATMS data. In the first case the agents exhibit *local-and-shared well-foundedness* and local consistency, whilst in the second case there is only *local well-foundedness* and local consistency [Huhns, 1991]. We chose the latter because we believe it is

more appropriate for modelling real world semi-autonomous agents that have their own beliefs, desires and intentions. Consequently, the community of agents behaves like a democratic society in which each individual can hold a different opinion once it is locally justified - an agent only accepts to revise its beliefs based on external information when it does not have its own convictions regarding that fact.

Using this scheme, a given agent's beliefs have to be divided into two separate sets:

- *private beliefs* that the agent has generated and kept to itself.
- *shared beliefs* that the agent has in common with at least one acquaintance.

Within a particular agent, a shared belief can either be *internal* (endogenous) or *external* (exogenous). The former means that the agent has deduced the fact for itself; the latter means that the agent has received the information from an acquaintance. This classification is central to the belief revision process because:

- private beliefs have a local scope and are automatically revised by the agent's ATMS.
- shared beliefs are revised only by the agent that created them (i.e. agents where the facts are classified as shared internal). The revision action is performed by the originating agents' ATMS module and then the updated beliefs are resent to all the acquaintances where they are known as shared external.

The shared facts belief revision is supported by the Cooperation Layer. A cooperative interaction is started, not only, when an agent needs assistance or when an agent is able to supply help, but also, when belief revision of shared knowledge occurs.

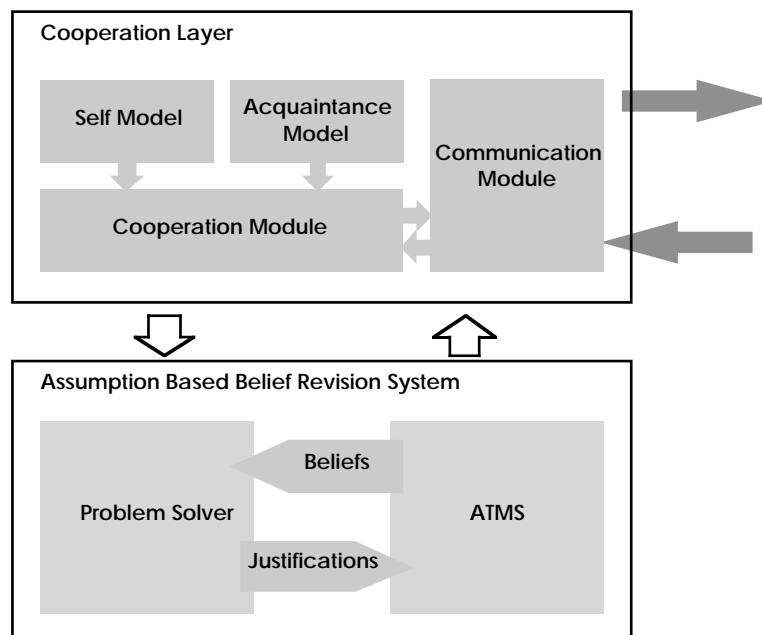


Figure 5 - Belief Revision Agent Architecture

The aforementioned multi-agent belief revision system is implemented. The agents Intelligent Systems are assumption based belief revision systems and their Cooperation Layers provide the necessary facilities for establishing, maintaining and monitoring cooperation and belief revision [Malheiro 1993]. The type of agent described here is well suited for agents with some degree of domain overlap. If the inter-agent domain overlap is high local consistency is not the best choice.

5. Intelligent Environmental System

As it was pointed out earlier in this chapter the Geographical Information System - Multi-Agent System interconnection is almost intuitive due to both systems inherent distributiveness. The architecture of the proposed Intelligent Environmental System is presented in the following figure.

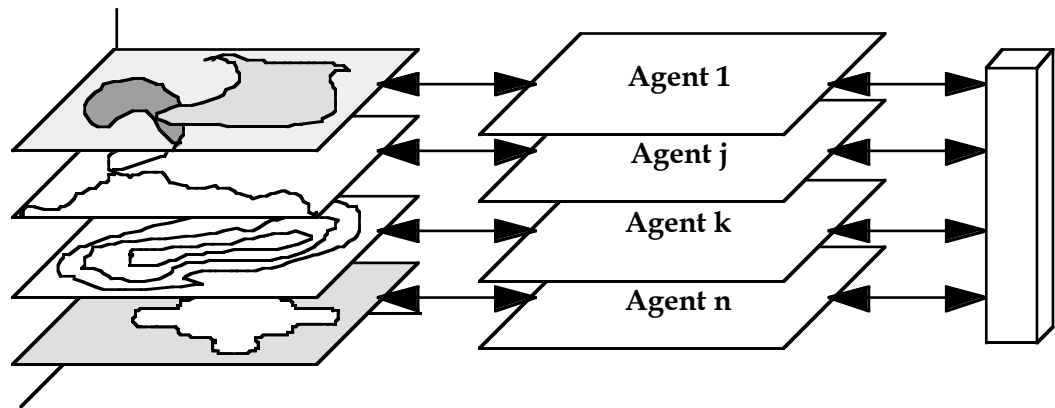


Figure 6 - System architecture

Although the adopted system architecture was inspired on the ARCHON project architecture, only the Cooperation Layer remains similar to the original project. The major differences lay on the new Intelligent System layer, as well as on the knowledge base facts or resources database. Agents are connected to spatial data layers and are responsible for the knowledge management of their spatial layer database.

The adopted agent architecture is based on the belief revision agent architecture presented in subsection 4.3.. Through this selection we hope to provide the system with an intelligent dynamic behaviour, sensible to database updates and capable of avoiding contradictions.

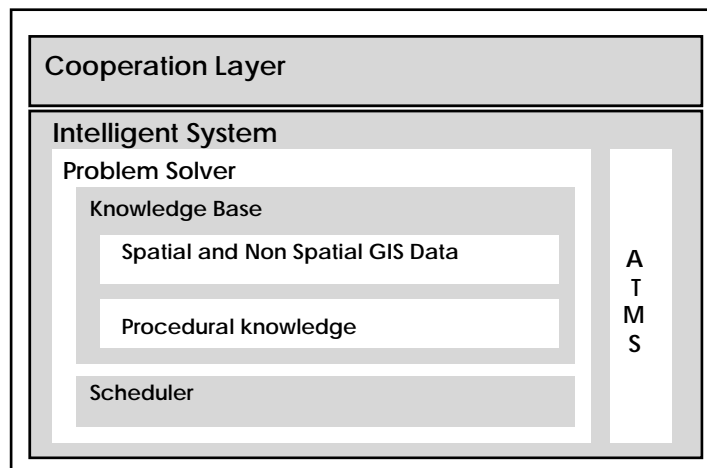


Figure 7 - Agent Architecture

The agents are semi-autonomous experts on the sub-domains associated with the GIS data they are connected to: the specific spatial data layer and its alphanumeric attributes.

5.1 THE INTELLIGENT SYSTEM.

The agent Intelligent System is a knowledge based system responsible for producing inferences and executing tasks based on the spatial data layer and respective database attributes. It is an Assumption based Belief Revision System of the type described in sub-section 4.3, enabling local consistency.

The knowledge represented contain the environmental constraints of that specific database subset needed for decision making. An agent attached to a geology data layer should exhibit a similar behaviour to a Geology Expert, while a cadastral agent ought to act like a regional planning agency or a transports network agent that should decide according to the transports bureau policies.

If a planner decides to build a new road, the system should be able to suggest the best location from the environmental point of view or even advise against its construction if the system has no evidence of its need. In such a scenario, the system would consult for example the geology agent, the elevation agent, the roads agent and the flora agent, in order to establish, according to the 3D model given by the elevation agent, the geological characteristics of the area provided by the geology agent, the existing flora types detailed by the flora agent, and the existing transport network and road construction knowledge contained in the roads agent, the optimal road location.

5.2. THE COOPERATION LAYER

The set of semi-autonomous agents cooperate in order to provide coherent management of information in an environment where many different types of attributes are used to characterise the same reality. The agents are knowledgeable about information resources that are local to them, and cooperate to provide global access to, and better management of the information. To cooperate effectively, the agents must have models of themselves, models of their acquaintances as well as models of the resources that are allocated to them. These resource models are in the present case schemes of the database assigned to the agent [Huhns, 1994].

Geographical Information System provides the system with both a common semantic and syntax (SQL is the interface language), as well as the same structured environment. A common ontology is used to map the different agents database spatial elements and their attributes in order to allow cooperation within the community. The new inter-agent knowledge mapping functions have been added to the every agent Acquaintance Model. Only the mapping functions that are relevant to the local agent activity are represented in his Acquaintance Model. With this procedure we avoid knowledge redundancy.

The Self Model and the Acquaintance Model provide the functionalities already described in 4.1.1.1..

The selected coordination policies (sub-section 4.2.) together with the belief revision capabilities provide the agents with the necessary mechanisms to solve and avoid conflicts that can occur in many DAI systems.

6. Conclusions

With the proposed system we hope to:

- (a) Implement an intelligent application capable of evaluating the environmental performance of a case-study area,

- (b) Use a Geographical Information System (GIS) for the correct modelling of the study area and that also constitutes the system's Facts' Knowledge Base,
- (c) Show the relevance of the belief revision in achieving a more accurate model of dynamic worlds,
- (d) Demonstrate the reusability and case study independence of the developed prototype,
- (e) Present an efficient, user friendly system, capable of becoming a reliable tool in the environmental domain.
- (g) Assess the appropriateness of distributed architectures for complex domains,
- (h) Evaluate how adequate are Geographical Information Systems to Distributed Cooperating Systems.

The correct modeling of a geographic area implies the management of large quantities of heterogeneous data. By selecting a GIS for data management, we aim to provide:

- the coexistence between heterogeneous data sets,
- an adequate data structure,
- a good storage capacity and
- a friendly user's interface.

By choosing a distributed architecture such as a Multi-Agent System, where each agent is a semi-autonomous Expert System with the necessary skills to cooperate with the others in order to solve a given task, we hope to ensure a dynamic problem decomposition and achieve a better performance than with a typical monolithic architecture.

Finally, the affinities between Geographical Information and Multi-Agent Systems make us believe in the good results achievable through their interconnection:

- in the Geographical Information Systems the data is distributed over different layers of information and when consulted as a whole it provides new knowledge dimensions,
- in the Multi-Agent Systems the knowledge is distributed over different agents that cooperate among themselves, thus solving problems that they would never be able to do individually.

We believe that this Distributed Intelligent System will be of great use for the regional authorities, providing them with automatic, sound, credible and easy-to-use capabilities in order to make the suitable decisions in such sensible domains as environmental management.

7. Bibliography

- [Bond, 1988] - Edited by Alan H. Bond and Les Gasser, (1988), "Readings in Distributed Artificial Intelligence", Morgan Kaufmann Publishers.
- [de Kleer, 1986] J. de Kleer, (1986) "An Assumption-based TMS" *Artificial Intelligence* 28 (2), 127-162.
- [Burrough 1992] - P. A. Burrough, (1992) "Principles of Geographical Information Systems for Land Resources Assesment", Monographs on soil and resources survey n° 12, Oxford Science Publications.
- [Doyle, 1979] J. Doyle, (1979) "A Truth Maintenance System" *Artificial Intelligence* 12, 231-272.
- [Huhns, 1991] M. N. Huhns and D. M. Bridgeland, (1991) "Multi-Agent Truth Maintenance" *IEEE Trans. on Systems, Man and Cybernetics* 21 (6).
- [Huhns, 1994] M. N. Huhns, M. Singh, T. Ksiezyk, N. Jacobs, (1994) "Global Information Management via Local Autonomous Agents" *Proceedings of the 13th Workshop in Distributed Artificial Intelligence*, Seattle.

- [Lesser, 87] V. Lesser and D. Corkill, (1987), "Distributed Problem Solving" , edited by Stuart C. Shapiro, Encyclopedia of Artificial Intelligence, John Wiley and Sons, New York.
- [Malheiro, 1993] - B. Malheiro, (1993), "Belief Revision in Multi-Agent Systems" MSc Thesis, University of Porto.
- [Malheiro, 1994] - B. Malheiro, N. Jennings, E. Oliveira, (1994) "Belief Revision in Multi-Agent Systems", Proceedings of the ECAI, Amsterdam.
- [Martins, 1990] J. P. Martins, (1990) "The Truth, The Whole Truth, and Nothing But The Truth" AI Magazine, Special Issue, 7-25.
- [Mason, 1989] C. Mason and R. Johnson, (1989) "DATMS: A Framework for Distributed Assumption Based Reasoning" in Distributed Artificial Intelligence Vol II (eds. L Gasser and M. N. Huhns) 293-317.
- [Oliveira, 1993] E. Oliveira, F. Mouta, A. Rocha (1993) "Negotiation and Conflict Resolution Within a Community of Cooperative Agents" Proc. of the First International Symposium on Autonomous Decentralized Systems, Japan, 1993.
- [Sridharan, 1987].N. Sridharan, (1987), "Report on the 1986 Workshop on Distributed Artificial Intelligence", AI Magazine, Fall 1987.
- [Wittig, 1992] T. Wittig (ed.), (1992) "ARCHON: An Architecture for Cooperative Multi-Agent Systems" Ellis Horwood.