An application of Multi-Agent Coordination Techniques in Air Traffic Management

Minh Nguyen-Duc, Jean-Pierre Briot, Alexis Drogoul
Laboratory for Computer Science (LIP6), University of Paris 6, Paris, France
{minh.nguyen-duc, jean-pierre.briot}@lip6.fr

Vu Duong
EUROCONTROL Experimental Center, Brétigny, France
vu.duong@eurocontrol.int

Abstract

Air Traffic Management (ATM) involves collaborative work from several actors: traffic flow managers, air traffic controllers (controllers) and pilots. The ever-increasing demand for commercial air travel poses great challenges to today's airspace-centered ATM system in which each controller only undertakes the responsibility to control the aircraft flying through her/his own airspace (sector). Any aircraft bunching occurring in a sector has the potential to cause the risk of instant traffic overload in another sector. It is therefore essential to further decentralize the system by redistributing the responsibility as well as workload. This paper presents our research to support this redistribution by setting up a methodological framework using multi-agent coordination techniques. A recently identified problem, i.e. Real-time Traffic Synchronization, is chosen as the first application

1. Introduction

Current ATM system is airspace-based. The airspace is divided in several sectors, the size of which depends on the number of aircraft in the region and the geometry of air routes. There are usually two air traffic controllers to handle the traffic in each air sector: a planning controller and an executing controller. The planning controller works at a strategic level to minimize the number of conflicts or their complexities. The executing controller works at a tactical level to ensure that there are no conflicts *i.e.*, infringements of standard separation, between aircraft. When many aircraft occupy the same region, the size of a typical sector can shrink to a minimum This incurs several problems. The percentage of time on "handling over" aircraft from one sector to the other increases drastically so that the remainingtime for traffic monitoring decreases to a minimum, which adds complexity and uncertainty in the process. The time and spatial constraints make it difficult to ensure safe traffic, and allows very little flexibility.

For long time, controllers have been assigned with responsibility and control authority to individual sectors. Although there are exceptions, generally speaking, they only interact with controllers of other sectors at the moment of inter-sector control transfer. The advantage to this operational paradigm is that in the case of an operational error (*e.g.* conflict between aircraft), the fault is readily determined. Yet, the key shortcoming, which especially emerges in the age of air traffic explosion, is that there is no impetus for controllers to collaborate on other operations than the inter-sector control transfer; and so conflicting actions can occur when controllers of different sectors work on the same traffic flow, or more particularly, control the same flight.

Nowadays, several investigations [3][5] in the ATM domain try to establish new operational concepts that allow the controllers in different sectors to work more together. Their objective is not only to rationalize the controllers' collective behaviors, but also to distribute steadily their responsibility and workload through sectors. For instance, the controllers (two controllers) of a congested sector can be ensured that traffic flows arriving at this sector are smoothened by controllers of the previous neighboring sectors. These studies all try to define new coordination operations for the classical actors of the ATM system: traffic flow managers, controllers and pilots. However, they still need in addition methodological framework that provide tools which treat essential aspects of coordination.

To address the "key shortcoming" mentioned above of the current ATM system, we intend to establish such a methodological framework based on multi-agent coordination techniques. But why "agents?" The first reason comes from the simplicity and coherence of their collective behaviors. Although the human coordination is really much more complex, many ATM guidance papers are also edited with the main objective to clearly define or standardize the operational procedures. And if a simple technique proves its effectiveness and security, it will be preferred to apply rather than more complex ones. The second reason is that the promising research results on multi-agent coordination are attractive. State-of-the-art of this filed has been fundamentals of the action coordination: distributed planning, common intention, collective reaction.

Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03)
0-7695-1931-8/03 $ 17.00 © 2003 IEEE

2. Research approach

Our general objective mentioned above was to set up a methodological framework using multi-agent coordination techniques that supports the collaborative work in ATM. Four top-level sub-objectives of our research are:

- Discover potential applications of multi-agent coordination techniques in ATM,
- Model a recently identified collaborative problem, *Real-time Traffic Synchronization*, by using a well-known generic coordination model, STEAM [6],
- Based on the experimental analysis of this modeling, continue with other techniques of coordination,
- And finally try to apply a wide range of coordination techniques to the ATM system.

3. Modeling tool - STEAM

STEAM (Shell for **TEAM**work) [4][6] is a generic model of *Teamwork* , which is based on *joint intention* theory [2] but also borrows some ideas from the *shared plan* theory [1]. This model is based upon the execution of hierarchical reactive plans among which *team plans* are distinguished from *individual plans* (agents always form together teams to perform collective activities.) When an agent selects a team plan for execution, a team's *common intention* is instantiated. Team plans explicitly express team's common activities, as opposed to the regular individual plans which express an agent's own activities.

When an agent $\boldsymbol{r_i}$ invokes a team plan for execution, the plan is annotated with an executing agent, which may be dynamically determined at execution time to be an individual, or a sub-team, or the team. In the hierarchy, each plan $\boldsymbol{p_j}$ no-leaf is a set of the child plans whose executing agents are members/sub-teams of that of $\boldsymbol{p_j}$. There is a common intention associated with each (team) plan.

To apply team plans, a STEAM-based agent maintains a team state that is usually initialized with information about the team, such as the members in the team, the possible sub-teams, the available communication channels, the pre-determined team leader and so forth. STEAM can also maintain sub-team states for sub-team participation. One key restriction is imposed in order to preserve the consistency of a (sub-)team state, *i.e.*, only the team plans representing that (sub-)team's common intentions can modify it.

The key to the team plan execution is a *persistent weak achievement goal (**PWAG**)*. ***PWAG*** $\boldsymbol{(v_i, OP, \Theta)}$ denotes commitment of a member $\boldsymbol{v_i}$ (of a team $\boldsymbol{\Theta}$) to its team task ***OP*** prior to the team's establishing a *common persistent goal* (***CPG***). To execute a team plan, agents must first establish it as a common intention by means of the *commitment protocol* described below:

- Team leader broadcasts a message to the team to establish ***CPG*** to $\boldsymbol{\Theta}$ operator ***OP***. Leader now establishes ***PWAG(leader, OP,*** $\boldsymbol{\Theta_1}$***)***. If ***CPG(***$\boldsymbol{\Theta}$***, OP)*** not established within time limit, repeat broadcast.
- Subordinates $\boldsymbol{v_i}$ in the team wait until they receive leader's message. Then, turn by turn, broadcast to establishment of ***PWAG(***$\boldsymbol{v_i}$***, OP,*** $\boldsymbol{\Theta_1}$***)*** for ***OP***; and establish ***PWAG(***$\boldsymbol{v_i}$***, OP,*** $\boldsymbol{\Theta_1}$***)***.
- Wait until $\boldsymbol{\forall v_i}$, $\boldsymbol{v_i}$ establish ***PWAG(***$\boldsymbol{v_i}$***, OP,*** $\boldsymbol{\Theta_1}$***)*** for ***OP***; establish ***CPG(***$\boldsymbol{v_i}$***, OP,*** $\boldsymbol{\Theta_1}$***)***.

STEAM also involves *monitoring* and *replanning* capabilities. It forms a *common intention* to replan whenever a team's *common intention* for a execution step is seen to be unachievable. Moreover, to avoid significant communication overhead, STEAM integrates *decision-theoretic communication selectivity.*

4. Real-time Traffic Synchronization

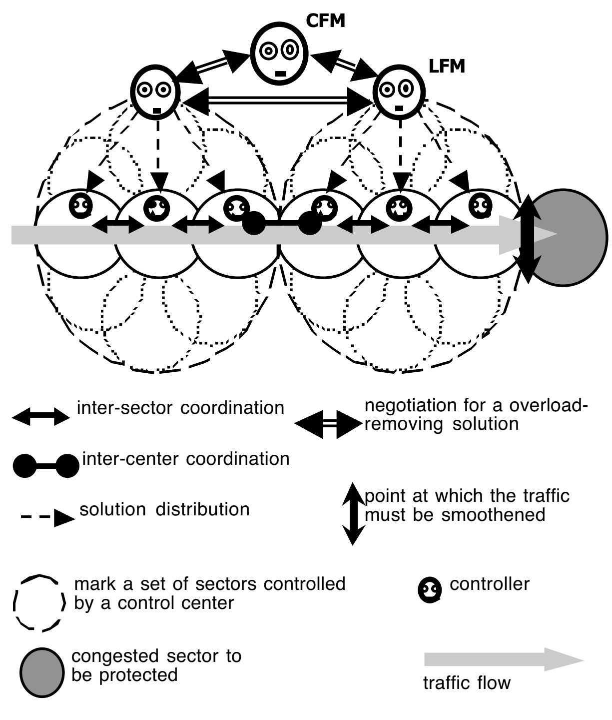


Figure 1. ***Real-time Traffic Synchronization***

The investigation on the concept of *Real-time Traffic Synchronization* is conducted by Stoltz et al. [5]. Its principal conceptual elements are the followings:

- *Bunching effect*: A major concern in leaving some loose end to ATM rules is the occurence of uncontrolled traffic peaks at the entry of a congested area. This phenomenon, often caused by some aircraft "in bunch", is known in the operational world as "traffic bunching" effect.
- *Real-time traffic synchronization*: A way to solve the problem could be to provide the ATM system with

means to correct or re-adjust the status of traffic flows with respect to the actual drifts. Such procedures should be locally adaptable and reactive solutions able to structure and organize the flows of arrival in real-time. Such techniques could be used for the readjustment of the times of arrival at a congested point, thus enabling to de-bunch problematical delivery. Provided a risk of bunching identified in a sector, these techniques should enable several controllers to "work" together on the traffic and "smooth" the bunching peaks before they affect the congested area.

This is in fact a collaborative work of some *actors*:

- The *Air Traffic Controller* (controller) controls the aircraft flying through her/his sector.
- The *Local Flow Manager* (LFM) manages the aircraft flow passing through her/his airspace zone. In each control centre, there is a LFM and several controllers managing together an airspace zone composed of several sectors.
- The *Central Flow Manager* (CFM) supervises inter-centre operations.

5. An illustrative example

This section concerns the outlines of application of STEAM to the *Real-time Traffic Synchronization*. We have to desribe this operational concept in the form of "teamwork" before directly using STEAM.

5.1. Operational description

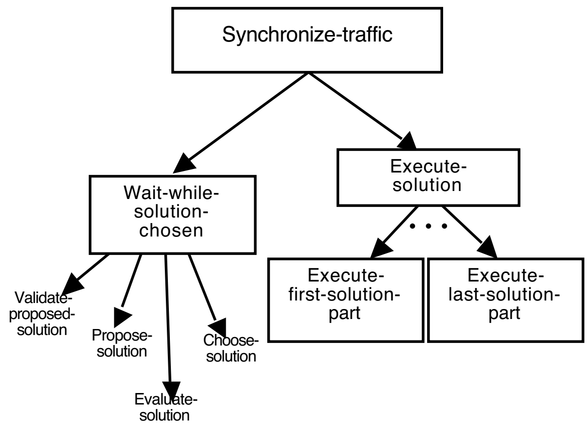
First of all, we define general team-related operations for the *Real-time Traffic Synchronization*:

- *Team formation*: A LFM identifies a risk of an overload caused by some aircraft in bunch, and warns the CFM and all concerned LFMs of this risk. Each LFM in turn warns all her/his subordinate controllers. Thus a global team composed by the CFM, LFMs and controllers is formed in order to remove the overload. Each team member knows that she/he is participating in the global collective activity and believes that these others will perform their assigned tasks.
- *Sub-team formation*: Once the global overload-removing team has been formed, each actor knows immediately the sub-teams in which she/he participates. There are three kinds of sub-team: *Solution choosing team* (CFM and LFMs), *Solution executing team* (LFMs and controllers), *Local solution executing team* (a LFM and subordinated controllers).
- *Solution proposition*: As member of a *Solution choosing team*, each LFM proposes the best overload-removing solution from her/his point of view, then broadcasts this solution to the other team members (LFMs). This operation requires a deep assessment of the current traffic state in the local airspace zone, and a pre-defined set of adjustable overload-removing solutions.
- *Evaluation of proposed solution*: As member of a *Solution choosing team*, once receiving an broadcasted solution proposed by another LFM, the given LFM evaluates the efficacy and feasibility of this solution corresponding to the current traffic state, then broadcasts the evaluation result.
- *Validation of proposed solution*: As member of a *Solution choosing team*, CFM validates all proposed solutions by examining their side effects.
- *Choosing solution*: Based on the LFM's solution evaluations and the CFM's validations, one of LFMs choose a solution to execute, broadcasts it to all members of the global team. It is her/him who plays the role of referee in the corresponding *Solution choosing team*.
- *Warning of impossible solution*: While negotiating for an overload-removing solution, if any LFM in a *Solution choosing team* recognizes that her/his subordinated controllers cannot execute one of partially pre-defined solutions, she/her must inform all members of the *Solution choosing team*. This warning makes all LFMs neglect the impossible solution and restarts the solution choosing process.
- *Executing chosen solution*: Each *Local solution executing team*, composed by a LFM and her/his subordinated controllers, execute a part of the chosen solution; this execution can be considered not only as a sub-plan of the global execution plan but also as a common plan of the LFM and concerned controllers in the same control center.

5.2. STEAM-based modeling outline

We use STEAM now to model team-related tasks described above. CFM, LFM and controller are all modeled as agents. An incomplete resulting plan hierarchy is represented in the figure 2.

In fact, STEAM provides the mechanism to create a team plan for each team formed. An example of team plan is the one named *Wait-while-solution-chosen*. First of all, the persistent weak achievement goal of a member (an agent) of the *Solution choosing* team to this plan is established when it commits to the team's task for waiting while the solution to execute is being chosen but doesn't know if the other members commit to this task. When the commitment of the others is ensured, it establishes the common persistent goal for the *Wait-while-solution-chosen* plan. (The detailed formal protocol is presented in Section 3.)



* framed plans are team plans

Figure 2. **Plan hierarchy for *Real-time Traffic Synchronization***

Note that the execution of the *Execute-solution* plan has to wait for the termination of the execution of this plan. When the referee (LFM agent) decides to choose a solution, it executes a communication plan for updating the team belief. A message is sent to all members of the *Executing solution team* and the result of the solution choosing process is attached to this message, which can be considered for establishing the common persistent goal of the *Execute-solution* plan.

The *Warning of impossible solution* can be moreover well modeled by the use of the *monitoring* and *replanning* provided by STEAM.

6. Discussion

Human factor researches in the ATM domain show that one of the most important causes of the controller's stress is uncertain actions of the controller himself, of the other controllers and pilots. During some short time, the action confirmations are always not adequate, the actors in the ATM system must thus be implicitly aware of the actions of the others. In this context, STEAM could provide a wide range of methodological tools to rapidly establish explicit *common intentions* and *commitments*. Intuitively the explicitness reveals the complexity and waste of time. We argue contrary that STEAM could support explicit real-time collective activity by providing automatable tasks.

In fact, STEAM adds some generic tasks, *e.g. commitment protocol*, to the LFM and controller's work. But the problem is if such additional tasks can be automated or not. We believe that those applied to agents are inherently automatable. However this requirement must be validated by experimentation and also by realistic implementation. In the example described above, generic communication messages not only can be integrated in real messages between LFMs but also automatically transmitted by software/hardware tools. For instance, it is easy to recognize that all tasks in the *commitment protocol* are automatable.

7. Conclusion

The work presented investigates the application of multi-agent coordination techniques to a recently identified problem in Air Traffic Management (ATM), *e.i. Real-time Traffic Synchronization*. We believe that simplicity and automatability of agent's collective behaviors can offer efficient solutions to this collaborative problem. Recent successes of the multi-agent coordination techniques are encouraging. At this stage of investigation, realistic operational scenarios and their modeling using some generic practical coordination models, *e.g.* STEAM, should be examined. The first analytic results on the STEAM-based model of *Real-time Traffic Synchronization* were promising.

The upcoming step will therefore be a "quantitative" experimentation with the objective of evaluating expected gains in a more realistic view. Moreover it is clear that the rapidity and flexibility (reactivity against incidents) of the overall system could not be really evaluated without human-in-the-loop experimentations.

8. References

[1] B. Grosz & S. Kraus "Collaborative plans for complex group actions", *Artificial Intelligence*, 86, pp. 269-358, 1996.

[2] N. R. Jennings , "Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems Using Joint Intentions", *Artificial Intelligence*, 75(2), pp. 195-240, 1995.

[3] K. Leiden & S. Green, "Trajectory Orientation: A Technology-Enabled Concept Requiring a Shift in Controller Roles and Responsibilities", *3rd USA/Europe ATM R&D Seminar*, Napoli, 2000.

[4] S. Marsella, J. Adibi, Y. Alonaizan, G. Kaminka, I. Muslea & M. Tambe, "On being a teammate: Experiences acquired in the design of robocup teams", *Proceedings of the International Conference on Autonomous Agents (Agents'99)*, 1999.

[5] S. Stoltz & P. Ky, "Reducing Traffic bunching more Flexible Air Traffic Flow Management", *4th USA/Europe ATM R&D Seminar*, New-Mexico, 2001.

[6] M. Tambe, "Towards Flexible Teamwork", *Journal of Artificial Intelligence Research*, 7, pp. 83-124, 1997.