Effective cooperative entities in a transport multimodal information system using mobile agents

HAYFA ZGAYA
LAGIS UMR 8146
Ecole Centrale de Lille
Cité Scientifique – BP 48
59651 Villeneuve D’Ascq cedex
FRANCE

SLIM HAMMADI
LAGIS UMR 8146
Ecole Centrale de Lille
Cité Scientifique – BP 48
59651 Villeneuve D’Ascq cedex
FRANCE

Abstract: - A Multimodal Information System aims to provide transport customers pertinent real time information during their travels. The world Multimodal means different types of transport devices but we choose to expand accessible services to various kind of information (transport, weather, stock exchange, itinerary, traffic, perturbations…) in order to provide travellers required information in any chosen data format, taking into account their mobility. However, the exponential growth of services and information providers available on large networks such as the Internet make the realisation of the expected structure very difficult. Therefore, the complexity of such systems can be properly described in terms of interaction between different kinds of autonomous entities. In this context, we propose a multi-agent system describing agents’ behaviours and interactions, using several approaches such as evolutionary algorithms and mobile agents’ technology which are already published in previous works. In this paper, we focus on the internal structure of our proposed system and we detail behaviours of agents responsible of the data flow management.

Key-Words: - Multi Agent System, cooperation, Multimodal Information System, Multimodal transport Network, Mobile agents.

1 Introduction

According to the exponential growth of the services and information available on large networks such as the Internet, transport client travellers require relevant, interactive and instantaneous information during their travels. A Multimodal Information System (MIS) can propose them a support tool in order to make available the needed information. The world Multimodal means different types of transport devices but we choose to expand accessible services to various kind of information (transport, weather, stock exchange, itinerary, traffic, perturbations…). The main goal of such system is to guide and help network customers to make good decisions when they’re travelling providing them all needed information in any existent and chosen data format (text, multimedia…). Information is represented by distributed data located on different nodes through a Transport Multimodal Network (TMN). These nodes represent servers which correspond to our system’s information providers. According to their eventual mobility, users need to access to relevant data through handheld wireless devices such as PDAs, laptops, cell phones, etc. Networks that connect this kind of devices suffer from low bandwidth and a high incidence of network errors. Mobility can also result in the loss or degradation of wireless connections.

In a previous work [1], we proposed a skeleton of a MIS based on a special kind of software agent called Mobile Agent (MA). A MA is a program composed of code, data and a state, able to move from a node to another through a network to perform tasks and finally return to its original node called Home [3]. They are inherently distributed software entities that reduce the load on the network when they move. In addition, MAs support disconnected operations because they continue to execute after they move, even if they lose network connectivity with their dispatchers [11]. So by employing MAs, handheld wireless devices could provide a reliable technology for message transport over wireless links. In this paper, we present a complete architecture of a MIS based on the cooperation of different kinds of autonomous entities to reach the same goal which is the satisfaction of the users. This paper is organized in six parts. In the following section, we situate the context of our work describing the problem to resolve. Then, we present in section 3 the global system functioning. In section 4, we describe the adopted cooperation model. Experimentations through a flexible multi-agent platform are presented in section 5. Finally, conclusion and prospects are mentioned in last section.

2 Problem Formulation

New distributed applications are spread on huge networks, employing heterogeneous and voluminous data sources located on geographically separated sites.
Through such networks, different actions (navigation, requests, storage, update, etc) can be launched intensely in a simultaneous way, requiring the access to several heterogeneous and distant data sources. This behaviour can be illustrated by different kinds of interaction among individual entities [4]. Thus, a multi-agents approach seems to be the most appropriate to implement the complex character of such a system. Distributed applications through wide networks are not easy to realize because of the limited aspect of bandwidth which remains restricted by several technical factors [5]. Therefore, permanent connexion can’t be constantly available. That’s why, we have to manage data information availability despite recurrent connexions in order to access and share distributed data. In this context, mobile technology can be complementary to the artificial intelligence because it can reduce considerably network traffic [2]. Providing a software agent the mobility character, gives it the possibility to migrate towards any node on the network which can receive mobile entities. This work belongs to a national French project VIATIC.MOBILITE from the industrial cluster i-trans [12]. In our proposed system, we use Intelligent Mobile Agents (IMAs) to collect needed information through the network from the subscribed information providers. The assignment of network nodes to IMAs to build their routes is called initial Workplans building process which was developed in [6]. After that, a sub-set of these nodes have to be selected to assign tasks, in order to satisfy users, building final effective Workplans. These ones, corresponding to the assignment of servers from S' to tasks from I', (see the formulation below), are computed thanks to an evolutionary approach [7]; the correspondent solution is an instance of a Flexible Tasks Assignment Representation (FeTAR).

A task is an independent sub-request which belongs to one or several customers’ requests formulated simultaneously. A request k is characterized by its due dates d_k which must be respected (figure 1). After that, information providers which propose services corresponding to identified tasks are recognized (figure 2). Finally, servers must be assigned, in the optimal way, to tasks in order to satisfy all asked services in real time knowing that a user is satisfied if his request was completely and rapidly answered with a reasonable cost.

This is a combinatorial problem with an exponential complexity, it is defined by:
- R requests, waiting for responses at the same instant t.
- The set of independent I tasks representing all proposed services on the MN is noted I,
- The realization of each task T_i ∈ I requires a resource, or node, selected from a set of J available nodes, noted S= {S_1,..., S_j},
- The set of independent I’ tasks (I’≤1) composing R is noted I’ (I’≤1),
- A request decomposition scheme corresponds the decomposition scheme of R into I’,
- The set of J’ nodes (J’≤J) selected from S, to perform I’ is noted S’ (S’≤S),
- We have partial flexibility, the realisation of each task T_i requires a node selected from a set of nodes which propose the same service performing the task T_i, with different cost, processing time and data size.
- A service is described by:
  i. A processing time P_{ij} of the task T_i on the node S_j. It corresponds to the estimated time to perform the task T_i by means of the resources of S_j,
  ii. The cost of the service Co_{ij} corresponding to the task T_i on the node S_j,
  iii. The data size Q_{ij} corresponding to the size of the information to collect from S_j to response to T_i.
- A same task may be performed differently on several nodes, namely with different processing time, different cost and different responses’ formats. These three characteristics (P_{ij},C_{ij},Q_{ij}) represent successively the first, second and last term of each element of what we call a service table (table 1), describing an example of different proposed services:

![Diagram of Requests' Decomposition](image1)

Fig 1. Requests’ decomposition

![Diagram of Identification of Nodes](image2)

Fig 2. The identification of nodes
3 System Description

To resolve the problem described previously, we propose a system based on the coordination of five kinds of software agents (figure 3):
- Interface Agents (IA): These agents interact with the users of the system allowing them to choose appropriate form of responses to their demands so they manage requests and then display results. When a TMN customer access to the MIS, an agent IA deals with the formulation of his request and then sends it to an identifier agent. This one relates to the same platform to which several users can be simultaneously connected, thus it can receive several requests formulated at the same time. An identifier agent has to identify and to choose nodes which propose services corresponding to customers’ demands.
- Identifier agents (IdA): These agents decompose received requests into sub-requests corresponding, for example, to sub-routes or to well-known geographical zones. Sub-requests are elementary independent tasks to be performed by the available set of nodes (information providers) on the TMN. Each node must login to the system registering all proposed services. A service corresponds to the response to a defined task with fixed cost, processing time and data size. Therefore, the agent IdA decomposes the set of simultaneous available requests into a set of independent tasks recognizing possible similarities, in order to avoid a redundant data collection. The decomposition process occurs during the intervention of the user.
- Scheduler Agents (SA): Several nodes may propose the same service with different cost, processing time and data size. The agent SA has to assign nodes to tasks minimizing total cost and processing time in order to respect due dates (data constraint). Selected set of nodes corresponds to the sequence of nodes building Workplans (routes) of the data collector agents. The agent SA has firstly to optimize collector agents’ number then nodes assignments to different tasks.
- Intelligent Collector agents (ICA): An agent ICA is a mobile software agent which can move intelligently from a node to another through a network in order to collect needed data. This special kind of agent is composed of data, code and a state and has an intelligent behaviour. Collected data should not exceed a capacity threshold in order to avoid overloading. Therefore, the agent SA must take into account this aspect when assigning nodes to tasks. When they come back to the system, the agents ICA transmit collected data to fusion agent.
- Fusion Agents (FA): These agents have to fusion correctly collected data in order to compose responses to simultaneous requests. The fusion procedure needs information on behalf of IdA and SA agents and progresses according to the collected data availability. Each new answer component must be complementary to the already merged ones. Providers are already selected and tasks are supposed independent. Therefore, there is no possible conflict. A response to a request may be complete when a full answer is ready because all concerned components are available; it can be partial when at least a task composing the request was not treated, for example, because of an unavailable service. Finally, there is no response when no component is available. If an answer is partial, the correspondent result is transmitted to the concerned user through the agent IA which deals with request reformulation, with or without the intervention of the user.

![Fig.3 System architecture](image)

As a recapitulative, we can say that the agent IdA manages the simultaneous formulated users’ requests
decomposing them into a set of independent tasks. The decomposition process includes the identification of the requests’ similarities in order to formulate a set of autonomous and independent tasks which are waiting for responses at the same time. Each task represents a service which can be proposed by different TMN nodes, with different cost, processing time and data size. To response to tasks, needed data is available through the TMN and their collect correspond to the ICA agents job. Therefore, the SA agent must optimize the assignments of nodes to tasks, minimizing total cost and processing time, in order to respect due dates. To this assignment problem, we proposed a two-level optimization solution. The first level [6] aims to find a suitable number of ICA agents building their Workplans in order to explore the TMN completely. The second level [7] corresponds to the data flow optimization corresponding to the nodes selection increasing the number of satisfied users. In the next section, we present the cooperation model which deals with the data flow optimization.

4 Cooperation Model

When a user validates its request, the correspondent IA agent looks for an IdA agent available for receiving the formulated request. This is called a One Shot behaviour which is happen only when some event occur. In this paper, we focus on the behaviours of agents IdA and SA.

4.1 Identifier Agent (IdA)

4.1.1 Action

Each IdA agent provides the service "requests-reception" in order to allow IA agents to send their requests. Therefore, an available IdA agent receives a set of $\varepsilon$-simultaneous requests. This set corresponds to all requests formulated throughout a period of time $\varepsilon$, fixed by the system (the programmer). The set of tasks $\Gamma$, composing globally all composed requests, is regenerated during each short period of time $\varepsilon$, and then resumed to zero. Just before making a fresh start, $S'$ is identified from the database where all information providers are subscribed to describe each proposed service (processing time, cost and data size). Thus, the correspondent service table to $S'$, expressing the execution of each task belongs to $\Gamma$, on each identified server (see section 2), is deduced and sent to an available SA agent. This service table is sent with the request decomposition scheme (see section 2) which corresponds to the decomposition of each $\varepsilon$-simultaneous request into $\Gamma'$.

4.1.2 Behaviour

An IdA agent has two cyclic behaviours; the first one is resumed when a request is formulated and blocked otherwise. This kind of behaviour is interesting because it avoids CPU consuming when it is blocked; the agent is alive but it is not active. This behaviour is described as follows: each time an available IdA agent receives a request, it updates the set of tasks $\Gamma'$, composing a $\varepsilon$-simultaneous requests. $\Gamma'$ is reset to zero through a $\varepsilon$-cyclic behaviour which restarts indefinitely each $\varepsilon$ period of time without blocking; this corresponds to a ticker behaviour described as follows: when there is a waiting set $\Gamma'$ after an $\varepsilon$ period of time (the number of $\varepsilon$-simultaneous requests is not null), IdA agent identifies all servers which provide services for this $\Gamma'$, Therefore, it generates the correspondent service table which expresses the execution of each task belongs to $\Gamma'$ on each identified server. At the end of each $\varepsilon$-cycle, this service table and the request decomposition scheme are sent to SA agent just before their fresh start. Finally, the number of $\varepsilon$-simultaneous requests is also reset to zero.

4.2 Scheduler Agent (SA)

4.2.1 Action

This agent provides a service called "TasksServers-Sets-reception" allowing the reception of service tables and the correspondent request decomposition schemes from IdA agents. SA agent computes initial Workplans of ICA agents each time it is necessary (see 4.2.2) then deduces final ones by building a FeTAR instance from a FeTAR scheme. Generated information (final Workplans and FeTAR instance) is sent to each ICA agent in form of a contract.

4.2.2 Behaviour

A SA has two kinds of behaviours; the first one is a one shot behaviour which is lunched every time there is some variation on the network. Therefore, SA has to compute the new number of needed ICA agents to explore it totally, building their initial Workplans. The second behaviour is a cyclic one which allow IdA agent to send service tables and the correspondent request decomposition scheme.

5 Experimentation

We are developing our system, with JADE platform (Java Agent DEvelopment framework) [8]. JADE is a middleware which permits a flexible implementation of multi-agent systems; it offers an efficient transport of ACL (Agent Communication Language) messages for agents’ communications which complies with FIPA specifications [9]. JADE is written in java language,
supports mobility, evolves rapidly and until there, it is the only existent multi-agent platform which tolerates web services integration [10]. In this paper, we used a JADE graphical tool which sniffs message exchange between agents. This tool is useful to debug a conversation between agents. The final assignment solution of servers to tasks is deduced from initial Workplans generation and our genetic algorithm results (SA behaviour).

6 Conclusion and Prospects
To provide transport’s users pertinent real time information, we propose a multi-agent system composed of cooperative entities, using mobile agents. In this paper, we presented the internal structure of our proposed system and detailed behaviours of agents responsible of the data flow management. The communication through the system was implemented by the way of JADE which is an efficient multi-agent platform presented within the experimental section. As future work, we aim to detail behaviours of ICA agents adapted to a high incidence of network errors. Therefore, we aim to define a negotiation protocol between the agents ICA and agents SA which are responsible of the Workplans of mobile entities, in order to satisfy users in spite of the unavailability of some information providers.

References