# Supporting collaboration in air traffic control through flight contracts

Stéphane Chatty, Hélène Uninski and Stéphane Sire
CENA
7 avenue Édouard Belin
31055 TOULOUSE CEDEX, France
{chatty,uninski,sire}@cena.dgac.fr

*Abstract*—**Research on user interaction on air traffic control can lead to dealing with new air traffic management (ATM) concepts. That is especially the case for research on collaboration tools between controllers. Focusing on collaborations quickly shows ATM as a workflow, that is an information-based process where many different actors interact in succession with the same pieces of information. In this paper, we propose a simple model of ATM that captures its essence as a workflow: the flight contract model. We show how various ATM concepts can be seen as different implementations of that model: variations on the terms of contracts, the way they are negotiated, and the way they are managed by air traffic services. We propose to build ATM systems based on that model, so as to allow a smooth evolution of ATC. We then focus on the bottom end of flight contract handling: the work of air traffic controllers and especially their coordinations between sectors. We summarise our research on collaboration tools and explain our current directions. We finally show how the collaboration tools we propose rely on and take advantage of the flight contracts model.**

*Keywords*— **Air Traffic Management, workflow, groupware, Computer Supported Collaborative Work, flight contracts, direct collaboration**

## Introduction

Research on new air traffic management (ATM) concepts and research on the improvement of air traffic control (ATC) workstations are often carried out separately. However, there is a lot to learn about the organisation of ATM and its possible evolutions from the observation of the work of individual actors. Four years ago we initiated at CENA a fundamental research program on collaboration tools for air traffic controllers. Our final purpose was to deal with collaboration in ATM in general, but we first concentrated on tools for collaboration between control sectors: telephone exchanges and possible computer support for negotiation.

After obtaining results on computer-telephony integration and the type of computer support that was needed to ensure smooth collaborations, we reached a point where we could no more focus on the nature of collaborations only: we had to take their informational contents into account. Furthermore, our research suggested that we identify the *objects shared or exchanged* by controllers during their collaboration phases. In practice in many countries, there are no more such objects in today's air traffic control. But all the pieces of informations exchanged by controllers are related to the same abstract notion: the flight plan of aircraft. We thus ended up studying flight plans, their life cycle and the way they support collaborations, looking for a model that would support future evolutions of ATC.

Studying flight plans from the point of view of collaborations between users highlights the nature of air traffic management as a collaborative information system, ie. a *workflow* system. In such systems, efficiency gains can be expected from an integration of all subsystems, obtained by organising them around the data exchanged through a business process reengineering. It thus may be important to go through such a process for ATM, beginning with a modelisation of the data exchanged. In this article, we propose a framework for building such models: the *flight contracts* model. In addition to being a potential basis for a process reengineering, that model serves two other goals: on the one hand, it provides a useful design space for thinking about a number of new ATM concepts; and on the other hand, it allows us to build collaboration tools for air traffic controllers, which was our primary goal.

## I. ATM as a workflow

In the late 70s and early 80s the micro-computer revolution was prepared and accompanied by the emergence of research in computer-human interaction and the apparition of graphical interaction paradigms. Air traffic control has recently tried to benefit from the results of that research. In the same way, the emergence of networks has been accompanied since the mid 80s by a new research field: *computer supported collaborative work* (CSCW). At the confluence of human-computer interaction, computer science, office information systems, human factors, multimedia and sociology, that field tries to understand how computer systems might improve collaboration between humans. Such systems are known as *groupware* systems.

Depending on the work context or the activity (collaborative editing, games, videoconferencing, etc.) there are many different types of groupware systems. A first useful classification has been proposed by Ellis [7]. It sets groupware systems on two axes: time and distance, as shown in figure 1. Collaborations where users are interacting in real time are said to be *synchronous*. Otherwise, they are asynchronous, as in electronic mail or answering machines. Distance speaks for itself: collaborating users may be in the same room or on different continents.

Whereas research in computer science has focused on syn-

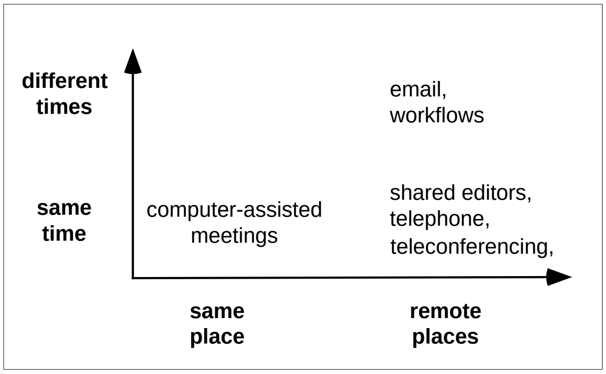


Fig. 1. The time-space classification of groupware.

chronous groupware because it is technically more challenging, most immediate applications of groupware are the extension of company information systems to support collaborative processes in companies. Such systems are known as *workflow* systems [1]. They belong to the top-right end of figure 1, and use email or more dedicated software technologies to support and enhance expenses handling in research laboratories, loans attribution in banks, or bug reporting and handling in software houses for instance. Workflows can be internal to a company, or involve actors from different companies working on the same information process. Usually, implementing a workflow system begins with analysing the flows of information that occur when the company is at work. This is used to build a model of the company processes, and very often to reengineer them, taking advantage of the situation to improve them. This is called a business process reenginering. Then, the adequate workflow software is developped. It is often centered around document circulation [16].

With the exception of air traffic control (that we will address in sections VI and VII), air traffic management matches the above definition of a workflow system. It is an information-based process, where the information is made of time slots, sector and route loads, user charges, departure times, flight plans and other related pieces of data. It is a collaborative system: capacities are declared by ATC centres to flow management units, who use them to give time slots to pilots in response to their flight plan application, etc. ATM is definitely a complex workflow system: its actors belong to many different companies, and often have conflicting goals. In that sense, ATM might be compared to stock exchanges for instance. That complexity, combined to the only recent interest in ATM (as compared to ATC), probably explains why current ATM systems are so little integrated. Not only do they use incompatible systems to implement different information flows (capacities on one hand, flight plans on another, etc.), but the process is even incomplete. Recent developments like the French ORCA system [2] and its European ASCOT equivalent aim for instance at controlling whether airlines have complied to the time slots allocated to them. Such complements to the current ATM systems are not much different from reporting systems offered to the directors of companies that use information systems. They both highlight the similarities between ATM and other workflows, as well as the current interest in fastening loose ends to make the process more consistent and integrated. Some approaches of Collaborative Decision Making (CDM) in air traffic management go further as what they suggest is close to a business process reengineering. When pleading in favour of an information system that provides full information and negotiation support to airlines about available time slots, they are proposing to reorganise and support part of the workflow process of ATM.

## II. From flight plans to flight contracts

The core of air traffic management systems is currently felt by many as being flight plans. One reason for this is that flight plans are currently the only pieces of data exchanged between air traffic services and airlines. Another reason is that flight plan management has long been the major software part of air traffic control systems: some engineers have become to believe that any extension to the ATC system should become an extension to flight plan management.

### *A. Limitations of flight plans*

Current flight plans, however, do not capture well the workflow process between airlines, crews, flow management units and en-route centres. First of all, they only represent the resources that were allocated to given aircraft, whereas other available resources are now nearly as important to airlines, since those resources are now perceived as limited. If airlines support CDM initiatives it is precisely because the current process based on flight plans is not sufficient for them.

Then, as new ATC technologies such as data-link [12], 4D FMS, or ASAS [3] becomes available, flight plans will have to evolve. Some evolutions might just change the vocabulary used: from waypoints to 4D points for instance. But other evolutions will be more radical, changing the semantic level of what is described. 4D points are just a more constrained way of describing trajectories. But what if ASAS brings us relative clearances such as ”stay 10nm behind this traffic”? Then the very notion of flight *plan* will be shaken because it is no more a plan in the plain sense: it is a new type of deal between the pilot and the ATC.

A third reason why flight plans are too limited is that a flight plan only represents a small part of the actual deal between aircraft operators and air traffic services. Actually, a flight plan is only a set of parameters to an implicit contract the other parts of which can be found in regulatory texts or are even implicit. For example in Europe when a time slot is allocated to a flight, the departure time in the flight plan can be anticipated by 5 minutes and delayed by 10 minutes tough this does not appear in the flight plan. In the same way, RFL is well known to be non-contractual, for instance. All these restrictions or tolerances that do not appear in flight plans belong to the implicit organisation of ATM and should not be overlooked when proposing or testing new concepts.

Finally, the current flight plans are associated to a certain type of relations between airlines and air traffic services providers, which is probably going to fade away in the future. Flight plans are not negotiated. In fact, the expression itself is wrong in that there is not a single flight plan. There is the flight plan that the airline submits and hopes to follow. Then the flow management unit returns another flight plan that is imposed on the airline: take it or leave it. Finally, during the flight there are two versions of the flight plan: the one entered in the FMS, and the one used by the ATC. Recent studies at CENA have shown discrepancies between those two pieces of data [15]. Then, flight plans are not negotiated, and neither are they followed up during the flight. Controllers do not have to comply with it, and they do not even know what happened to the flight earlier in time: they might be the third or fourth sector to impose a costly manoeuvre to an aircraft without even knowing. If a flight follow up was to be provided by air traffic services as an added value to customers, then flight plans would provide no support for that.

### B. *Toward flight contracts*

As stated and shown above, flight plans do not capture well the present and potential future relationships between operators. Consequently, they cannot serve as a basis for structuring a workflow process. However, flight plans contain essential information, and they currently are the core of air traffic control. The information they contain should undoubtedly appear in the information model of the process. This is why we propose to generalise the notion of flight plan to the notion of contracts between airlines and pilots on one side, and air traffic services providers on the other side. That is a generalisation in terms of data: the contract is all what partners have agreed upon before the flight departs. It is also a generalisation over time: whatever the time scale and available technology, there will still be a contract between partners as long as air traffic service providers exist. And finally, it is a generalisation over socio-economical relations, with the possibility to evolve from the current provider-to-user relation to a real provider-to-customer relation or even more sophisticated relations such as a free time-slot market.

With such generalisations, flight contracts are just an abstraction, of which the usefulness can be questioned. We believe, however, that such an abstraction can be useful in three different ways.

The first use of flight contracts consists in using them as the basis for air traffic control interfaces, as explained in section VII of this article.

Secondly, flight contracts are also useful to study ATM as a workflow, and eventually to build supporting systems. In order to model and support a workflow, one needs to identify the pieces of information that circulate and the way they are handled by actors in the process. Flight contracts definitely serve that purpose. First, they represent an essential notion, and suggest other essential notions to the ATM process: a contract (the flight) suggests the existence of resources (sectors, routes, available time slots), and some sort of billing (user charges). Then, contracts capture the nature of the exchanges between airlines and air traffic services. A contract is negotiated in a way or another. That is the first part of the process, mostly related to flow management. The contract is then transferred to operational operators: pilots and control centres. It is then implemented. This corresponds to air traffic control. During that phase, the contract can be amended in different ways, or even renegotiated through clearances or rerouting. Finally, whereas there is currently little feedback on the implementation of a contract, there could be several sorts of feedback. The airline could gather logs so as to evaluate the detailed performance of ATC or even their pilots' performance. The en-route centre could do the same. The resulting data could also be fed back to flow management units so as to refine capacity predictions. And finally, one could imagine a billing system based on the way the contract was implemented. Note that so far we did not have to go into the details of contracts: whatever their contents, the process would be roughly the same. One could even go as far as defining the architecture and some components of a supporting software without knowing what contracts are made of, because what is important is the workflow process. One might also suggest variations on the process – and supporting tools – still without entering the details of contracts. For instance, contracts might be negotiated in real time through a video-conferencing system, or available resources might be auctioned through a dedicated groupware system. That corresponds to ideas aired about Collaborative Decision Making systems.

The third use of the flight contract model is as a basis for analysing current and future ATM concepts. As we are going to show in the next sections, many ATM concepts can be seen as variations on the contents of flight contracts, variations on the way they are negotiated, and variations on the way they are handled during flights.

## III. Varying terms of contracts

A contract is made of a number of clauses, which define the mutual obligations of signers. That is usually all there is to find in a contract at first sight. But in a more subtle way, a contract is also made of specific words, that usually have a precise meaning in the domain it applies to. That is part of the contract too, in that it defines the meaning of clauses. In short, a contract is defined by a vocabulary (a set of words or names) and a set of constraints. By varying those two sets, one can explore a number of possible concepts for air traffic management.

Of course, there is currently no such thing as a contract between carriers and air traffic services operators, and consequently the vocabulary cannot be identified by reading contracts. But the vocabulary emerges from all regulatory texts, internal notes, computer systems and even dialogues among controllers or between pilots and controllers. It is composed of beacons, waypoints, routes, frequencies, that is the basic items of airspace structure. Plain and obvious as it may look, that vocabulary is essential to all dialogues, and therefore to

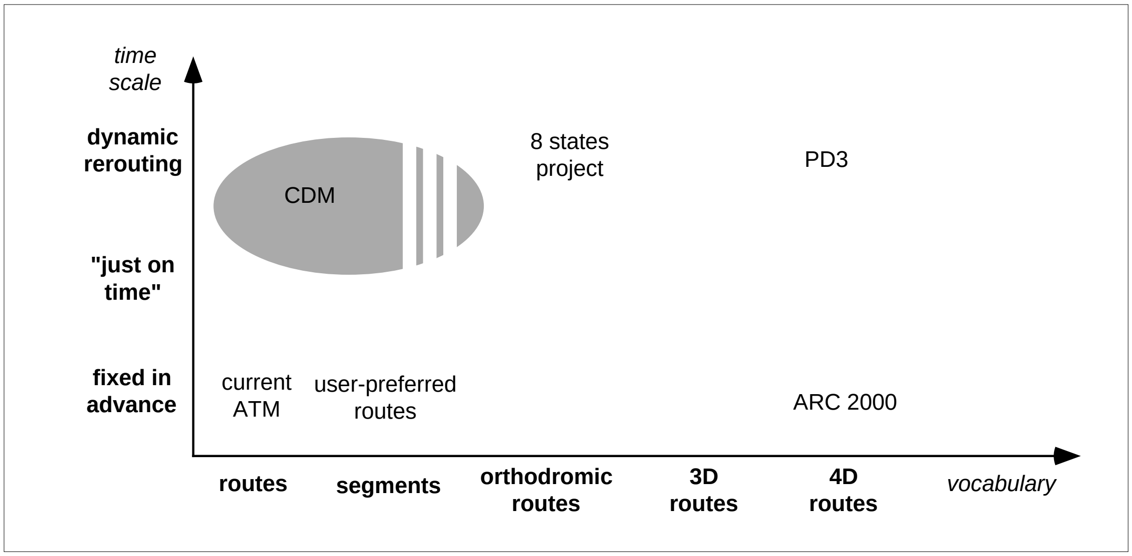


Fig. 2. A two-dimension space for ATM research: on one axis is the vocabulary used in contracts, and on the other axis is the way (and especially when) contracts can be negotiated.

every mutual commitment: what would clearances or flight plans become if waypoints did not exist or had no name? That is precisely the point: they would be much different. A number of recently tested ATC concepts have merely explored new vocabularies and some of their consequences on ATC: trajectories defined in 3D rather than through waypoints, of even 4D (3D + time) trajectories or tubes as in ARC 2000 or Phare Demonstration 3. Varying the vocabulary that way has a deep impact on possible dialogues in ATC. As for 3D and 4D trajectories for instance, it was often deemed impossible to have human dialogues with such vocabularies, and that is used as a justification for data-link communications.

Once a vocabulary has been chosen, partners may agree on a number of different things. That is true in ATM too. In our current systems, even though agreement is not formal, it consists of a certain type of aircraft or speed, a departure slot, a series of waypoints, etc. It goes along with a number of implicit rules on the way an aircraft should implement the contract (ie. fly) and the way controllers should implement it (ie. give clearances). Some definitions of free routing or user-preferred routes correspond to variations on the terms of contracts, without changing much their vocabulary. For instance, a possible choice of free routing might consist in limiting the constraints in large airspace zones to an entry point and an exit point, with an orthodromic route between them. Another possible definition would be to allow airlines to freely build their route from the available set of points, without imposing a set of predefined routes. Other similar variations are obviously possible. But it is also possible to imagine more radical variations on the terms of contracts. When an Airborne Separation Assurance System (ASAS) is available, one might want to introduce clauses such as ” aircraft will not get closer than 10nm from any preceding traffic on the same route ”.

Current contracts also contain a number of tolerances that are not always documented: pilots do not contact a sector immediately after leaving the previous one, they often cut angles, or do not respect departure slots with due precision. In a way, such tolerances are part of the contract, and changing them may have visible consequences on the global behaviour of the system or even on its efficiency. That may have been overlooked by some studies on new ATC concepts, which seem to have suppressed all tolerances along with changing the vocabulary. The rigid behaviour of many traffic simulators is a noticeable consequence of that approach, and might in turn create misconceptions about new ATM concepts. Ideally, the definition of a new concept should come with an analysis of what acceptable tolerances could be, what they might be, and what their consequences would be.

## IV. Varying contract negotiation

Independently of their contents, another important issue about flight contracts is the way they are negotiated. As we explained before, current flight plans are not negotiated in the strict sense. But another point about flight plans is that they are not easily renegotiated at a later time. That gives us two directions for varying contract negotiation: the social rules of negotiation, and the timescale of negotiation.

As regards social rules, current research on Collaborative Decision Making systems provides or will provide many options: whether airlines can just take the resources they are allocated (as in today’s ATM), or pick in available resources (through an adequate information system), or try and exchange resources with the provider, or will buy them at a price that depends on their relative value, or even will they exchange them as stock.

Time scale of negotiation has been addressed by CDM studies, by some interpretations of free routing, and by ATC projects like Phare Demonstration 3. It is related to the moment when airlines or pilots can try to negotiate or renegotiate their flight contract, whatever the social rules of negotiation are. The negotiation can be done long before departure, with no or little possibility to renegotiate. It is the current organisation. One can also imagine more flexible ways of renegotiating until departure time, for instance in case of a late departure. It has been suggested in some CDM scenarios [18], [21] and in some interpretations of free routing. Finally, renegotiation can occur when the aircraft is airborne. Actually, this kind of renegotiation is currently possible, as pilots can submit new flight plans while airborne or even reroute in case of emergencies. But this is not very usual. Some approaches of CDM have suggested to generalise that process, for instance by involving airline dispatchers in countries where they play an active role. ATC projects like Phare Demonstration 3 also have tackled that issue, by proposing a simple form of flight plan negotiation through data-link communications

A number of the previous considerations can be summarised on the two dimension graph of figure 2. On the X axis are represented different vocabularies or terms of contracts, and on the Y axis are the possible time scales for contract negotiation. A number of recently proposed evolutions have been set on that diagram according to what kind of contract negotiation they propose. Innovative studies that redefine the syntax of contracts, like ASAS, have not been represented as this does not fit easily along the X axis. In that diagram the name '8 states project' refers to the recent study launched by 8 northern European states on free routing.

## V. Varying contract handling

Usually, once a contract has been negotiated, it is implemented and followed up by partners. This is especially the case if one of the partners is a service provider: it will try and make sure that the customer is provided with all the needed support or quality of service. And at some point in time, the contract is used as a basis for billing the user. In ATM, those two phases roughly correspond to air traffic control and user charges. The latter is very simple today: flight plans and flight strips are used as a proof that an aircraft has used the air traffic services, but the relationship between the contents of the flight plan and user charges is very coarse. In the future, one might imagine finer dependencies: the charges might depend on the route and the time slot used. Even further, why not try and imagine a charging system that would depend on the quality of service: how much was an aircraft delayed? how many times did it have to change its route? how longer was its actual route? This brings us to contract follow-up: air traffic control.

Air traffic control is currently as a centralised and planified system where airspace users are imposed solutions that the central agent – a government agency until recently – thinks most efficient. When airlines try and explore decentralised air traffic control solutions, is it because they believe it is the most efficient solution? or just because they do not feel that the current nature of relationship is no more adequate? Perhaps variations of the way flight contracts are handled could be other interesting directions. One can identify at least two axes for variation: the extent of information sharing and decision making, and the underlying model for real time collaboration, especially within air traffic services.

Current ATC leaves nearly all information and decision with air traffic controllers. Pilots have information about their own aircraft, and gather pieces of information through the party-line effect of radio, but have no global view of the situation and their possible options. As regards decision making, they have little saying except in case of emergencies, or through tentative negotiations. Variants are possible, though. TCAS definitely introduced a variant: by getting more information about their context, pilots have a stronger feeling of controlling the situation and this definitely influences their relations with controllers. At the opposite, a straightforward implementation of data-link communications would reduce the information level of pilot. Other variants that provide more information, or less information, can probably imagined and would lead to a different nature of relationships between pilots and controllers. Then information and decision making can be extended to other parties, and especially airline dispatchers. This is the hypothesis of some CDM approaches.

Contract handling is also a collaborative activity within air traffic services: aircraft cross one sector after another and all those actors need to work and coordinate. One could then argue that all possible organisations of air traffic control tools and procedures are variants of flight contract handling. At least they have a major influence on the efficiency of contract. But focusing on ATC as part of a workflow, and thus collaboration itself, variants are possible on the architecture of workflow, the nature of exchanges and shared information between controllers, and the supporting tools that are offered to controllers to collaborate. The current workflow architecture of ATC is purely linear: while a executive controller is dealing with an aircraft, the planning controller of the next sector prepares its arrival, then it is the turn of his or her fellow executive controller to take the aircraft in charge, and so on. Research on multi-sector planning proposes to change that architecture to a more hierarchical one [19]. Then the contents of collaboration itself could be changed. Currently, controllers have only a very limited view of the overall life of a flight. This does not allow for much customer support, since there is no history available. A notable exception is the way controllers manage to negotiate long direct routes at night by calling several successive sectors: controllers are willing to provide the best service they can when it does not exceed the time they can spend on it. Providing them with more easily accessible information on the flight and its history might thus yield interesting results. And finally, the tools that are provided to controllers to collaborate with each other have an impact on the global efficiency of the ATC system. This is the topic of the next sections.

## VI. Collaborations in ATC

Whereas ATM in general is a workflow, that is a mainly asynchronous collaborative information system, ATC is a very special part of ATM in that everything in ATC happens more or less in real time. Consequently, collaborations in ATC are often synchronous collaborations, that call for different support than workflows. In this section we survey the specific needs of coordination between controllers, and the research alleys explored by ATC researchers. We then survey the results of research in synchronous CSCW and propose our approach to negotiation support: direct collaboration. We finally present our first step toward direct collaboration: DuoPhone. The next section will then focus on how the flight contract model can be used to go further toward direct collaboration in ATC.

### *A. Current tools and research*

The main reason why air traffic controllers need to collaborate is that the clearances they give to crews may have a consequence on the task of other controllers. They thus need to get last minute information, get an OK, or simply inform about their decisions. As the ATC system as evolved since its origins, two types of communications have emerged as well as the two supporting technologies. The first has been telephone, that provides for synchronous communications and thus safely covers all needs. However, telephone communications are costly in terms of working load, especially because they require two controllers to be available at the same time for talking. Consequently, telephone is not the adequate solution for non urgent communications, for instance for a route modification that occurs before the flight has been taken into account by the other sector. In addition, telephone is not efficient when an information should have multiple recipients. That is why computing systems known as flight data processing systems (FDPS) are used as another communication media, with an entirely different communication policy than telephone. Communication through a FDPS is asynchronous: the information entered by controllers is used to update a data base, so that it is up-to-date when other controllers use it, for instance when a strip is printed. More synchronous operation is sometimes possible by asking a strip to be printed on another sector, but this remains much less demanding than a phone call. This way, today's controllers are provided with complementary means of communications that address their different needs.

However, as the traffic load increases, telephone communications are progressively appearing as too demanding again: automatic coordinations and asynchronous communication through FDPS do not alleviate the load enough. That is why a number of solutions are being imagined to design new communication means, taking advantage of the computer technologies that are or should soon be available at the ATC workstation. Research is done on communication technologies as well as communication strategies implemented with those technologies. A good example of an alternative communication strategy is Eurocontrol's coordination system at Maastricht, though it is not current research at all: rather than calling another sector to ask for permission to clear a flight to a new level, the controller just enters the clearance into the system; the clearance then blinks for two minutes on radar screens, allowing other sectors to notice it and use their telephone to coordinate if necessary.

Recent research on coordinations has usually been focused on the use of graphical interfaces to collaborate. Two main directions can be identified. The first direction consists in enriching FDPS with more coordination facilities, as in the French Daarwin project [8]. However, that approach overlooks the fact that different communication strategies may be needed. Communication through a FDPS is essentially centralised and asynchronous: the FDPS acts as an intermediate agent between controllers. It is far from obvious that it is that type of communications that need to be improved. It rather appears that it is the other type that would need more support: the decentralised and synchronous one, currently based on the telephone. This is the second research direction. Transpositions of tools from office systems or Internet communications have been tried: email or popping dialogue windows. A successful example of such research is Eurocontrol's Sysco [10]. However, similar projects have been considered by French test controllers as too rigid for real collaborations. For that reason we decided to explore how the results of research on synchronous CSCW could be applied to ATC.

### *B. CSCW and direct collaboration*

As we mentioned earlier in this article, most research in CSCW has been performed on synchronous groupware. Until very recently, research was carried out in two distinct directions: communication medias and shared editing tools. Those two directions corresponded to different technologies, and thus concerned different types of specialists. On the one hand, a lot was done by ethnographists and telecommunications then multimedia specialists on videoconferencing, mediaspaces and other communication technologies [17]. On the other hand, specialists in distributed computing systems explored the shared editing paradigms: situations and tools where users remotely interact with the same data at the same time. The most popular paradigm was that of shared word processors, supposed to be useful when researchers themselves were collaborating on a paper for which the deadline was approaching or past. This lead to identifying a number of models and techniques for making such systems usable: mutual awareness techniques, shared widgets, etc [9]. However, so far such systems have failed to be applied as is in commercial systems.

When trying to apply synchronous groupware to real-life situations like we did for ATC and others are doing for other domains like banking, it soon appeared why shared editing systems were not adequate as is. The problem with them is that they make simplistic assumptions on the nature of collaborations and do not take into account more complex situations like negotiations. Furthermore, it appears that many real-life situations involve such complexities [5]. To begin with, collaborating is usually not the only task of users: it comes on top of other activities and is intertwined with them. This means that collaboration cannot be thought of as purely synchronous, because users may come and go from pure synchronous situations (talking over the phone for instance) to less synchronous ones (being on the phone but keeping the conversation on hold). Shared editors take little account of that. Then shared editors make the assumption that users have exactly the same goal, share all their information, and share it in a flat and plain way. In real life situations, people have conflicting goals, keep information from each other, or display information in an insisting way so as to try and convince each other. Finally, the techniques proposed by shared editors to exchange data or dialogue with each other are very formal and rigid, probably because they are transposed from distributed computing algorithms, where computers rather than users collaborate with each other. Trying to address those issues and especially the latter, we introduced the notion of *direct collaboration* as a property that groupware tools for situations like ATC should exhibit.

Direct collaboration can be best described by using the clover model of groupware introduced by Coutaz et al [4]. That model splits collaborative activities and the tools that support them in three facets: communication, production and coordination. *Communication* represents information exchanges between users: oral dialogues, Internet chats, emails, etc. *Production* represents the joint production of an artefact, which is usually the goal of the collaboration: production of a text or a drawing, for instance. This can be extended to situations like games, for which the produced artefact is the state of the game, or ATC. In that latter case, the produced artefact is the situation in the airspace or, using the flight contracts model, the set of contracts. Finally, *coordination* corresponds to all activities necessary to get the activities of other facets executed in an orderly and efficient way: turn taking, for instance. As far as shared editing is concerned, many techniques that have been introduced correspond to coordination: access rights, tools for taking roles, etc. Such tools are usually very obtrusive and that is where shared editors become too rigid. In real world situations, coordination is much more subtle. For instance, except for formal situations like a trial or a plenary session of an international organisation, turn taking is not an activity of its own: it is supported by communication itself through hints such as silences or prosody. There are other examples where coordination is supported by production: in a factory, giving an object to another workman is usually enough to ensure that this person will now be in charge of working on the object. We thus proposed the following definition of direct collaboration, inspired from the earlier notion of direct manipulation: a direct collaboration system is a groupware system where coordination is supported by production and communication, rather than by specific artefacts [20]. Of course, direct collaboration may or may not be a desirable property, depending on situations. As a rule of thumb, it is desirable except in specific complex situations where the collaboration procedure is too complex to follow without support or too safety critical to avoid the need of an automatic verification.

### *C. Direct collaboration and ATC*

In the case of ATC, we consider that direct collaboration is desirable in most situations. Our main reason for that is that current synchronous collaborations are performed through the telephone (or radio when pilots are concerned). This means that as it is, the collaboration support is a direct collaboration system, and is well accepted as is. Communications between sectors are not considered too complex or safety critical, in that no incident can be related to a bad coordination between actors (along the definition the clover model). They are considered too frequent and time consuming, which is a completely different issue. Therefore our goal is to produce more efficient systems, while keeping direct collaboration.

How can one build direct collaboration systems? By using the clover model and exploring collaboration situations, we identified patterns and supporting techniques that would allow for direct collaboration [20]. The first pattern is the availability of communication channels that support coordination hints. In the case of ATC, the telephone is such a channel, but its use is deemed too costly. For that reason, our first step toward direct collaboration was to try and lower the workload induced by the use of telephone. This was obtained by a better integration of the telephone and the computing system, in a system named DuoPhone. We also explored the use of video and its integration with the computing system, in a system named DuoView. However, video was not convincing enough for us to keep working on it: controllers thought of it as an interesting gadget. The second pattern that supports direct collaboration is rather a set of patterns, that are based on the use of production objects to support coordination. By exploring the way objects are used in the real world to embed coordination, social rules, persuasion and other social constructions [13], we were able to devise a research agenda to explore how those uses could be transposed to digital objects. For instance, the way an object is given to someone else is important: if it is a paper, for instance, it can be put on the side of the table or just in front of the other person; it can also be held above the table, or even agitated in front of the person's eyes, depending on how intrusive one wants to be. The use of time and space in the way objects are shared is clearly a pattern. In another direction, some specific objects embed more complex social rules: keys are a simple example, but more complex ones can be found. We currently are exploring such patterns and the way they can be transposed to ATC. However, this supposes that we identify production objects in ATC systems.

In the rest of this paper we will describe our work in those two research directions. We will first describe computer-telephony integration in DuoPhone and its formal evaluation with air traffic controllers. Then, in the next section, we will explain how we extended the flight contracts model to identify production objects in ATC collaborations, and we will describe our first prototype that uses those objects: DuoTrade.

### *D. Integrating communication and production: DuoPhone*

DuoPhone is a software layer developed at CENA to integrate telephone communications to any software application. By extension, it is also the name of a series of prototype ATC workstations that make use of that integration. DuoPhone uses the ISDN capability of Sun SPARCstations 10. This allows us to use a high quality voice link (digital telephone), with no constraints on the computer network. DuoPhone is a server modelled after the X Window System. It manages the ISDN link of the workstation, and allows client applications to put, answer, and transfer calls, connect the microphone and loudspeaker of the workstation to the phone link, and subscribe to events such as incoming calls or disconnections. DuoPhone also monitors the telephone or any device connected to the same ISDN outlet, so that end users may freely choose between their favourite device: telephone handset or computer microphone and loudspeaker. Finally, DuoPhone allows applications to manipulate the facilities offered by ISDN: caller ID and messages. Therefore, when an application puts a call, it can decide to send a short text message with the call. If the recipient of the call is another computer running DuoPhone, client applications are informed of the origin of the call and the contents of the message, and thus can decide how to handle or display the call.

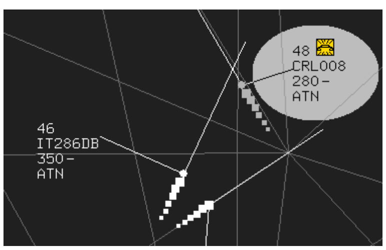


Fig. 3. Someone is calling about aircraft CRL008

Using DuoPhone allows a seamless integration of voice communications into the main user interface of an ATC workstation. For instance, we used DuoPhone to enrich our pen-based ATC workstation Grigri [6] with communication capabilities: we just had to add a gesture to the vocabulary understood by the workstation. In the final prototype used for a formal evaluation, we came back to a more classical interface, still based on a touchscreen however. Controllers have two touchscreens in front of them. One is a radar display, and the other is an emulation of the specialised keyboard used in French control centres to put phone calls. Controllers can put classical phone calls through the keyboard, or they can designate an aircraft on the radar display before putting the phone call. The workstation software interprets that as a call related to that precise aircraft. It thus sends the corresponding request to the DuoPhone server, passing the ID of the aircraft along with the request. DuoPhone then puts a phone call and uses the special capabilities of ISDN to send the aircraft ID as a message along with the call. As a result, the DuoPhone workstation that receives the call knows which aircraft it is about, and can display through an animation as shown in figure 3. This allows controllers to keep the voice link for what it is good at, freeing it from what makes it inefficient: locating the aircraft that caused the call.

A formal evaluation of DuoPhone was performed with 12 air traffic controllers from the French Bordeaux control centre. Its purpose was to evaluate the impact of knowing what aircraft a phone call is about. Controllers went through two simulations: one in which they received normal phone calls, and one in which they received DuoPhone calls. The numerical results showed a gain of about 15% on communication times. DuoPhone was subjectively rated as useful to very useful, and there was very little opposition as shown in the experimental report [11]. Other applications of DuoPhone were even suggested, such as a use in collaborations with the French airforce, which are known to represent a heavy workload in French ATC centres. On the overall, more than 90% of the controllers that were exposed to DuoPhone, whether through demonstrations or through the experiment, thought it was a good candidate for operational implementation.

## VII. A CONTRACT-BASED COORDINATION TOOL

We will now describe our second research direction in trying to develop more efficient direct collaboration tools in ATC systems. That direction is based on an extension of the flight contracts model. The extension makes it possible to introduce production objects that controllers can manipulate during their collaborations.

### *A. Amendments*

As explained in the previous section, many subtleties of human collaborations are carried by the objects manipulated during exchanges or common work, in addition to their primary role as the support of workflows. If one is to try and take advantage of computer-based communication channels in ATC, it might thus be useful to focus on the objects exchanged during coordinations between sectors. However, such objects are not obvious to the eye: controllers do not exchange physical objects anymore, and if there are objects exchanged, they are virtual. Aircraft (or their representations) and flight contracts are not good candidate either: they are not exchanged during coordinations. This lead us to identifying a new abstraction, which is at the heart of exchanges: not contracts themselves but amendment to the contracts, that is modifications that are made to the contracts after it is signed. If there were no such modifications, there would be no coordinations: everything would be silent. Coordinations exist because flight contracts are modified and it has a consequence on other sectors. That is why we introduced an addition to the flight contract model: the *amendment model*, that is mainly dedicated to describing ATC.

All the actions of a controller can be reinterpreted through the amendment model, thus allowing to build a invariant functional core around which many interface designs for ATC workstations can be experimented with:

- giving an instruction to a pilot corresponds to negotiating and integrating an amendment to a flight contract. In the future, this might be performed through datalink communications, and amendments might be negotiated with pilots and dispatchers if the implementation of a real-time downstream Collaborative Decision Making system is possible.
- taking notes on a paper strip after giving instructions corresponds to archiving an amendment. In the current system, most amendments are not stored in the computing system and thus are available only for future reference by the same controller or in case of an incident analysis. In future systems, minor amendments may or may not be discarded if entered into the computing system.
- taking tentative notes on a paper strip or a sheet of paper corresponds to drafting an amendment. In the future, this might be performed with a computing system if a very efficient interface is provided.
- similarly, entering data in the Flight Data Processing system corresponds to archiving an amendment.
- talking over the phone with another sector corresponds to negotiating an amendment within the ATS providers.

As suggested in the list, all the actions described above may or may not be implemented through the computing system: the current non-automated system can be interpreted through the amendment model as well as a fully computerised one. Similarly, many different interface designs are possible for those actions, depending on the interaction styles used and the way amendments are represented or event not represented. For instance, a fully datalink-based system as Phare Demonstration 3 groups creating, implementing and archiving of an amendment under a menu selection and never represents the corresponding amendment. At the opposite, the DuoTrade prototype we have developed explores the path of representing amendments and making them the basis of coordination between sectors.

### B. *DuoTrade: a first prototype*

We have developed a first prototype of an amendment exchange interface, named DuoTrade. This prototype is based on a classical ATC organisation with flight plans and radio links with cockpits. The only addition it supposes to current operational systems is a computerised version of annotations to flight contracts, whether they represent actual amendments of draft ones; DuoTrade is presently coupled to a variety of touchscreen input systems in CENA's demonstration workstation Toccata.

DuoTrade represents amendments by icons. As soon as a controller creates an amendment by entering instructions in the system (figure 4), the corresponding icon is created. Users have two ways of validating data input: the button

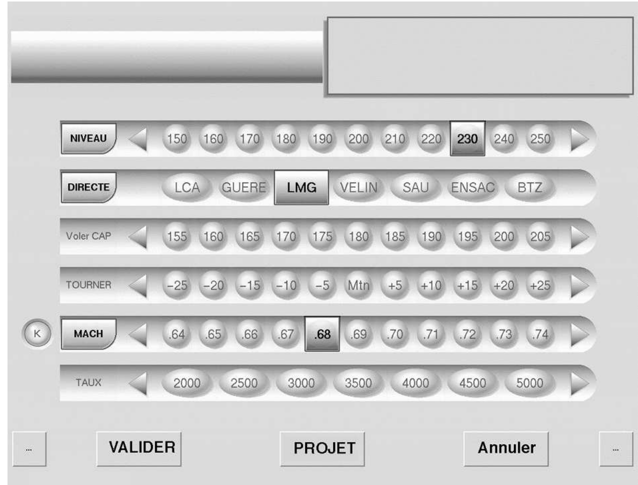


Fig. 4. The interface used to enter instructions into the computing system. Validating creates draft or implemented amendments depending on the button pressed.

VALIDER (validate) makes amendments considered as firm and stores their icons in a history window that is not directly visible, whereas the button PROJET (draft) stores them in a window of draft amendments on the controller's working surface. Double clicking (with a finger) on the icon of an amendment opens a full view of it. Though the current version presents the contents of an amendment in plain text (figure 5), future versions could allow pictorial representations if adequate (in a 4D environment for instance), and could also provide a link to the corresponding flight contract.

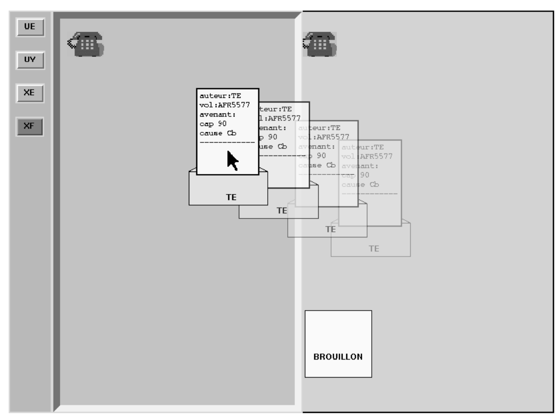


Fig. 5. Our first prototype for exchanging amendments. Here, an amendment is dropped over another controller's workin surface.

DuoTrade currently implements two variants for amendment exchange. The first variant is dedicated to asynchronous communications, and is not much different from Sysco messages. Non urgent amendment proposals can be picked and dropped over a zone that represents another sector. As a result, the amendment is transferred to the working surface of the other sector. In order to avoid conflicts of access to amendments and to improve the feeling of a workflow by making virtual objects as 'real' as possible, the amendment disappears from its source window as soon as it reaches its destination, and is replaced by a shadow. The recipient controller can then read the amendment at will, and choose between two courses of action: either accept the proposed amendment, or call back to negotiate. In the current ATC organisation, acceptance has to be sent to the sender by clicking a button, but with datalink communications a possible design might be to have the recipient controller to directly send the amendment to the aircraft, the sender being only notified. Negotiation can be triggered by another button, and uses computer-telephony integration: the sending sector is called and a phase of oral dialogue begins, with a possible joint edition of the amendment.

The second variant of amendment exchange is used when negotiations are immediately necessary. By dragging the icon of an amendment over the zone that represents another sector and waiting for a few fractions of a second, one obtains access to a representation of the other sector's working surface (figure 5). It is then possible to decide on the way to drop it: close to the centre of the surface or not, with agitating it before or not, open or closed, etc. It is also possible to drop the icon over the iconic representation of a telephone, thus triggering a phone call and reaching a dialogue situation as with DuoPhone.

### C. Discussion

There has been no formal evaluation of DuoTrade so far, and the current prototype is not enough integrated to a complete ATC environment to allow for that. Though such evaluations are planned in the near future, present discussions of the amendment-based can only be subjective. When exposed to the notion of contracts and amendments, the two controllers consulted during the design of DuoTrade found them natural. However, they questioned the manipulation costs an interface based on amendments would induce. A similar but deeper question was raised by one of the designers of Eurocontrol's Sysco. Introducing new objects to the control environment is not a neutral operation, as it adds to the conceptual model controllers have to master about their task. Thus, imposing the manipulation of such abstract objects might divert controllers from their main task, where the only objects are aircraft. That question will definitely need to be part of our evaluations, and will even be taken care of in the design of the interface that will be used for the evaluation. But the recent evaluations of Erato [14] hint that new abstract objects might be acceptable in ATC workstations. In its *reminder* Erato introduces abstract objects named problems that are organised along a time line and represent problems controllers have to solve and when. They seem to be well accepted by controllers, thus suggesting that well chosen (or rather well identified) abstract notions can be smoothly added to workstations.

Another concern that will need to be addressed in evaluations will be the effect of transforming a number of synchronous exchanges into asynchronous ones. Two major consequences will have to be measured. Firstly, asynchronous dialogues potentially have a very different structure. The efficiency and the nature of negotiations will then need to be evaluated. Secondly, asynchronous communications will be one of the several new sources of asynchronous events at the workstation, that is unexpected events that need controllers' attention. The increasing number of such events is going to challenge the controllers' attention capabilities, and adequate signals will need to be designed. Our team's project Avatars addresses that issue; nevertheless, that point will have to be specifically checked for communications in DuoTrade. However, in the case asynchronous exchanges do not appear convincing for operational use, we believe that contract amendments will still be a useful basis for exchanging information during phone calls.

## Conclusion

In the article, we have studied air traffic management and air traffic control as collaborative systems. We have explained why ATM is a workflow, and introduced the flight contracts model as an abstraction that would support a possible reengineering of ATM as a workflow system, and reveals itself as an efficient tool for surveying and analysing many current research works in ATM. This suggests that the flight contracts model or a similar model would provide valuable help in designing an ATM architecture that would support the progressive evolution toward a future ATM system through the successive implementation of new concepts and tools focused on the contents of contracts, the way they are negotiated or the way they are handled. We then focused on ATC and the synchronous collaborations it implies. We have highlighted several specific aspects of negotiation situations that are not addressed by the state of the art in groupware systems. We thus have introduced the direct collaboration property and described two research directions we took to provide direct collaboration in ATC: computer-telephone integration with DuoPhone, and exchange of amendments to flight contracts with DuoTrade. In the future, we plan to keep working on these two directions. We presently are exploring possible implementations of the DuoPhone through collaborations with Eurocontrol. As regards flight contracts and amendments, we are working on a more refined version of DuoTrade and on its extension to data-link communications. We also plan to work on other flight contract manipulation tools to be used in other parts of the ATM workflow, so a to contribute to the apparition of Collaborative Decision Making systems.

## Acknowledgements

The authors of this paper wish to thank Dominique Colin de Verdière (CENA) for his comments on the paper, as well as Alistair Jackson (Eurocontrol) for discussions on the amendment concept and Bernard Diot and Roger u (CRNA Bordeaux) for their comments on the project. DuoPhone and DuoView were developed by François-Régis Colin and Duo-Trade by Hélène Uninski.

## References


[1] K. R. Abbott and S. K. Sarin. Experiences with workflow management: issues for the next generation¡. In *Proceedings of ACM CSCW'94*. ACM Press, Oct. 1994.

[2] E. Allard. Répartition en catégories des vols à l'arrivée à Orly en 1996. Technical Report R97-025, Centre d'Études de la Navigation Aérienne, 1997.

[3] B. Bonnemaison, T. Miquel, and F. Casaux. Operational assessment of co-operative ASAS applications. In *Proceedings of the ATM'98 Seminar*, 1998.

[4] G. Calvary, J. Coutaz, and L. Nigay. From single-user architectural design to PAC*: a generic software architecture model for cscw. In *Proceedings of the ACM CHI*, pages 242–249. Addison-Wesley, 1997.

[5] S. Chatty, P. Girard, and S. Sire. Vers un support multimédia au collecticiel synchrone. *Technique et Science Informatiques*, 15(9):–, 1996.

[6] S. Chatty and P. Lecoanet. A pen-based workstation for air traffic controllers. In *Proceedings of the ACM CHI*. Addison-Wesley, 1996.

[7] C. Ellis, S. Gibbs, and G. Rein. Groupware: some issues and experiences. *Communications of the ACM*, 34(1):38–58, 1991.

[8] D. Figarol, J. Velten, S. Chambon, and V. Khang. Les etudes sur le traitement des vols du CAUTRA 5: le projet DAARWIN. Technical Report NT90-209, Centre d'Études de la Navigation Aérienne, 1990.

[9] C. Gutwin, M. Roseman, and S. Greenberg. A usability study of awareness widgets in a shared workspace groupware system. In *Proceedings of ACM CSCW'96*, pages 258–267. Addison-Wesley, 1996.

[10] A. Jackson. EATCHIP III evaluation and demonstration phase 2, SYSCO level 1 Technical Report Eurocontrol Experimental Centre, Sept. 1997.

[11] L. Karsenty and I. Pecaut. Évaluation d'un collecticiel d'aide aux coordinations dans le contrôle aérien. Technical Report NR98-798, Centre d'Études de la Navigation Aérienne, 1998.

[12] K. Kerns. Data link communication between controllers and pilots: a review and synthesis of the simulation literature. Technical Report MP-90W00027, The MITRE Corporation, 1990.

[13] B. Latour. *Petites leçons de sociologie des sciences*. Edition de la Découverte, Paris, 1993.

[14] M. Leroux. An in-depth evaluation of Erato tools. In *Proceedings of the ATM'98 Seminar*, 1998.

[15] R. Lourme. Évaluation de la faisabilité opérationnelle du concept EDYNAV. Technical Report NT98-208, Centre d'Études de la Navigation Aérienne, 1998.

[16] A. MacLean and P. Marqvardsen. *Crossing the border: Document coordination and the integration of processes in a distributed organisation*, pages 109–124. Kluwer Academic, 1998.

[17] M. Mantei, R. Baecker, A. Sellen, W. Buxton, and T. Milligan. *Experiences in the use of a Media Space*, pages 803–808. Morgan Kaufman, 1991.

[18] P. Martin, A. Hudgell, N. Bouge, and S. Vial. Improved information sharing: a step towards the realisation of collaborative decision making. In *Proceedings of the ATM'98 Seminar*, 1998.

[19] C. Meckiff, J. Nicolaon, and R. Chone. The tactical load smoother tool for multi-sector planning. In *Proceedings of the ATM'98 Seminar*, 1998.

[20] S. Sire and S. Chatty. Designing groupware for direct collaboration. Technical Report NR98-694, Centre d'Études de la Navigation Aérienne, 1998.

[21] M. Wambsganss. Collaborative decision making through dynamic information transfer. *Air Traffic Control Quarterly*, 4(2):107–123, 1997.