MULTI-AGENTS SYSTEM BY GRID COMPUTING FOR REAL TIME TRAFFIC SYNCHRONIZATION

Li Weigang [1] (weigang@unb.br)

Marcos Vinicius Pinheiro Dib [2] (dib@bsb.politec.com.br)

Alba Cristina Magalhães Alves de Melo [1] (alves@unb.br)

Claudio Jorge Pinto Alves [3] (claudioj@ita.br)

[1] Department of Computer Science, University of Brasilia, Brasilia – DF, Brazil
[2] Politec Informática Ltda., Brasilia – DF, Brazil
[3] Division of Aronautic Infra-Structure, Instituto Tecnológico de Aeronáutica, Brazil.

Abstract

This study presents a new solution for the Air Traffic Flow Management (ATFM) problem. Its main objective is to improve computational efficiency for ATFM by combing grid computing with multi-agent coordination techniques. ATFM in Grid Computing (ATFMGC) was proposed to balance the communication among Multi-Agents, in ATFM for grid computing environments, in order to deal with synchronization problems. The function of the agents, their knowledge representation and inference processes were discussed, as well as the grid architecture, in which the agents execute it, was described to demonstrate the developed model. Also, a metric criterion called Standard of Balancing among Agents (SBA) was used as a basic index to measure the effectiveness of reducing the amount of communication among agents, and the delay of flights. The simulation results show the efficiency of the developed model, and its successful application in a case study.

1. INTRODUCTION

Air Traffic Flow Management (ATFM) is a problem example of Real Time Traffic Synchronization (RTTS) (Stoltz, Guerreau, 2002), (Stoltz, Ky, 2001). A common critical problem is to achieve computing efficiency due to the large-scale, safety and synchronization characteristics of most models and systems of ATFM. Since the 1970s, scientists from Artificial Intelligence, Operation Research and Air Transportation have worked together to develop more efficient Air Traffic Control (ATC) and Air Traffic Management (ATM) systems, but the computing-based solution still needs to be further enhanced in order to reduce aircraft delays.

Some Knowledge-Based Systems (KBS) have also been developed in ATC/ATM, such as the 4D-Planner; a ground based planning system using a rule-based approach for arrival sequencing and scheduling (Dippe, 1989), (Vôlckers, 1989). Gosling (1987) pointed out the potential of Artificial Intelligence application in ATC. A real time knowledge-based support for ATM has been developed by IBM Switzerland (Schlatter, 1994). In Brazil, an expert system for ATFM has also been evaluated to set timetable schedules and traffic flow control (Weigang, 1997), (Weigang, 1993).

A distributed ATM system has been studied in Australia (Tidhar et al, 1992). The advantages of that approach are its inherent distribution, autonomy, communication and reliability. Prevôt, from NASA Ames Research Center, has studied a distributed approach for operator interfaces and intelligent flight guidance, management and decision support (Prevôt, 2002). Eurocontrol (Nguyen-Duc et al, 2003) has also recently presented an application of multi-agent coordination techniques in ATM that sets up a methodological framework, using multi-agent coordination techniques, to support the collaborative work in ATM. Although the multi-agent coordination technique is a useful methodological framework, its research (Nguyen-Duc et al, 2003) is limited to a software shell. Considering the huge amount of information traffic in ATFM, its implementation may not be straightforward when brought into practical fields.

Recently, grid computing represents a prospective solution for large-scale computation tasks such as ATFM. A computational grid has been defined as "coordinated resource sharing, and problem solving, in dynamic, multi-institutional virtual organizations" (Berman et al, 2003 to Foster, C. Kesselman, 1999). A grid consists of hardware and software infrastructure,

providing dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. An agent-based resource management system, for grid computing, shows the importance of combining research in both grid computing and AI (Cao et al, 2002). Grid resource management has been defined as the process of identifying requirements, matching resources to applications, allocating those resources, as well as scheduling and monitoring Grid resources over time, in order to run Grid applications as efficiently as possible (Nabrzyski et al, 2003).

Based on these mentioned researches (Weigang et al, 1997 to Nguyen-Duc, 2003), we proposed in this study a multi-agent system in grid computing for ATFM: the ATFMGC (Air Traffic Flow Management in Grid Computing). The main contribution of this study is to develop an approach of cooperation and negotiation among agents, using grid computing in a real time traffic synchronization problem. It describes the synchronization concepts in ATFM, the main structure of the system, and the relation among its components. At the same time, the function of agents, their knowledge representations and inference processes are also discussed. Then, the ATFMGC was implemented as a prototype, in a network composed of three computers. A tactical planning case study, related with some real data from Brazilian airports, was illustrated as an example. A new metric criterion, called Standard of Balancing among Agents (SBA), was defined to evaluate the amount of communication among agents and the tolerated flight delays.

As a preliminary study, the investigation of using a multi-agent system and grid computing to ATFM is not only a simple application, but also an important topic in Artificial Intelligence, grid computing and Air Traffic Control. Using real data from four huge Brazilian airports, our approach was able to reduce the total of flight delays in air, from 20 minutes into 0, if compared to the traditional approach.

The following sections are organized in seven parts: after this introduction we placed section II presenting the basic concepts and characteristics of ATFM. Section III shows a general view of grid computing. Section IV discusses the work related in the area of artificial intelligence and distributed computing for ATFM. Section V presents the design of the ATFMGC, including the design of the agents, the inference process and the grid infrastructure. A case study is illustrated and discussed in section VI. Finally, section

VII presents the conclusions and future studies.

2. REAL TIME TRAFFIC SYNCHRONIZATION PROBLEM - ATFM

ATFM is developed to ensure an optimum flow of air traffic to or through areas, within which, the traffic demands may exceed at certain times the ATC system available capacity (Weigang, 1993). The general scenario of ATFM is shown in figure 1. ATFM includes four main functions:

a) Strategic Planning: deals with long-term planning, using typical projections of air traffic demands, in a long period of time;
b) Pre-Tactical and Tactical Planning: analyzes the total capacity available and the current demand. If congestion is detected, tactical operations are performed to solve it. If these operations occur at least one day before the occurrence of the detected congestion, it is used a pre-tactical planning operation. On the other hand, if the congestion is treated in the same day that it would occur, then tactical planning is used;
c) Short Term planning: to be used in case of ongoing flights: 45 minutes before a flight enters in a control sector, or 20 to 45 minutes before a flight arrival;
d) Monitoring and Control: also called an on-line operation; it is responsible to continuously monitoring the aircraft; from the moment it takes off to the moment it lands at the airport

As mentioned by Stoltz and Guerreau (2002), the ATFM system has two special properties, as follows:

a) Real time traffic synchronization: is an activity that consists in implementing corrective actions on traffic applicable, until traffic is actually received by the controllers.
b) Dynamic collaborative decision-making: aimed at achieving prompt dynamic "agreements" between Traffic Managers, co-involved in the implementation of corrective actions on traffic transit, performed from one sector to the other.

There are three types of ATC rules: for landing aircrafts, for both landing and taking-off aircrafts, and only for aircrafts in taking-off condition. In Brazil, the rules for landing aircrafts are presented as follows (DEPV, 1999):

a) A-Rule 1: If a second aircraft would arrive, at a time space between the estimated arrival time and the concluding landing time of another aircraft coming first, then this second aircraft will need to

be delayed;

b) A-Rule 2: If two aircrafts would arrive at the same time, the larger one will have landing priority;
c) A-Rule 3: If two aircrafts would arrive at the same time, the aircraft with passengers will have landing priority over the others;
d) A-Rule 4: If two aircrafts, of same category, would arrive at the same time, the one on schedule (following its previously scheduled arrival time) will have landing priority;
e) A-Rule 5: If two aircrafts, of same category and schedule, would arrive, the program will randomly choose which one will have landing priority;
f) A-Rule 6: If an aircraft is kept waiting for more than ten minutes (arrival rate is higher than runway capacity), the aircraft departure time, at the last airport along the route, will need to be rescheduled (adding its arrival delaying time).

3. GRID COMPUTING

The word grid was introduced to denote a new infrastructure for distributed computing. Grid Computing can be defined as "a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements" (Buyya, 2005).

Grid Computing technologies and concepts were initially developed in order to enable resource sharing between scientific institutions with common projects, who needed to share data, software and computational power. In developing applications for the grid, it is essential to have a unified middleware, to provide a transparent interface to the underlying protocols.

The Globus Toolkit (Foster, C. Kesselman, 1999) emerged in 1998 as an open source project, and quickly became a, de facto, standard for grid computing infrastructure. Globus defines and implements a set of protocols, APIs, and services, used by hundreds of grid applications all over the world. Moreover, it worked as a pioneer in the development of interoperable grid systems.

In 2002, the Open Grid Services Architecture (OGSA) (Nabrzyski et al, 2003) was introduced by the Global Grid Forum (GGF) to expand standardization. OGSA provided a new architecture for grid applications, based on Web Services, in order to achieve interoperability using industry standards.

Based on the concepts and technologies from both the Grid and the Web services

communities, OGSA defines a uniform semantic view of service (the grid service) and standard mechanisms for creating, naming, and discovering transient grid service instances. It also provides transparency of location and multiple protocol bindings for service instances, and supports integration with underlying native platform facilities. OGSA also defines mechanisms, concerning the Web Services Description Language (WSDL) interfaces, and associated conventions required for creating and composing sophisticated distributed systems, including lifetime management, change management, and notification. Service bindings can support reliable invocation, authentication, authorization, and delegation. Globus Toolkit 3 is an implementation of a part of the OGSA architecture, called the Open Grid Services Infrastructure (OGSI).

4. RELATED STUDY

The study proposed by Gosling (1987) is pioneer in showing how artificial intelligence can add benefits to the ATFM. It is in fact an expert system helping the controller to define flight scheduling.

Weigang et al. (Weigang, 1993) studied the Brazilian ATFM system, and developed a centralized knowledge-based system to re-schedule flight timetables, aiming to reduce conflicts.

ISAC is a centralized case-based reasoning to solve ATFM, proposed by Bonzano et al. (1996). Its goal is to help flight controllers to reduce conflicts.

Sekhavat and Sastry (1998) proposed an ATFM distributed system, by dividing the air space where planes fly, into virtual tubes. In this model each aircraft knows where it must be, in each and every moment, and an optimization function is applied to reduce its waiting time.

The OASIS system, proposed by Ljundberg and Lucas (2002), is one of the first systems using agents in ATFM. The system is divided into macro tasks, using separate agents to solve part of the problem. The agents are autonomous and cooperative, communicating with each other by asynchronous messages. Two classes of agents are defined: global agents, responsible to coordinate the aircrafts; and aircraft agents, solving problems that are specific to each aircraft. There is also a reasoning module, used in real time.

The system proposed by Callantine (2002), uses multiple intelligent agents as air traffic controllers, in a synchronous centralized architecture. A central controller sends a message to each agent, in every clock tick and, the agent should act when receiving the message, according to its contents and the internal condition of the agent.

The Virtual National Airspace Simulation (VNAS) (Mcdermott, 2003) is a simulation environment for ATFM, using a grid-based solution. The VNAS structure is composed of governmental and non-governmental research organisms, within the USA, connected by a computational grid infrastructure. Besides simulating ATFM scenarios, VNAS is also able to simulate aircraft modeling. When the resource becomes idle, the computing power is allocated to the VNAS through a specific screen-saver. Due to its characteristics, this is rather a peer-to-peer (P2P) architecture, if compared to the grid computing.

ARMS (Cao et al, 2002) is an agent-based system to schedule resources in a grid environment. In ARMS, each agent acts as a representative of a grid resource. The proposed agent architecture is hierarchical, and the agents cooperate to find out the available resources. This is a generic system and does not deal specifically with the ATFM.

Table I presents a comparative view of the systems discussed in this section.

5. THE ATFMGC DESIGN: A GRID-BASED MULTI-AGENT SYSTEM FOR ATFM

The ATFMGC system aims to reduce conflicts among flights, by analyzing the potential congestion points and solving them before they occur, as shown in this section. The scope is, thus, limited to the ATFM pre-tactical and tactical planning (section II).

In this system, see figure 1, we opted to use agents and grid computing to reduce conflicts in ATFM. As seen in table I, artificial intelligence has been successfully applied to solve a variety of ATFM problems. More recently, distributed architectures have also been used as an infrastructure to ATFM systems. However, the combination between agents and grid computing to ATFM has not yet been applied. As far as we know, this is the first grid-based multi-agent system for ATFM.

Table I - Related Studies in the areas of Artificial Intelligence and Distributed Systems applied to ATFM

System	Purpose	AI Technique	Architecture
Gosling (1987)	ATFM	expert system	centralized
Weigang (1993)	ATFM	knowledge-based	centralized
ISAC (Bonzano, 1996)	ATFM	case-based reasoning	centralized
Sekhavat (1998)	ATFM	mathematical modeling	distributed
OASIS (Ljungberg, 2002)	ATFM	agents	centralized
Callantine (2002)	ATFM	agents	centralized
VNAS (Buyya, 2005)	ATFM	mathematical modeling	P2P
ARMS (Mcdermott, 2003)	generic	agents	grid

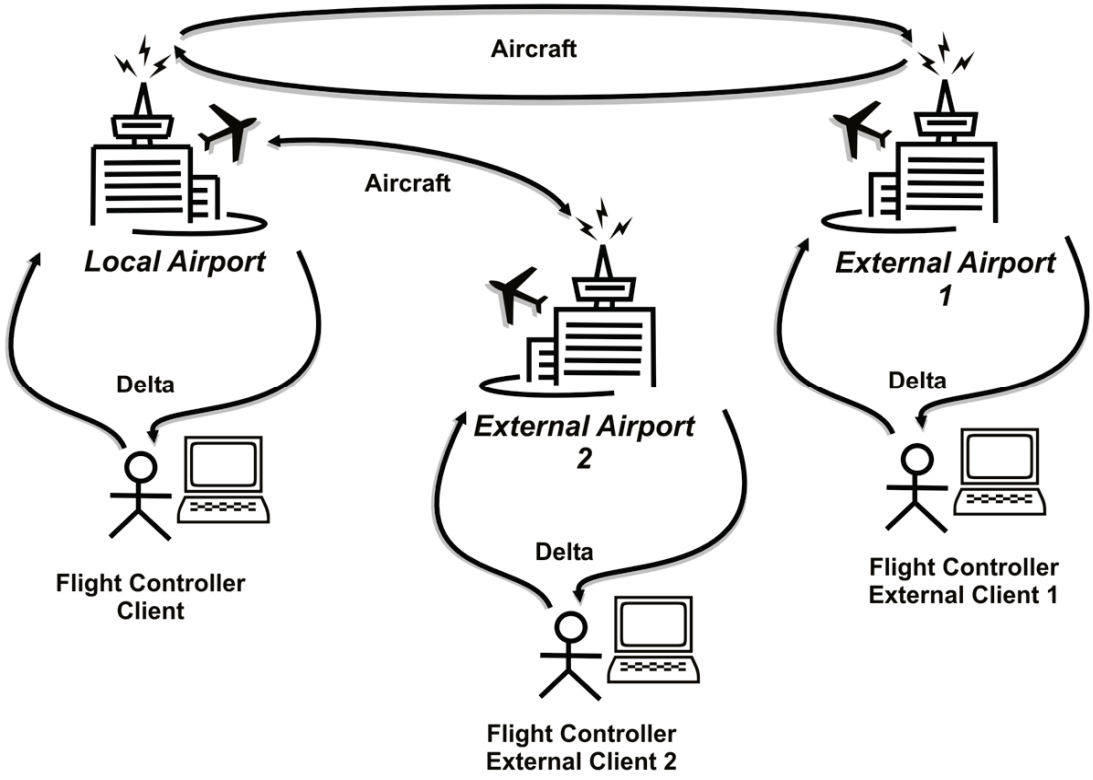


Fig. 1 - The ATFM system

The following constraints in airport terminal area are considered in the ATFMGC:

a) The arrival flights have absolute priority over departures;
b) Aircrafts are served on a FIFO basis (first in, first out);
c) No two aircrafts are permitted to be on the same runway at the same time;
d) Aircrafts under special conditions, have priority over others;
e) A taking-off aircraft may not be authorized if the subsequent arrival flight is closer than a specified distance, concerning that runway threshold;
f) Successive departures are kept apart for at least the time required for departure service rendering.

This proposed system, in short, analyzes the available capacity of departure, en-route, and arrival airspace, and takes tactical actions, in case the predicted demand exceeds its available capacity. There are four main functions in this system (Weigang, 1997), (Tidhar et al, 1992):

a) Creating a schedule for all departure and arrival flights, to every airport, trying to maximize the utilization of its runways and terminal;
b) Identifying congestion areas with regards of airspace, aircraft and terminal constraints;

c) Negotiating with other agents, over expected traffic flow, both to and from the airport;
d) Re-scheduling the flights, according to the negotiation outcomes, and communicating the new schedule to all agents involved in it.

5.1. Definition of the SBA

To measure traffic congestion, a metric criterion called SBA (Standard Balance among Agents) is used, which was adapted from the Weighted Combined Total Delay (WCTD) criterion (Tidhar et al, 1992). The SBA uses a different delay cost function, for flights landing and taking-off procedures. Considering that the objective of the ATFMGC is to minimize the air holdings of aircrafts, the function of delay cost is simplified in this study to multiply every minute of delay by a weight rank. For landing flights, weight ranks 5, and for taking-off flights, weight 1 is ranked.

When an airport is congested, some flights are delayed, and the extension of the delaying time is proportional to the severity of the congestion. But the flights do not have all the same importance, and the delay cost of a certain flight can be higher than the delay of others. Thus, the SBA is calculated according to the duration of delay (*minutes_delay*) and its weight rank (*weight_delay*), as shown in equation (1).

$$SBA(t1, t2) = \sum_{i=1,n} weight_delay(f_i) \times minutes_delay(f_i)/10$$

where: SBA (t1, t2): Standard of Balancing among Agents between instants *t1* and *t2*;
n: number of flights between instants *t1* and *t2*;
f_i: the flight under analysis

In ATFMGC, the system administrator can choose the SBA threshold value that fits better his/her system. A small value for this threshold will typically reduce the average flight delay, requiring more communication and negotiation among the agents. On the other hand, high threshold values will cause higher average flight delay, requiring less communication among the agents.

The calculation of SBA is thoroughly carried, in each of the airports. The schedule is created for one determined interval of time, TS (Time Slice), taking into consideration the exiting appointments, previously set, and the estimated arrival time of the flights. Time Slice is changeable and can be either reduced, when the airport load is high, or increased, when it is low.

5.2. Negotiation Process

Congestion is identified when the total number of flights, scheduled for an airport in a given time slice (TS), is higher than the airport maximum throughput. When this

occurs, flights are delayed or kept hold in air. In general, high levels of congestion cause high average flight delays.

In ATFMGC, congestion is defined in terms of SBA (section V.A). Whenever the airport's SBA is higher than a given threshold (called acceptable SBA), the airport is considered congested. When this situation is detected, a negotiation must take place.

Considering that the input flow of an airport is influenced by the output flow of other airports, the agents must interact, hours before the congestion would actually occur, in order to solve congestions. The agents of each airport negotiate by a metric criterion called average flight delay, which is derived from SBA. After reaching an agreement, each airport agent must recalculate its new SBAs, now considering the flight(s) that agreed to be delayed.

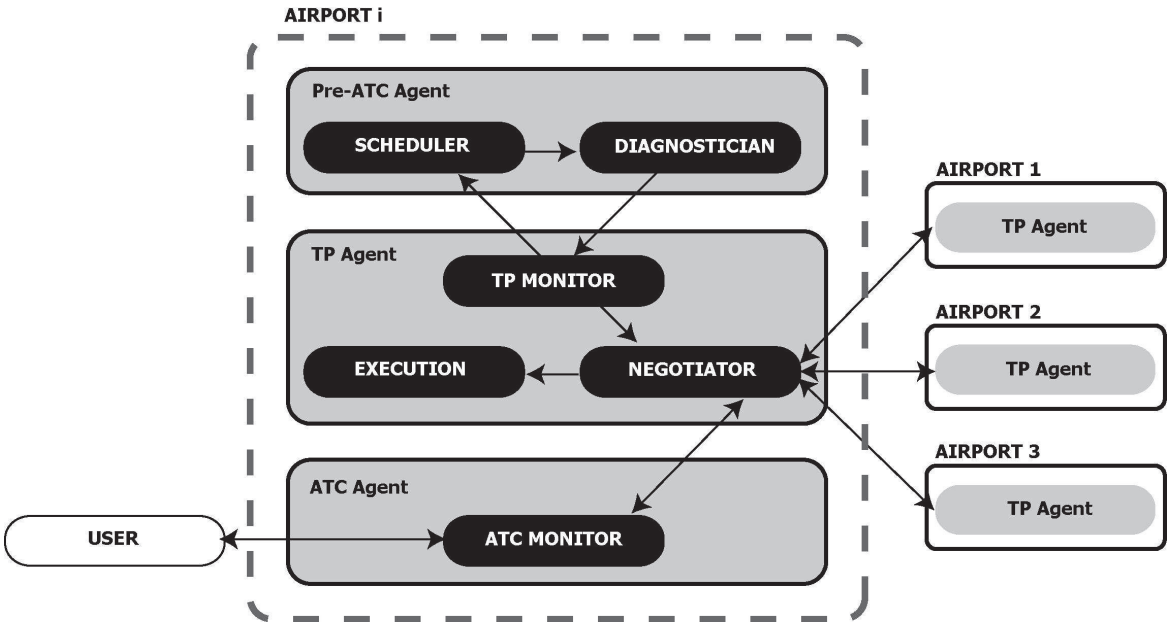


Fig. 2. Communication and Negotiation among Agents in ATFMGC

5.3. Design of the agents

We developed in this system, three types of agents: the Pre-ATC agent, the Tactical Planning (TP) agent and the ATC agent. The first agent is an expert agent for local air traffic control, designed only to simulate the situation for ATFM purposes. The second agent is a distributed Knowledge-Based System for Tactical Planning. Its goal is to manage traffic flow over the entire airspace. The ATC agent is responsible to manage the traffic flow, in and out the local airport, monitoring any possible changes and interacting with the external world. Figure 2 shows the interaction and negotiation among agents in the ATFMGC system.

5.3.1. The Pre-ATC Agent

The Pre-ATC agent of ATFMGC is represented according to the ATC/ATM rules, defined by the Brazilian Department of Air Space Control - DECEA (Weigang, 1997), (DEPV, 1999).

This agent has two modules: scheduler and diagnostician. The scheduler is responsible to re-schedule flights showing conflicting conditions. The diagnostician receives the re-scheduled flight timetable, from the scheduler, and calculates its SBA. With this SBA, the module will contact the TP agent, and negotiate with agents of other airports.

5.3.2. The Tactical Planning Agent (TP)

The Tactical Planning Agent is composed of three modules (figure 2): the TP monitor, the negotiator, and the executor.

The TP monitor is responsible for contacting the scheduler, in the Pre-ATC agent, to activate the re-scheduling of the flight timetable, providing the time slice (TS) to be used in the SBA computation. Also, it is responsible for receiving the re-scheduled flight(s) request(s) from the diagnostician and sending it/them to the negotiator.

The negotiator is the one that actually negotiates every new flight schedule. It receives the re-schedule request and interacts with the negotiators of other TP agents, until one acceptable SBA is reached.

After reaching an agreement, the negotiator contacts the executor, in order to change the data of the flights that were actually re-scheduled.

The TP agent, in knowledge representation, considers the following processes:

a) The schedule process proposes the best overload-removing solution, from its point of view. The distributed design of the system allows each airport to have its own local optimization criteria, for determining the schedule;
b) Diagnosis is designed to check any changes in the system (local and related airports), and re-schedule knowledge representation. The warning informing impossible solution is also a function of this process;
c) The Broadcasting and Negotiation process sends the proposed solution to other Tactical Planning agents, then, it receives the evaluation and validation from them; The negotiation is also a function of this process;
d) Evaluation and Validation is designed to evaluate and validate the proposed solution, received from other TP agents. The choosing solution is also a function of this process.

5.4. Inference Process

The main inference process used in the Pre-ATC agent was already described in literature (Weigang, 1997), (Weigang, 1993). The following steps are inference processes of TP agent:

a) The TP agent, at airport A, communicates with Pre-ATC agent, at airport B (and others), to get the flight departure delay requirements at airport A.
b) The schedule process of TP agent, at airport A, re-schedules the departure flights, according to the accepted WCTD value, at airport A.
c) The diagnosis process checks the new schedule and stores it in the database of airport A, if there is no conflict. When any conflict takes place, the schedule process repeats the procedure described in (a) and (b).
d) Broadcasting and Negotiation generate a message to be sent to the TP agents, at the related airports.
e) At the same time, the ATFMGC, at the related airports, is also proceeding with the changed schedule.
f) If the system detects conflicts at any related airport, a new schedule is generated at other airports, and the information is sent back to airport A. Broadcasting and Negotiation processes negotiate with the related airport.
g) Evaluation and Validation processes verify the actual schedule, within a certain time period.
h) The PT agent at airport A is recursively invoked until there are no more conflicts.

5.5. Grid Infrastructure

The ATFMGC is proposed to support a sharing and coordinated use of diverse ATFM resources, in an air traffic control system. As airports are geographically distributed, distinct controllers operate the ATFM components, some of the times with differing policies. The ATFMGC is developed in every airport as a virtual computing system, sufficiently integrated to deliver the desired quality of service, over all ATFM.

Using web methods, controller clients in ATFMGC can perform computations on their own airport data, and then, use the grid-computing infrastructure to interact with other controllers. The controller client does not directly connect to the "Final Service" that wants to use, but to an Interface of the web service. That way, the clients controller program is protected from the complexity of grid computing, to resolve the ATC/ATM conflicts, and insert the changes of flight schedules, at each airport. As shown in figure 3, all the clients communicate through web methods.

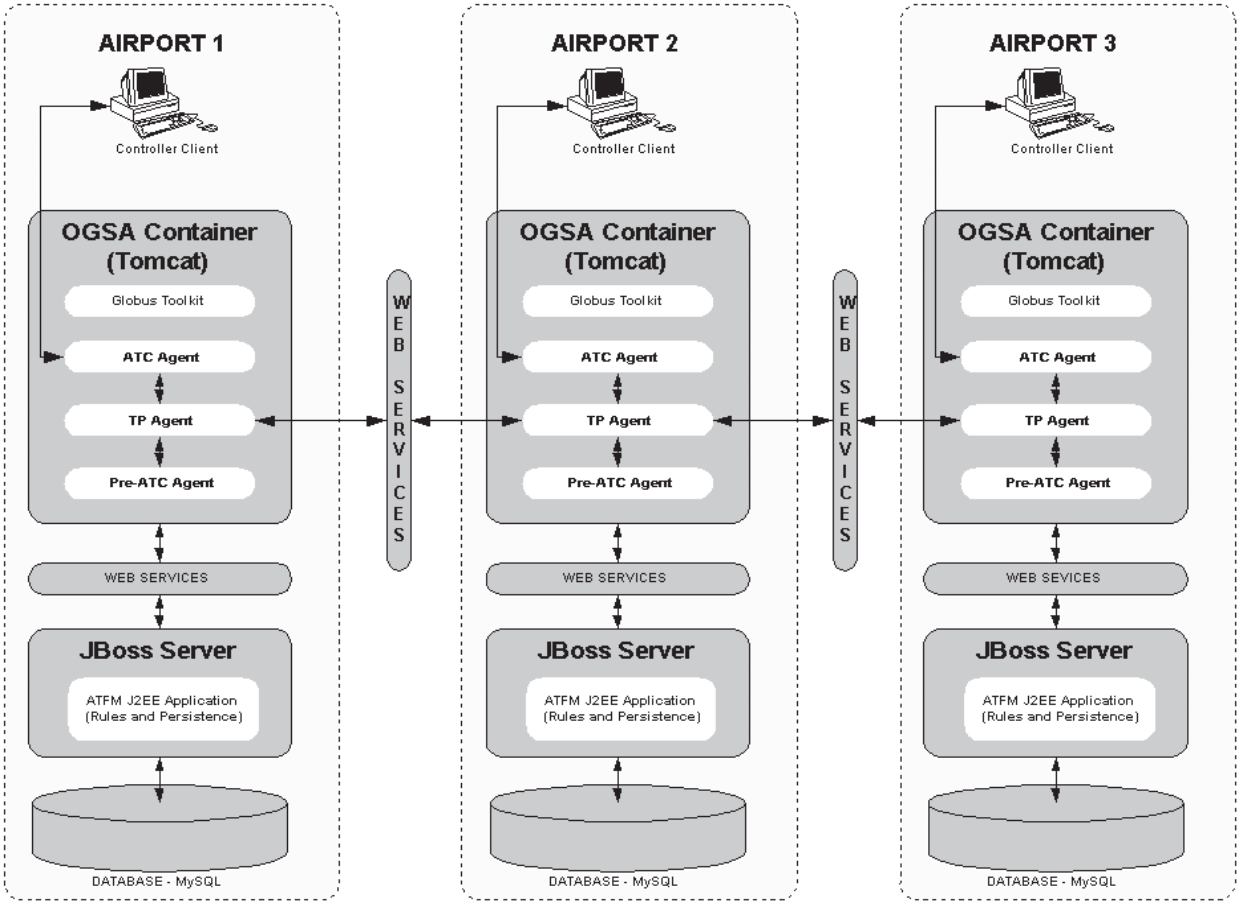


Fig. 3. The ATFMGC Architecture

In fact, the ATFMGC is applied on airports computing system by communicating accordingly with the "Final Services". With a simple interface in the client program, ATFMGC allows the users to get computing power in the grid.

The basic components of the ATFMGC architecture and the relation among the components are shown in Figure 3. The proposed ATFMGC architecture consists of five components: Interface, Open Grid Services Architecture (OGSA), Web Services, KBS and Database, distributed in a set of airports.

a) Interface: the function of the interface is to help the flight controller to use the ATFMGC, through Web Services. Assuming that each airport has its own operation system, the controller uses the interface function to send requirements and receive the answers from related airports, through some specific notification services of grid computing.

b) OGSA Globus Toolkit 3: provides the basic infrastructure to create the

synchronism service among involved airports.

c) Web Services: This is an essential component for the implementation of the ATFMGC. Since web services use standard XML languages, they are both platform and language independent. Since most Web Services use HTTP for transmitting messages (such as service request and response), to build an Internet-scale ATFM application represents a significant advantage. Web services address heterogeneous distributed computing data, by defining techniques for describing software components, methods for accessing these components, and discovering methods enabling the identification of relevant service providers. The neutrality of Web services concerning their programming language, model, network, and system software, is its key advantage.

d) Agents and Knowledge-Based System (KBS): KBS contains both domain knowledge (facts and rules) and the processes for structuring knowledge (knowledge representation), and it is distributed at each airport, in ATFMGC. The facts represent the local and real time updated information. The rules and processes for structuring knowledge are set almost equally, in every airport. Each referred airport has its own Knowledge Base, constituted of a variety of functional agents, such as Pre-ATC agent and Tactical Planning (TP) agent.

e) Databases: at each airport, databases consist of three parts: flight timetables, on-line traffic information and database administration. Due to the real time traffic synchronization property, database systems should be easily and quickly accessed. All data, such as aircraft and runway conditions, are stored in database by a XML form. The database is constructed through JDOM parser, an open source, tree-based, pure Java API.

In ATFMGC, the following parameters are considered (Nabrzyski et al, 2003):

a) A precedence directed acyclic graph (DAG), describing the control flow between ATC, Pre-ATC and Tactical Planning agents;

b) The processing time of each flight re-scheduling with the related airport. This may be a function defined by the kinds of resources on which this task can be performed, and by its input data;

c) A communication graph, listing the files to be transmitted between the re-scheduling tasks, as well as the sizes of the files;

d) Deadlines and/or ready times of flight re-scheduling with the related airports, if they are imposed;
e) Point-to-point characteristics (bandwidth, latency, etc.) of the network connections, between the nodes of a Grid (airports of ATFMGC); and
f) The characteristics of resources, on which particular re-scheduling tasks can be executed.

5.6. Implementation Details

At that point, an ATFMGC prototype was developed, using the Globus Toolkit 3. This prototype consists of three personal computers (Pentium IV). Each computer, with ATFMGC interface, represents the ATFM operation system of an airport. The whole system was programmed using JAVA.

The KBS, which consists of rules and facts, was developed using XML language. XML documents use the JDOM parser library for interpreting the stored data. Web Services and client interface were also integrated with the ATFMGC in JAVA. Apache TomCat 4.0 was used as a Servlet Container. For installing and manipulating the Globus Toolkit, Cygwin was chosen as emulator.

6. THE BRAZILIAN ATFM CASE STUDY

Based on the original data of flight schedules of 19, July 2004 (Guia de Horário de Nacionais e Internacionais, 2004), we analyzed the airport capacity and the air traffic flow of the main airports in Brazil (Guarulhos-GRU, Congonhas-CGH, Brasília-BSB, Galeão-GIG).

6.1. Air Traffic Flow in Four Airports

The air traffic flow and airport capacity of the four main Brazilian airports, are different. Through this evaluation we can observe the real traffic situation at GRU airport. We can also demonstrate the capacity of the ATFMGC to rationalize and create flight schedules, evaluating the minimum time space for separating flights concerning their landing and taking-off procedures. Table I shows the legend of some concepts, used in this section, and figure 4 shows the process of communication and negotiation in the ATFMGC simulation.

The capacity of GRU airport operation is 54 flights per hour (landing and taking-off procedures). Using 5 minutes as a Time Slice (TS) will mean that the capacity refers to 5 flights in every TS. The real schedule at this airport is 27 flights per hour, but the flights are not well distributed per TS. At some TS, more than 5 flights were scheduled. This is

the reason for the delay. Analyzing the flight schedule of GRU airport, we can observe that

a) The actual schedule is 6 delaying hours, total, in which 55.9% of them refers to taking-off procedures, with an average delay of 2.1 minutes per flight; and 31.52% of them are landing procedures, with delays of 1.5 minutes per flight.
b) There is a problem of actual schedule at 20h55, with 4 landing flights, and a 21h15 peak of delay, for flight taking-off procedures. If any meteorological problem occurs, from 20h50 to 21h10, heavy congestion may take place.

Table I - Legends OF THE CONCEPTS CODES

Code	Meaning
SAD	Inquiry for a possible re-schedule of a flight take-off – message sent by the airport in charge of a flight take-off, inquiring the destination airport the acceptance possibility for landing re-schedule of the referred flight.
PAP	Request of landing re-schedule – request of a flight landing re-schedule received by the destination airport of the referred flight.
SAP	Inquiry for a possible re-schedule of a flight landing – message sent by the airport in charge of a flight landing, inquiring the origin airport the acceptance possibility for taking-off re-schedule of the referred flight.
PAD	Request of take-off re-schedule – request of a flight take-off re-schedule received by the origin airport of the referred flight.
RPA	Acceptance of re-scheduling request – message sent by the airport requested of flight re-scheduling, agreeing with the referred PAP or PAD
HE	Time of the Event
HS	Requested Time of re-scheduling
ST	Status
A/N	Accepted / Not accepted

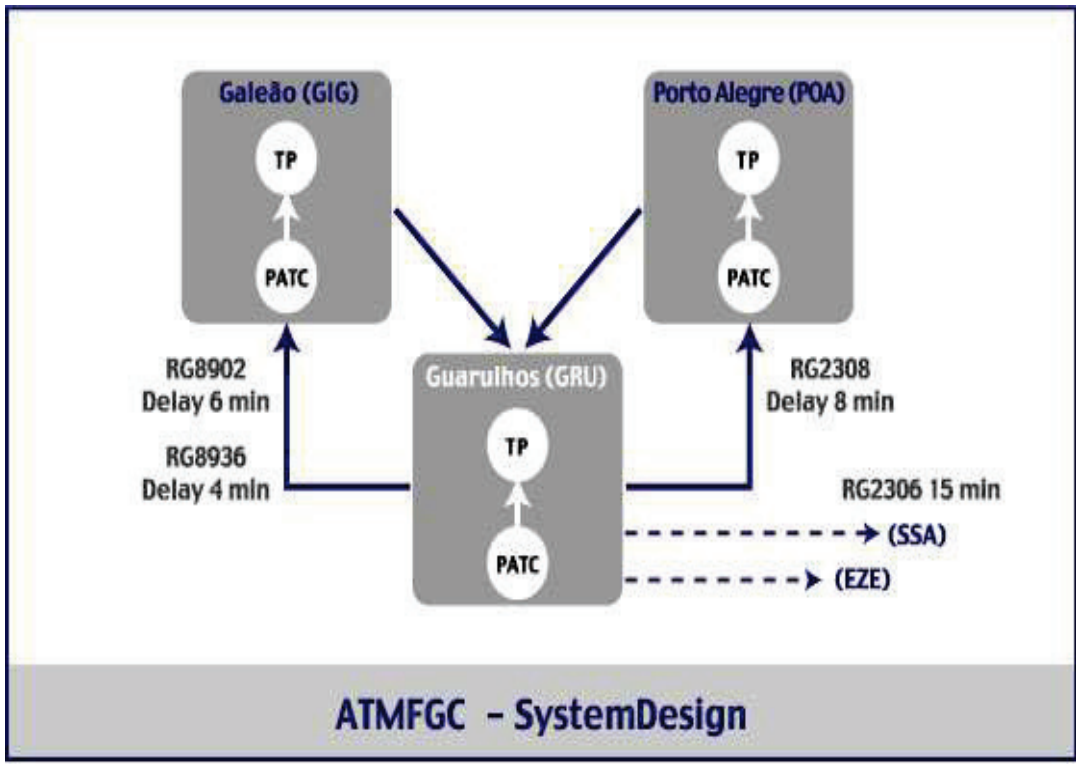


Fig. 4. The Communication and Negotiation Process in the ATFMGC Simulation.

6.2. Simulation

The first task, for every TP agent, is to get a solution for conflicts and inform the problems to other airports, within the Grid. The TP agents can communicate to each other to reduce delays, using the SBA (section V.A.2) to measure the delaying situation in every airport, and generate a standard balancing.

SBA is locally calculated, for each airport, at each TS, and will indicate traffic and congestion. With a high SBA value, the congestion situation represents a more complicated coming situation. There is a local acceptable SBA, based on the traffic capacity of each airport. If the real SBA is higher than the local acceptable SBA, the system needs to communicate with the other airports, to negotiate a suitable solution. It is necessary to establish an acceptable SBA to unify the interpretation of the ATFMGC. Five accepted SBA cases were considered in the simulation: 0, 0.75, 1, 1.25 and 2.5. Table II shows the result of the simulation.

a) The Accepted SBA is 0. In this case, the system does not allow any delay (even if it is impossible in a real case). For any delay, the system needs to communicate and negotiate among four airports. In this case, 421 communications took place, totally within the system. Just at CGH, the number of negotiation rounds was 218, and it took 8

minutes to eliminate any delay at this airport (tables II and III). Besides, 133 messages were required to eliminate take-off delays of the flights, and 53 messages for landing procedures. The system processed 9 messages asking to re-arrange landing procedures and 7 to re-arrange taking-off procedures. Finally, 16 messages answered these requirements.

Table II - Number of Messages Sent to Each Airport According to the Accepted SBA

SBA	Airport	SAD	SAP	PAP	PAD	RPA	Sub Total	Total
0	GRU	28	1	17	2	19	67	421
	CGH	133	53	9	7	16	218	
	BSB	64	23	6	4	10	107	
	GIG	9	2	8	1	9	29	
0.75	GRU	11	1	1	1	2	16	98
	CGH	21	5	6	1	7	40	
	BSB	18	9	1	1	2	31	
	GIG	7	2	0	1	1	11	
1	GRU	0	1	1	0	1	3	33
	CGH	0	0	6	1	7	14	
	BSB	5	1	1	1	2	10	
	GIG	3	1	0	1	1	6	
1.25	GRU	0	1	0	0	0	1	21
	CGH	0	0	4	1	5	10	
	BSB	3	1	1	0	1	6	
	GIG	2	0	0	1	1	4	
2.5	GRU	0	0	0	0	0	0	17
	CGH	4	1	2	0	2	9	
	BSB	3	2	0	0	0	5	
	GIG	2	1	0	0	0	3	

b) Accepted SBA is 1. In this case, the system allowed a reasonable delay. For any delay, the system needed to communicate and negotiate among four airports. In this case, 33 communications took place, within the system. At CGH, the number of negotiation rounds was 14 and it took one and half minutes to eliminate any delay at this airport (table III).

c) *Accepted SBA is 2.5*. This means that the system allowed considerable delay. In this case, just 17 communications took place. At CGH, there were 9 negotiation rounds, and it took 20 seconds to eliminate any delay at this airport (see table III).

Table III - Execution Times for ATFMGC WITH SBAS 0, 0.75, 1, 1.25 and 2.5

Accepted SBA	GRU	CGH	BSB	GIG
0	6:53.408	8:07.354	8:24.220	7:27.628
0.75	0:47.715	1:37.587	0:53.843	0:13.329
1	0:11.896	1:21.377	0:45.431	0:09.794
1.25	0	0:48.478	0:27.106	0:06.800
2.5	0	0:19.453	0:04.366	0:00.360

6.3. A Proposal for Traffic Management

The ATFMGC was designed to analyze conflicts, 2 hours ahead of the operation real time. In case there will be a conflict between two landing flights in B airport, one or the two of them may not have taken off yet. The system will arrange a delay, to one of the taking-off flights at A airport, equivalent to the time this flight would have to wait for landing at B airport. When this aircraft takes off, its arrival will be synchronized with its landing time window. As this is not in real time, it is assumed that the flying time is always constant.

As the landing flights have absolute priority over taking-off ones, these flights tend to be delayed. However, since this flight will have its take-off delayed at the original airport, the

ATFMGC will negotiate a new landing schedule with the destination airport. The agents of the ATFMGC will synchronize all the flights, of all the airports, within the Grid. The proposed method considers the synchronization of the flights, as well as the behavior analysis of the Tactical Planning and Pre-ATC agents. The advantage of the flights synchronization process is to reduce flights in air, consequently, reducing the possibility of congestions.

Table IV - Flight Delay comparison using the Two Methods (in minutes)

Flight	From	To	Delay in Air Calculated by the Method:	
			In use	Proposed
JJ3540	CGH	BSB	1	0
JJ3716	CGH	BSB	0	0
RG8920	GIG	GRU	2	0
VP4261	GIG	GRU	2	0
RG2329	BSB	GRU	1	0
G31712	GIG	CGH	5	0
RG2635	BSB	CGH	7	0
VP4281	BSB	GIG	2	0
Total			20	0

Table IV compares flight delays using two methods: the ATFMGC and the conventional method. For the conventional method, the conflicts of flights show a delay in air of 20 minutes, total. This means an average of 1.7 minutes, per flight, during one hour. Using the ATFMGC method, with the synchronization of schedules for flight landing and taking-off procedures, there are no more delays in air. Considering security factors, saving 20 minutes in one hour is significant in air delaying.

CONCLUSIONS AND FUTURE STUDY

A multi-agent system, using grid computing to deal with real time air traffic synchronization problems, was proposed in this research. Through negotiation and synchronization of flight schedules, among the airports within the Grid, a lack-of-conflict situation is reached. In the strategically planning phase, the ATFMGC can assist air traffic flow control, in planning and scheduling the flights, allowing their identification and reducing overload and conflicts. Its structural components allow verifying and synchronizing, in real time, flight taking-off and landing schedules, in all airports, throughout their routes, observing the capacity of each airport. Standard of Balancing among Agents (SBA) was conceived to balance the communication among the airports; however, it may be used as well, for solving similar problems in multi-agent system.

The ATFM is an interesting domain for multi-agent system application. As far as we know, a study about multi-agent system in real time air traffic synchronization problems, using grid computing, has not been reported yet. Bearing in mind the advantages of grid computing, this proposal also presents a

solution for air transportation. The result obtained in this study may represent a significant contribution to the research of artificial intelligence, grid computing and air traffic transportation (Cao et al, 2002), (Weiss, 2002). The following aspects may be deeply evaluated in further studies:

a) Using multi-agent coordination techniques to guide the development and implantation of the grid computing platform;
b) Building a special grid-computing environment for ATFM, to study its main components (both hardware and software), and the relation of computing and air traffic control.

REFERENCES

Berman, F., Fox, G., Hey, T., *Grid Computing: Making The Global Infrastructure a Reality*, John Wiley & Sons, 2003.

Bonzano, A., Cunningham, P., Meckiff C., ISAC-A CBR System for Decision Support in Air Traffic Control, *Advances in Case-Based Reasoning*, Springer-Verlag, 1996.

Buyya, R., *Grid Computing Info Center (GRID Inforware)*,http://www.gridcomputing.com, accessed in 2005.

Callantine, T. J., *CATS-Based Air Traffic Controller Agents*, NASA Contractor Report 2002-211856, NASA Ames Research Center, 2002.

Cao, J., Jarvis, S. A., Saini, S., Kerbyson, D. J., Nudd, G. R.. *ARMS: an Agent-based Resource Management System for Grid Computing*, Scientific Programming, vol. 10, 135-148, 2002.

DEPV, IMA 100-12, portaria DEPV No. 46 de 30/06/99, Departamento de Eletrônica e Proteção ao Vôo – DEPV, Ministério de Aeronáutica, 1999.

Dippe, D., *4D-Planner – A Ground Based Planning System for Time Accurate Approach Guidance*, DLR- Mitt, 89-23, 1989.

Ferreira, L., Berstis, V., Armstrong, J., Kendzierski, M., Neukoetter, A., Takagi, M., Bing-Wo, R., Amir A., Murakawa R., Hernandez, O., Magowan, J., Bieberstein, N., 2003. *Introduction to Grid Computing with Globus*, IBM, 2003.

Foster, I., Kesselman, C., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publisher, 1999.

Gosling, G. D., *Application of Artificial Intelligence Application in Air Traffic Control*, Transportation Research, 21A(1), 1987.

Guia de Horário de Nacionais e Internacionais, Panrotas Editora Ltda, 2004, No. 370.

Ljungberg, M., A. Lucas, *The OASIS Air Traffic Management*. Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence (PRICAI), 1992.

Mcdermott, W., Maluf, D., Gawdiak, Y., Tran, P., *Airport Simulations using Distributed Computational Resources*, The Journal of Defense Software Engineering, 2003.

Nabrzyski J., Schopf, J. M., Werglaz, J., *Grid Resource Management: State of the Art and Future Trends*, Kluwer Academic Publishers, 2003.

Nguyen-Duc, M., Briot, J. P., Drogoul, A., Duong, V.. *An application of Multi-Agent Coordination Techniques in Air Traffic Management*, Proc. of the IEEE Int. Conference on Intelligent Agent Technology, 2003.

Portella, G. J., Melo, A. C. M. A., *A Load Balancing Strategy to Schedule Independent Tasks in a Grid Environment*, IX Workshop de Sistemas Distribuidos y Paralelismo Arica, Chile, 2005.

Prevôt, T., *Exploring the Many Perspectives of Distributed Air Traffic Management: The Multi Aircraft Control System MACS*, Proc. of the Int. Conference on Human-Computer Interaction in Aeronautics S. Chatty, J. Hansman, and G. Boy (Eds.), 149-154, 2002.

Schlatter, U., *Real Time Knowledge Based Support for Air Traffic Flow Management* , IEEE Expert, 21-24, 1994.

Sekhavat, S., Sastry, S., *A Distributed Automait Air Traffic Management System*, Proceedings of the International Symposium on Robotics and Automation, Saltillo, 1998.

Sotomayor, B., *The Globus Toolkit 3 Programmer's Tutorial*, 2003.

Stoltz, S., Guerreau, R., 2002. *EUROCONTROL: Future ATFM Measures (FAM) operational Concept*, EEC Note No. 13/02, 2002.

Stoltz, S., Ky, P., *Reducing Traffic Bunching More Flexible Air Traffic Flow Management*, Proc. of the 4th USA/Europe ATM R&D Seminar, New Mexico, 2001.

Tidhar, G., Rao, A., Ljunberg, M., 1992. *Distributed Air Traffic Management System*, Tech. Rep. 25, Australian Artificial Intelligence Institute, January, 1992

Vôlckers, U., *Approach Towards a Future Integrated Airport Surface Traffic Management*, DLR- Mitt, 89-23, 1989.

Weigang, L., Alves, C. J. P., Omar, N., *Knowledge - Based System for Air Traffic Flow Management: Timetable Rescheduling and Centralized Flow control*, Applications of AI in Engineering VIII, Vol. 2, G. Rzevski, J. Pastor, and R. A. Adey (Eds.), Elsevier, 655-670, 1993

Weigang, L., Alves, C. J. P., Omar, N.,1997. *An expert system for Air Traffic Flow Management*. Journal of Advanced Transportation", Vol. 31, No. 3, 343-361, 1997.

Weiss, G., *Multiagent systems*, The MIT Press, 2002.