Vérification et correction d'un protocole de contrôle aérien

A. de Jacquier[†], T. Massart , C. Hernalsteen

Université Libre de Bruxelles, CP 212
Boulevard du Triomphe
1050 Bruxelles
Belgique

Tél: +32 2 650 55 91
Fax: +32 2 650 56 09
email: ajacqu@ulb.ac.be

10 février 1997

Résumé

Cet article présente l'analyse formelle d'un protocole de négociation entre un avion et la tour de contrôle pour obtenir une autorisation de décollage. La modélisation du protocole en LOTOS a permis de mettre en évidence certains dysfonctionnements. Le protocole étant déjà implémenté sur certains appareils, nous montrons comment le modifier pour obtenir une version sans erreurs, tout en conservant certaines entités telles quelles. L'environnement LOTOS Eucalyptus est utilisé pour soutenir notre démarche.

Mots-clé: Technique de description formelle, expérience industrielle, étude de cas, outil de vérification, systèmes à transitions labellées, LOTOS, protocole d'autorisation de vol.

1 Introduction

Avant qu'un avion ne décolle, il est nécessaire que son équipage, en particulier le pilote, mène à bon terme un certain nombre de procédures (vérifications, demandes d'autorisation, etc.). Une de ces procédures consiste à négocier avec la tour de contrôle le plan de vol qui devra être suivi et à obtenir l'autorisation de décoller avec ce plan de vol. Cette négociation se fait souvent via une connexion radio entre le pilote et le contrôleur qui a en charge la gestion de l'appareil.

L'organisation européenne Eurocontrol développe depuis des années des protocoles visant à automatiser ce type de procédures permettant ainsi de réduire les charges de travail du pilote et du contrôleur au décollage, d'accélérer les échanges d'information et d'augmenter la fiabilité de ces procédures (réduction des erreurs de communication entre le pilote et le contrôleur, etc.).

Cet article étudie le protocole d'Autorisation de Décollage DCL (Departure Clearance) [Eur96] recommandé par Eurocontrol entre l'avion et la tour de contrôle. Ce protocole, qui se veut être un premier pas pour supprimer les communications vocales radio entre le pilote et le contrôleur aérien, a déjà été implémenté sur certains appareils.

La prochaine section le décrit informellement, en s'inspirant de la norme [Eur96]. La section 3 décrit la manière dont nous l'avons spécifié formellement en LOTOS [BB87] et les problèmes

[†]Ce travail a été supporté par le Fonds National Belge de la Recherche Scientifique (FNRS).

découverts grâce aux outils de l'environnement Eucalyptus [Gar96] sont expliqués à la section 4. Dans la section 5, nous décrivons les modifications 'idéales' à apporter pour corriger le protocole. Cependant, étant donné que dans de nombreux appareils déjà construits, l'entité embarquée suivant ce protocole est déjà implémentée, et qu'il semble que le coût de modification du logiciel soit élevé, à la section 6, nous essayons de trouver une solution convenable assurant qu'aucun malentendu dangereux entre l'avion et la tour de contrôle n'est possible, tout en ne modifiant pas l'entité automatisée implémentée dans l'avion. Enfin la section 7 contient nos diverses conclusions sur ce cas d'étude.

2 Le protocole DCL et le service qu'il doit fournir

Cette section est composée de 3 parties. Tout d'abord, nous décrivons le protocole DCL tel que présenté dans [Eur96]. La deuxième partie essaye d'identifier les ambiguïtés ou omissions de la norme que nous avons pu éclaircir avec l'aide de spécialistes d'Eurocontrol. Enfin, la troisième partie lève les ambiguïtés et omissions décelées, décrit l'architecture du système et explique informellement le service attendu du protocole.

Description existante du protocole

Le protocole DCL est utilisé avant le départ d'un avion pour obtenir l'autorisation de décoller ainsi que son plan de vol. Le protocole est principalement décrit en anglais et est accompagné de diagrammes décrivant la séquence des événements. Les échanges s'effectuent à travers une *liaison* entre 2 entités automatisées, l'une embarquée dans l'avion, que nous dénommerons *avion* dans ce qui suit, et l'autre dans *la tour de contrôle* (ainsi nommée ci-après). Par ailleurs, le *pilote* d'un côté et le *contrôleur* de l'autre chapeautent ces 2 entités en participant aux négociations.

La figure 1 schématise l'architecture du 'système' considéré.

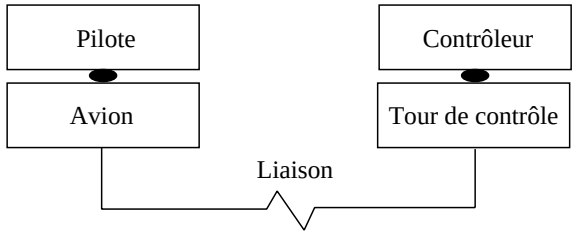

FIG. 1 – *Architecture du système*

Mode Normal

Le fonctionnement normal du protocole est très simple et consiste en une séquence de messages pour obtenir un accord entre le pilote et le contrôleur sur le plan de vol à suivre.

Le protocole débute normalement à l'initiative du pilote par une *demande d'autorisation RCD (Request Clearance for Departure)*, envoyée de l'avion à la tour de contrôle, indiquant les coordonnées de l'avion et la destination prévue. Si le RCD est valide et que l'autorisation ne peut être envoyée immédiatement, un acquittement du RCD est renvoyé sous forme d'une trame *FSM (Flight System Message)* d'acceptation indiquant que l'autorisation suivra. Ensuite, l'autorisation *CLD (DCL Clearance)* contenant les informations sur le vol ainsi que le plan de vol proposé est construite par la tour de contrôle, éventuellement avec l'aide du contrôleur, et envoyée à l'avion. Notons que si le CLD est disponible rapidement après la réception du RCD, l'envoi du FSM d'acceptation n'est pas obligatoire. Le pilote reçoit cette proposition de plan

de vol par l'intermédiaire de l'avion, et renvoit un message *CDA (abréviation de Departure Clearance Echoback?!)* donnant l'accord sur le plan de vol proposé. Enfin, la tour qui reçoit le CDA, renvoit un FSM d'acceptation à l'avion avant de clôturer le dialogue. La réception de l'acceptation par l'avion notifie au pilote que le plan est accepté et que le contrôleur l'a enregistré. Le diagramme temps-séquence en figure 2 décrit le mode normal du protocole DCL.

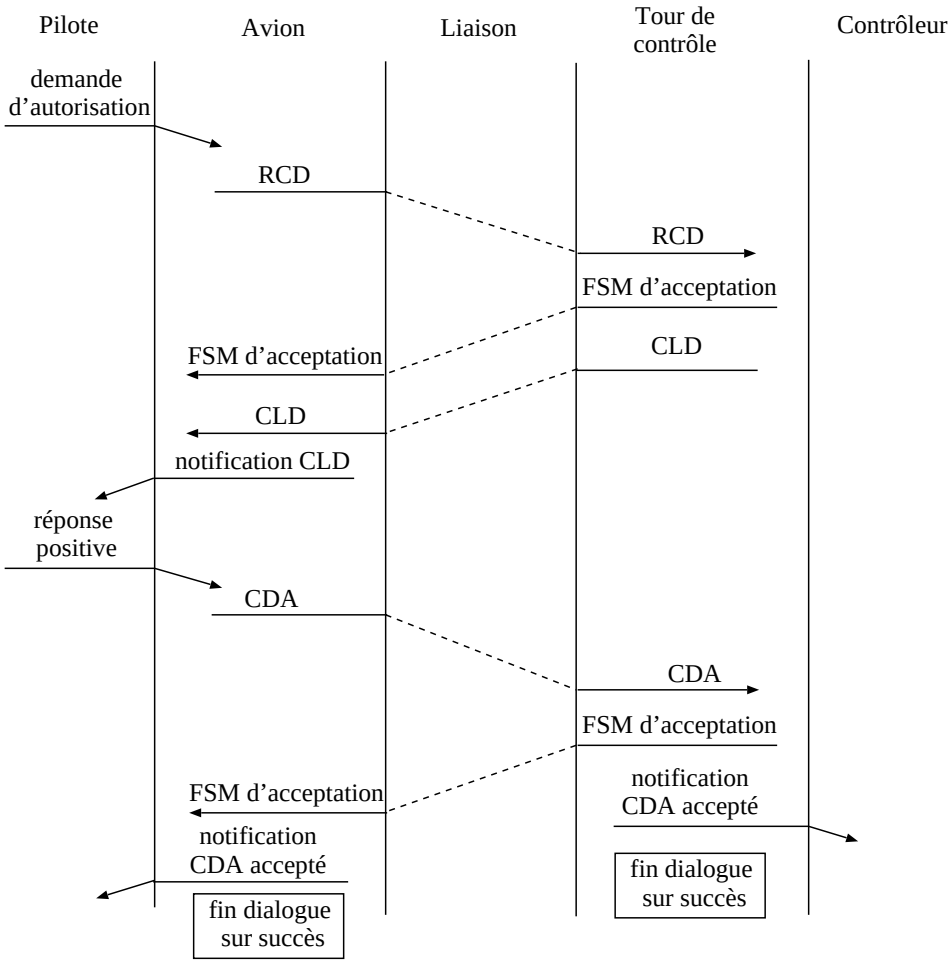

FIG. 2 – *Diagramme temps-séquence de DCL en mode normal*

Mode Anormal

Les faits suivants peuvent modifier le mode de fonctionnement normal du protocole:

- Un message RCD ou CDA peut être non valide et rejeté par la tour de contrôle.
- Le pilote peut refuser une proposition de vol.
- Le milieu de transmission de la liaison n'est pas fiable.
- Le pilote ou le contrôleur peut, à tout moment, demander la clôture du dialogue ouvert.
- Le contrôleur peut, en cours de dialogue ou après sa clôture, demander un accord sur un plan de vol modifié, ce qui entraînera un nouveau dialogue (en interrompant un dialogue éventuellement déjà en cours). De même, après un accord, le pilote peut faire une nouvelle demande.

Le protocole a ainsi prévu une série d'alternatives:

1. En cas de non réception, après un laps de temps donné, d'un message FSM, CLD ou CDA attendu, le pilote ou le contrôleur en est notifié, et le dialogue est clôturé.

2. Si les informations dans une trame RCD ou CDA ne sont pas acceptées par la tour de contrôle, un FSM de rejet est envoyé, et le dialogue est clôturé.

3. Un CLD non accepté par le pilote clôture également le dialogue.

Dans ces 3 cas, l'accord sur le plan de vol devra s'effectuer par transmission vocale radio[1]. De plus, le protocole prévoit une fonctionnalité supplémentaire. En cours de dialogue ou après sa clôture, le contrôleur peut demander la transmission d'une nouvelle autorisation de vol (CLD). Le traitement de ce nouveau CLD interrompt l'éventuel traitement de tout CLD précédant. Notons que tout message CDA envoyé par l'avion reprend toutes les données opérationnelles du message CLD qu'il acquitte. Si un message CDA, reçu par la tour de contrôle, ne correspond pas au dernier CLD envoyé (les données opérationnelles contenues dans le CDA ne sont pas identiques à celles du dernier CLD envoyé), le CDA est rejeté (un FSM de rejet est envoyé à l'avion).

Ambiguïtés ou omissions de la description existante

L'absence d'une description formelle complétant la description du protocole DCL dans [Eur96] entraîne quelques omissions ou ambiguïtés dans la description de certains éléments du protocole et de son architecture. Ainsi:

1. les caractéristiques du milieu de transmission assurant la liaison entre l'avion et la tour de contrôle n'y sont pas évoquées,

2. le type d'interface entre l'avion et le pilote d'une part, et entre la tour de contrôle et le contrôleur d'autre part n'est pas précisé. Il faut déterminer s'il est fiable ou non fiable (par exemple, le pilote peut-il ne pas voir certains messages?) et synchrone ou asynchrone (par exemple, le pilote peut-il envoyer un message pendant qu'un autre message s'affiche sur son écran?).

Compléments de description sur l'architecture, le protocole et le service

L'obligation de fournir une spécification formelle du protocole et de son environnement préconise l'intégration des éléments omis et la levée des ambiguïtés identifiées.

Ainsi, après discussion auprès de concepteurs du protocole DCL à Eurocontrol, il est apparu que:

1. La liaison utilisée entre l'avion et la tour est totalement non fiable et permet des pertes, dépassements et duplications de messages. De plus, le temps de transmission des messages est imprévisible.

2. Les interfaces pilote-avion et contrôleur-tour sont asynchrones et peuvent être non fiables au niveau de la réception des messages par les humains.

1. Sauf en cas de non réception du FSM d'acceptation suite au RCD. Dans ce cas, un nouveau RCD peut être renvoyé plus tard, à la demande du pilote.

Enfin, pour permettre la vérification du protocole, le point essentiel est de déterminer les séquences d'un dialogue qui sont acceptables au niveau du pilote et du contrôleur, ainsi que les séquences erronées.

En particulier la conclusion d'un dialogue peut se solder positivement par l'établissement du plan de vol, ou négativement. En cas de problème ou de désaccord constaté par le pilote ou le contrôleur, le dialogue est interrompu et cet accord devra s'effectuer plus tard, en utilisant une liaison vocale entre eux.

Le protocole peut donc, sans gravité, se terminer sans accord sur le plan de vol, même si un accord à travers le protocole est désirable. Cet accord peut être obtenu a posteriori par transmission vocale radio.

Par contre, la terminaison du protocole sur un succès du côté de l'avion et un échec du côté de la tour de contrôle, ou un succès de part et d'autre, mais sur des plans de vol différents est une erreur grave. Notons que si le protocole se termine en déclarant l'échec des négociations du côté du pilote, mais un succès du côté du contrôleur, la situation est incohérente mais pas critique, puisque le pilote ne fera pas décoller son avion avant d'avoir recontacté le contrôleur.

3 Spécification formelle du protocole

Le protocole DCL a été spécifié en LOTOS [BB87, ISO88] et vérifié grâce aux outils de l'environnement Eucalyptus [Gar96].

La vérification du protocole s'axant principalement sur la validation du comportement global cohérent des deux entités principales (l'avion et la tour de contrôle) communiquant à travers la liaison, nous n'avons pas inclus dans la spécification LOTOS du protocole les comportements du pilote et du contrôleur. Ces derniers sont considérés comme faisant partie intégrante de l'environnement du système. Signalons cependant qu'il peut être intéressant d'étudier le comportement du protocole en ayant spécifié explicitement les comportements du pilote et du contrôleur par des processus LOTOS. Certains types d'erreurs pourraient en effet être liés à leur comportement ou au caractère asynchrone de la communication qui les lie respectivement aux deux entités principales (voir [Mas93] où le type de problèmes possibles est analysé).

La spécification LOTOS du protocole DCL décrit les comportements des trois entités considérées: l'avion, la tour de contrôle et la liaison dont les comportements sont respectivement modélisés par les processus LOTOS AV, TC et $Liaison$. L'architecture globale de la spécification est représentée graphiquement à la figure 3. Le comportement global de la spécification est

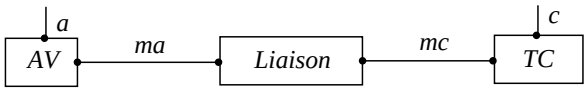

FIG. 3 – *Les trois processus de la spécification LOTOS*

décrit en LOTOS simplifié par $(AV|||TC)|[ma, mc]|Liaison$. Les processus AV et TC communiquent à travers le processus $Liaison$ via, respectivement, les portes ma et mc. Le processus $Liaison$ permet de modéliser le caractère asynchrone de la communication entre les deux entités principales. Les messages échangés entre le processus AV et le pilote, dont le rôle est joué par l'environnement du système, sont émis et reçus par le processus AV à travers la porte a. Il en est de même entre le processus TC et le contrôleur à travers la porte c.

Le comportement du processus $Liaison$ est décrit de manière à acheminer un message reçu par une des deux entités principales à l'autre. Cependant, comme nous l'avons dit précédemment, le milieu de transmission peut avoir un comportement anormal puisqu'il peut perdre, dédoubler, provoquer des dépassements de messages, etc. Pour éviter une explosion d'états (voir section

4), nous avons dû nous limiter, dans la spécification du processus *Liaison*, à la modélisation de deux types d'événements 'parasites' pouvant se produire dans le milieu de transmission. Ces deux types d'événements sont le croisement possible de messages émis par l'avion et la tour de contrôle dans le milieu de transmission, ainsi que le dépassement possible d'un message par un autre message émis par la même entité. Une pré-étude du protocole nous a en effet permis de conclure que ce sont ces deux comportements du milieu de transmission qui sont les plus intéressants à prendre en considération.

Les différents messages pouvant être émis ou reçus par les deux processus AV et TC ont été décrits avec les types de données LOTOS. Trois sortes LOTOS décrivent respectivement les trois classes de messages à considérer: ceux qui sont émis à travers la porte a (communication entre l'avion et le pilote), ceux qui sont émis à travers la porte c (communication entre la tour de contrôle et le contrôleur) et ceux qui sont émis à travers les portes ma et mc (communication entre l'avion et la tour de contrôle).

4 Vérification du protocole

Les outils utilisés

Nous avons travaillé avec l'environnement Eucalyptus [Gar96] qui regroupe plusieurs outils de travail pour le langage LOTOS. Parmi ceux-ci, *caesar* et *caesar.adt* [FGM+91, GS90, GT93] sont utilisés pour produire l'automate correspondant à la spécification LOTOS donnée en entrée. Cet automate implémente le système à transitions labellées dérivé à partir de la spécification et de la sémantique du langage LOTOS. C'est sur base de ce format d'automate que travaille l'outil de vérification *exhibitor* [Gar96] qui a été utilisé.

Deux modes possibles de production de l'automate existent. Le premier consiste à produire directement l'entièreté de l'automate correspondant à la spécification. Le second permet de produire cet automate de manière compositionnelle: un automate est produit pour chacun des processus invoqués dans le comportement principal de la spécification LOTOS et ils sont ensuite recomposés en un unique automate qui correspond alors à l'entièreté de la spécification. Tout automate produit par l'un ou l'autre de ces deux modes peut ensuite être réduit en utilisant l'outil *aldebaran* [Fer90]. Ce dernier permet de diminuer la taille d'un automate (en nombre de sommets et de transitions) en produisant un automate minimum équivalent modulo une certaine relation d'équivalence: équivalence forte, équivalence de branchement, etc. Réduire la taille d'un automate est très intéressant non seulement pour des raisons de stockage, mais aussi parce que le temps et l'efficacité d'exécution des outils qui manipulent un automate sont proportionnels à la taille de ce dernier. De plus, ces techniques de réduction offrent la possibilité de traiter des systèmes de plus grande complexité.

Sans réduction et en utilisant le premier mode de production, l'automate généré pour la spécification du protocole DCL comporte 805.302 sommets et 3.414.960 transitions. En utilisant le second mode de production et en réduisant chaque automate modulo l'équivalence forte avant de le composer avec un autre, l'automate final correspondant à la spécification comporte 6.781 sommets et 28.780 transitions, soit un gain global de 99,2 pourcents par rapport à la taille de l'autre automate. Ce résultat nous montre à quel point il est intéressant d'utiliser les outils de réduction (modulo une relation d'équivalence qui préserve le comportement) et de recomposition d'automates, ceux-ci pouvant être ensuite plus aisément manipulés par les différents outils.

Ces techniques de réduction ne peuvent cependant pas annuler les risques d'explosion d'états, problème classique lié aux algorithmes de génération d'automates. Même en utilisant ces techniques, il est possible de ne pas pouvoir générer l'automate correspondant à une spécification, les différents sommets et transitions de l'automate étant trop nombreux pour être stockés dans la mémoire de la machine utilisée. C'est à cause de ce phénomène d'explosion que nous avons dû

limiter, dans la spécification LOTOS du protocole, les comportements anormaux que peut avoir le milieu de transmission. Nous n'avons en effet jamais pu produire un automate pour une spécification du protocole où le processus *Liaison* modélisait en même temps **tous** les comportements possibles du milieu de transmission.

Afin de tester différentes propriétés de sûreté du protocole, nous avons utilisé l'outil de vérification *exhibitor* de l'environnement Eucalyptus. Cet outil accepte en entrée l'automate produit par les outils *caesar* et *aldebaran* ainsi qu'une expression régulière décrivant un canevas de séquences d'actions. *Exhibitor* vérifie alors s'il existe ou non une trace d'exécution de l'automate correspondant à ce canevas. Les tests effectués sur le protocole avec cet outil et les résultats qui en ont découlés sont exposés ci-dessous.

Les vérifications effectuées sur le protocole

De façon générale, la vérification d'un protocole porte sur des propriétés de sûreté et de vivacité. Rappelons qu'intuitivement, une propriété de sûreté assure que rien de "mauvais" ne pourra se produire, tandis qu'une propriété de vivacité assure que quelque chose de "bon" pourra toujours avoir lieu.

Dans notre cas, il y a principalement deux propriétés de sûreté à vérifier: l'absence de blocage et l'absence de traces non désirées; ainsi qu'une propriété de vivacité à vérifier: chaque dialogue ouvert doit respecter les différentes étapes du protocole et toujours se terminer du côté du pilote et du contrôleur par une notification de succès ou d'échec.

Les différentes vérifications que nous avons effectuées ont essentiellement été dirigées pour vérifier si le protocole, tel qu'il a été décrit, satisfait bien le service qu'il est censé fournir (voir section 2). Suite à une exécution d'une session de dialogue avec ce protocole, nous pouvons discerner quatre cas de figure potentiels:

1. Les deux entités principales terminent toutes les deux leur exécution sur un échec.
2. L'entité embarquée dans l'avion termine son exécution sur un échec et celle située dans la tour de contrôle termine la sienne avec succès.
3. L'entité embarquée dans l'avion termine son exécution avec succès et celle située dans la tour de contrôle termine la sienne sur un échec.
4. Les deux entités principales terminent leurs exécutions avec succès.

Les deux premiers cas de figures sont acceptables car ils ne constituent pas des erreurs graves. En effet, dans chacun de ces deux cas, l'entité s'exécutant dans l'avion terminant sur un échec, le pilote n'ayant pas obtenu d'autorisation de décoller devra établir un contact radio avec la tour de contrôle pour recommencer la procédure vocalement, comme le prévoit le protocole. Le troisième cas n'est par contre pas acceptable. En effet, l'entité s'exécutant dans l'avion terminant son exécution avec succès, le pilote pense avoir obtenu un plan de vol et une autorisation de décoller alors qu'une erreur s'est produite dans la tour de contrôle. Si le contrôleur met trop de temps pour établir le contact radio avec le pilote, il se peut que celui-ci ait déjà entamé les manoeuvres nécessaires pour décoller. Le quatrième et dernier cas est également très intéressant à étudier. Nous pouvons le diviser en deux sous-cas. Le premier est la situation où à la fois le pilote et le contrôleur disposent du même plan de vol, ce qui correspond au cas tout à fait normal. Le seconde possibilité est la situation où, bien que l'avion et la tour de contrôle aient terminé leur dialogue avec succès, le pilote et le contrôleur ne disposent pas du même plan de vol. Ce scénario, s'il est possible, permettrait à l'avion de décoller avec un plan de vol qui ne correspond pas à celui dont dispose la tour de contrôle, avec les conséquences que cela peut entraîner.

C'est sur les occurrences potentielles de ce scénario et du troisième que nous avons axées nos vérifications. Pour pouvoir modéliser et vérifier ces scénarios, certaines actions ont été rajoutées au sein du protocole de façon à pouvoir étudier la terminaison avec ou sans succès des deux entités principales. Ces actions ont été utilisées pour exprimer, via un canevas de séquences d'actions, les deux scénarios qui nous intéressent. Nous allons voir dans la sous-section suivante que les tests ont été concluants puisqu'ils ont révélés que ces deux scénarios étaient possibles. En effet, pour chacun d'eux, *exhibitor* a trouvé plusieurs traces d'exécution de l'automate produit pour le protocole.

Résultats des tests et analyse des erreurs identifiées

Pour chacun des deux scénarios qui nous intéressent, nous présentons, une session de dialogue du protocole identifiée par *exhibitor* qui montre clairement l'occurrence possible du scénario correspondant et qui illustre bien les problèmes posés par le protocole.

Chacun de ces deux exemples est présenté sous forme d'un diagramme temps-séquence et est accompagné d'une brève analyse.

Désaccord entre l'avion et la tour de contrôle

Le diagramme temps-séquence associé à ce scénario, qui se solde par un succès du côté de l'avion et un échec du côté de la tour de contrôle, est présenté à la figure 4. L'ouverture d'une

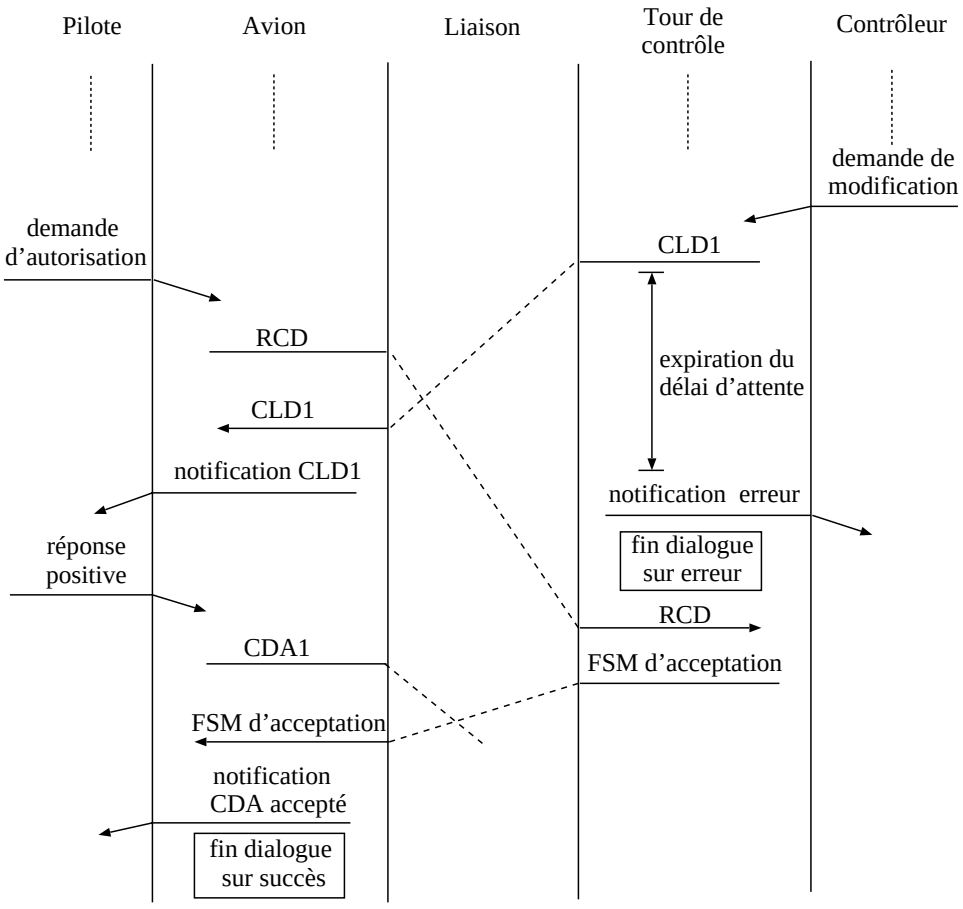

FIG. 4 – *Diagramme temps-séquence du désaccord entre l'avion et la tour de contrôle*

session de dialogue du protocole est initiée du côté de la tour de contrôle par la réception d'une demande de modification du plan de vol émis par le contrôleur. Cette possibilité est prévue par le protocole et a généralement lieu lorsque le contrôleur souhaite modifier le contenu du dernier plan de vol envoyé à l'avion. Suite à cette requête, un nouveau plan de vol (CLD1) est composé et envoyé à l'avion. Supposons qu'au même moment, le pilote de l'avion souhaite également obtenir un nouveau plan de vol et que c'est sa demande qui initie le dialogue du côté de l'avion. Celui-ci transmet alors la requête à la tour de contrôle par le biais d'un message RCD. Faisant suite à cet envoi, l'avion reçoit le plan de vol préalablement émis par la tour de contrôle. Il apparaît déjà ici une première confusion: du point de vue de l'avion, ce plan de vol est une réponse naturelle à sa demande alors qu'en réalité, il a été envoyé sur l'initiative du contrôleur. L'avion transmet alors ce nouveau plan de vol au pilote qui lui notifie son accord. Il compose donc un message CDA pour le plan de vol accepté et transmet ce message à la tour de contrôle. Supposons maintenant que le délai d'attente de la tour de contrôle pour recevoir une réponse au plan de vol ait expiré. L'entité s'exécutant dans la tour de contrôle avertit le contrôleur de cette erreur et termine ensuite son exécution. Une nouvelle session de dialogue est ensuite ouverte du côté de la tour de contrôle sur la réception du message RCD préalablement envoyé par l'avion (bien qu'une erreur ait été détectée, le protocole n'interdit pas l'ouverture d'une nouvelle session de dialogue). La tour de contrôle acquitte alors positivement cette requête par l'envoi d'un message FSM d'acceptation. Ce message est ensuite reçu par l'avion qui l'interprète comme étant un acquittement positif du message CDA préalablement envoyé, alors qu'en réalité, il n'en est rien.

Il est clair que la deuxième session de dialogue ouverte dans la tour de contrôle par la réception du message RCD se terminera, comme la première session, sur une erreur. Cependant, l'unique session de dialogue exécutée par l'avion s'étant terminée avec succès, le pilote ne sait pas qu'une erreur est survenue du côté de la tour de contrôle et conclut donc qu'il a le droit de décoller. Plus le contrôleur mettra du temps à établir le contact radio suite à la première erreur notifiée par la tour de contrôle, plus les manoeuvres entamées par le pilote pour décoller seront avancées.

Deux conditions sont nécessaires à l'occurrence de ce scénario. L'avion doit avoir émis le message RCD avant la réception du plan de vol émis par la tour de contrôle et le délai d'attente lancé par celle-ci, en attendant une réponse au plan de vol, doit avoir expiré avant la réception du message RCD. Nous pouvons identifier plusieurs causes ayant rendu possible l'occurrence de ce scénario:

- Ni l'avion, ni la tour de contrôle n'ont pu se rendre compte que deux sessions de dialogue de la tour de contrôle ont communiqué avec une seule et même session de dialogue de l'avion.
- L'avion n'a pas pu s'apercevoir que le plan de vol envoyé par la tour de contrôle ne constituait pas une réponse au message RCD préalablement envoyé.
- L'avion n'a pas pu s'apercevoir que le message FSM d'acceptation envoyé par la tour de contrôle était destiné à acquitter le message RCD et non pas le message CDA.
- La tour de contrôle a accepté d'ouvrir une nouvelle session de dialogue alors que la session précédente s'était terminée sur une erreur.

Accord sur des plans de vol différents

Ce second scénario montre que le protocole DCL peut être mis en défaut si un dépassement de messages particuliers se produit dans le milieu de transmission. Nous allons voir que dans ce cas, l'avion obtient de la tour de contrôle l'autorisation de décoller avec un plan de vol qui ne correspond pas à celui qu'a enregistré la tour de contrôle. Ceci est d'autant plus grave que

ni l'avion, ni la tour de contrôle n'auront l'opportunité de se rendre compte de cette erreur. Le diagramme temps-séquence associé à ce scénario est présenté à la figure 5.

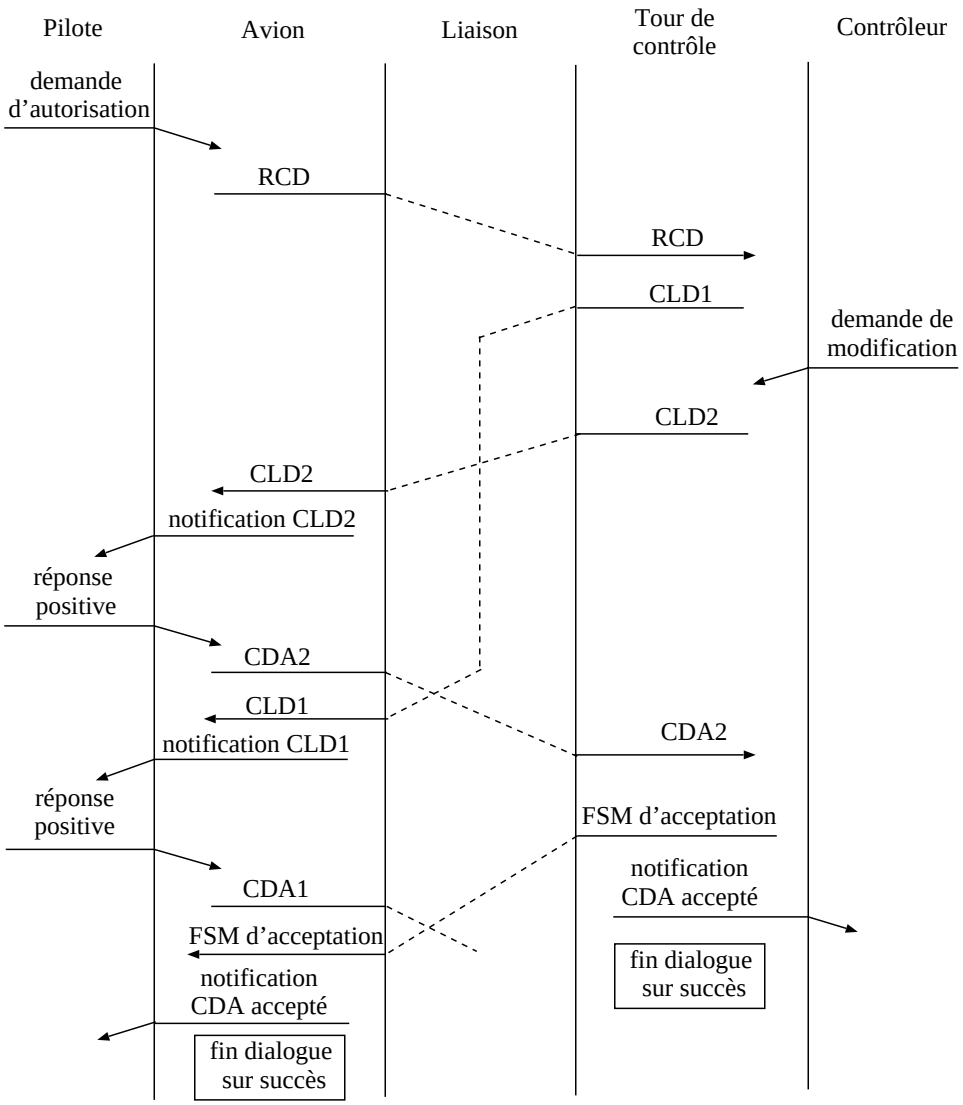

FIG. 5 – *Diagramme temps-séquence de l'accord sur des plans de vol différents*

Le dialogue est initié du côté de l'avion par la réception d'une demande d'autorisation de décoller émise par le pilote. L'avion compose donc un message RCD qu'il envoit à la tour de contrôle. Celle-ci ouvre une session de dialogue sur la réception de ce message et compose un plan de vol (CLD1) qu'elle envoit à l'avion. Le contrôleur décide ensuite de modifier ce plan de vol pour une raison quelconque et un nouveau plan de vol (CLD2) est alors envoyé à l'avion. C'est ici que se produit le dépassement de messages évoqué plus haut: le plan de vol CLD2 est délivré à l'avion par le milieu de transmission avant le plan de vol CLD1. Le pilote approuve ce plan de vol et l'avion compose alors un message CDA2 (correspondant au plan de vol CLD2) qu'il envoit à la tour de contrôle. L'avion recevant ensuite le plan de vol CLD1 du milieu de transmission en déduit logiquement qu'il s'agit d'une modification envoyée par la tour de contrôle et annule donc, comme le prévoit le protocole, toutes les opérations effectuées pour le plan de vol CLD2 préalablement reçu. Le pilote acquitte ensuite positivement le plan de vol CLD1 et un nouveau message CDA1 (correspondant au plan de vol CLD1) est alors envoyé à la tour de contrôle. Celle-

ci recevant le message CDA2 conclut que l'avion a bien reçu et positivement acquitté le dernier plan de vol transmis (CLD2). La tour de contrôle envoit donc un message FSM d'acceptation à l'avion pour lui signifier qu'il a l'autorisation de décoller (avec, du point de vue de la tour de contrôle, le plan de vol CLD2), communique cette information au contrôleur et termine sa session de dialogue avec succès. L'avion recevant ensuite le message FSM d'acceptation conclut que la tour de contrôle a acquitté positivement le dernier message CDA1 envoyé et qu'elle lui donne donc l'autorisation de décoller avec ce plan de vol (CLD1). Pour terminer, l'avion communique cette information au pilote et termine sa session de dialogue avec succès.

Ce qu'il adviendra ensuite du message CDA1 envoyé par l'avion n'a pas d'importance. il peut être perdu ou bien délivré à la tour de contrôle. Dans ce dernier cas, il sera ignoré par la tour de contrôle puisque le protocole prévoit de détruire tout message reçu qui serait déséquencé.

Il y a essentiellement deux causes qui expliquent l'occurrence possible de ce scénario:

- L'avion ne s'est pas aperçu qu'il y avait eu un dépassement de messages dans le milieu de transmission.
- L'avion n'a pas pu s'apercevoir que le message FSM d'acceptation envoyé par la tour de contrôle était destiné à acquitter le message CDA2 et non pas le message CDA1.

Une analyse de ce scénario nous permet de constater que le cas de figure présenté peut être aisément généralisé. En effet, comme le montre la figure 6, tous les scénarios satisfaisant les trois

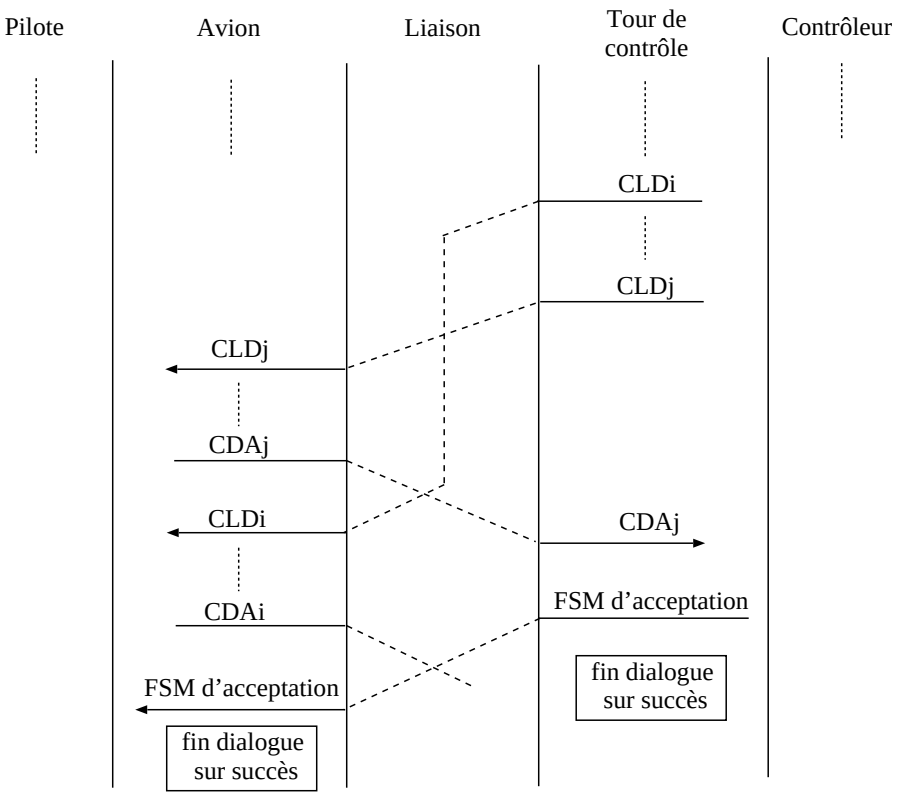

FIG. 6 – *Diagramme temps-séquence général de l'accord sur des plans de vol différents*

conditions suivantes se solderont par ce résultat:

- Le **dernier** CLD envoyé par la tour de contrôle (CLDj) doit parvenir à l'avion avant un autre CLD préalablement transmis (CLDi).

- Le message CDAj acquittant positivement le plan de vol CLDj doit être envoyé par l'avion avant la réception du plan de vol CLDi.
- Le message CDAi acquittant positivement le plan de vol CLDi doit être envoyé par l'avion avant qu'il ne reçoive le message FSM d'acceptation envoyé par la tour de contrôle.

Autres événements parasites liés au milieu de transmission

En plus des dépassements de messages, nous avons vu qu'il était possible que d'autres événements "parasites" se produisent dans le milieu de transmission, tels que la génération de doublons et les pertes de messages.

Il est clair que les pertes ne peuvent pas, à elles seules, être à la source d'une erreur dans le protocole. En effet, le temps d'attente des deux entités principales pour la réception d'un message est toujours borné par une quantité fixe de temps. Si un message n'est pas reçu par une entité dans le délai fixé, une erreur sera levée et communiquée à l'humain qui chapeaute l'entité (le pilote ou le contrôleur), lui signifiant qu'il doit recommencer la procédure vocalement.

Par contre, il est possible que la génération de doublons dans le milieu de transmission mette en défaut le protocole, de la même manière que les croisements et dépassements de messages peuvent le faire. Par exemple, le cas général présenté à la figure 6 est également valable si le message CLDi est un doublon d'un message CLD préalablement transmis. D'autres erreurs peuvent être conséquentes au dédoublement de messages FSM. Par exemple, un FSM d'acceptation envoyé pour un RCD se dupliquerait pour acquitter malencontreusement un CDA.

Enfin, d'autres dépassements de message peuvent également conduire à des erreurs (cas d'un FSM destiné à acquitté un RCD et qui ne serait délivré à l'avion qu'après l'envoi d'un CDA acquittant un CLD reçu après l'envoi du message RCD).

Nous proposons, dans la section suivante, les corrections idéales qu'il faut apporter au protocole pour empêcher l'occurrence de ce type de scénario et, d'une manière plus générale, pour empêcher que l'entité embarquée dans l'avion ne termine son exécution en ayant communiqué au pilote un plan de vol qui ne correspond pas à celui qu'a enregistré la tour de contrôle.

5 Modification du protocole sans contraintes

L'étude des deux scénarios d'erreur présentés en section 4 a permis d'identifier un ensemble de circonstances qui ont mené à la défaillance du protocole. En analysant plus profondément ces deux scénarios, on remarque qu'ils proviennent des mêmes problèmes. En effet, dans les deux cas, un message FSM d'acceptation est interprété différemment entre l'avion et la tour de contrôle. Dans le premier scénario, la tour de contrôle envoit un message d'acceptation d'un RCD qui est interprété par l'avion comme l'acquittement d'un CDA. Dans le second scénario, la tour de contrôle acquitte, via un FSM d'acceptation, un certain CDA alors que l'avion l'interprète comme un acquittement d'un autre CDA.

Ce type de confusion est rendu possible parce qu'aucune des deux entités n'a la possibilité de s'assurer qu'un message reçu est correctement séquencé ou de s'assurer qu'il constitue bien la réponse de l'autre entité au dernier message envoyé. Les erreurs identifiées sont toutes des conséquences d'un déséquencement de messages notamment suite à des dépassements ou duplications.

Ce problème peut être enrayé en plaçant dans chaque message émis par une entité un numéro d'identification unique pour cette entité. En attribuant les numéros de manière croissante, une entité peut aisément s'assurer qu'un message reçu est correctement séquencé en vérifiant que le numéro qu'il porte suit le numéro du dernier message préalablement reçu. De même, si un message (FSM) est transmis dans le but de répondre à un message émis par l'autre entité, il

est nécessaire, pour éviter toute confusion possible, d'y insérer le numéro du message auquel il répond.

En corrigeant le protocole de cette manière, les scénarios qui ont été évoqués dans la section 4 ne peuvent plus se produire. D'une manière plus générale, ces corrections assurent qu'il n'est pas possible que l'entité embarquée dans l'avion confirme au pilote un plan de vol qui ne serait pas celui enregistré et acquitté positivement par la tour de contrôle. Notons que l'apport de ces modifications peut de plus permettre d'étendre les fonctionnalités du protocole, en y intégrant par exemple des mécanismes de retransmission de messages qui n'ont pas été acquittés dans les délais fixés.

Nous avons introduit dans cette section la façon intuitive permettant de résoudre les problèmes relevés en section 4. Cette résolution préconise la modification des deux entités principales du protocole. Il n'est cependant pas envisageable, dans la réalité, de modifier l'entité embarquée dans l'avion, les coûts d'une telle opération étant très élevés. La section suivante propose une correction du protocole en tenant compte de cette contrainte.

6 Correction du protocole sous contraintes

Nous décrivons ici notre solution pour corriger le protocole DCL tout en ne modifiant pas l'entité embarquée dans l'avion. Après étude des solutions possibles et étant donné le risque de confusion lié aux dépassements et aux duplications possibles de messages à travers la liaison, les modifications suivantes se sont imposées:

1. Aucune demande d'accord sur un plan de vol modifié ne peut être émise par le contrôleur. De plus, aucune demande de modification provenant de l'avion ne sera prise en compte. Ceci assure qu'il n'y aura jamais dans le milieu de transmission deux messages CLD différents qui, comme nous l'avons vu, peuvent être à la source d'une confusion entre les deux entités communicantes. Cette restriction assez importante résulte de l'analyse du scénario d'accord sur des plans de vol différents présenté à la section 4. Notons également que sans cette limitation, il n'est pas possible d'éviter qu'après un accord entre les deux parties, le contrôleur demande une modification du plan de vol qui se soldera par un échec unilatéral de son côté, par exemple dû au fait que la liaison ne laisse plus passer aucun message.

2. Les messages FSM d'acceptation ou FSM de rejet ne peuvent pas être utilisés pour acquitter deux types de messages différents (RCD et CDA) puisque, dans ce cas, une confusion peut également avoir lieu.

   De ce fait,

   (a) plus aucun FSM d'acceptation ni de rejet n'est transmis en réponse à un RCD. Soit le CLD est prêt à temps et est envoyé à l'avion, soit, après expiration du délai, le pilote sera averti de l'absence de réponse à sa requête. Comme précédemment, il lui sera alors loisible de faire une nouvelle demande via le protocole DCL ou en utilisant une transmission radio.

   (b) de la même façon, plus aucun FSM de rejet n'est transmis en réponse à un RCD déséquencé.

3. Enfin, pour éviter le problème du désaccord entre l'avion et la tour de contrôle décrit à la section 4, il faut empêcher que la tour de contrôle puisse entamer avec l'avion un dialogue ultérieur à un autre où un message CLD a été envoyé.

Pour vérifier ce protocole modifié, nous avons généré les 3 automates pour les nouveaux processus LOTOS *AV*, *TC* et *Liaison*. Ce dernier a été spécifié de manière à permettre l'échange

simultané de 2 messages dans chaque sens avec pertes, duplications ou dépassements possibles. Nous les avons ensuite recomposés après les avoir minimalisés modulo la relation d'équivalence forte. Après réduction finale modulo cette même relation d'équivalence, l'automate global étendu est formé de 346 sommets et de 1082 transitions. Après analyse en utilisant *exhibitor* et *aldebaran*, nous avons pu vérifier que les propriétés de sûreté et de vivacité étaient bien satisfaites.

7 Conclusions

Nous avons, dans cet article, spécifié et vérifié un protocole de contrôle aérien. Cette vérification nous a permis de détecter un certain nombre de scénarios mettant en évidence des dysfonctionnements du protocole. Nous avons montré que, de façon générale, le type de liaison utilisé pour transmettre les informations entre l'avion et la tour de contrôle étant excessivement peu fiable (pertes, dépassements, doublons, etc.), le protocole n'est pas sûr. En effet il permet au pilote d'avoir un accord sur un plan de vol quand le contrôleur aérien reçoit l'information inverse. Il permet également un malentendu sur le plan de vol à utiliser.

Une analyse plus fine du protocole a montré que certains mécanismes habituellement utilisés dans les protocoles de réseaux ne sont pas présents dans celui-ci, tels que l'identification précise de certains messages, le contrôle des déséquencements de messages, etc. Une solution utilisant ces mécanismes a été proposée.

Enfin, étant donné le coût qu'occasionnerait la modification du logiciel embarqué dans certains appareils, nous avons proposé une solution 'sûre' pour corriger le protocole, c'est-à-dire ne permettant pas de malentendus dangereux entre le pilote et le contrôleur tout en assurant un certain service et en continuant à utiliser exactement la même entité de protocole qu'actuellement au niveau de l'avion.

Notons que ce type de modifications pour corriger un protocole tout en ne modifiant qu'une des deux entités communicantes est généralement difficile voire impossible à obtenir.

Notre étude a été supportée par l'utilisation de l'environnement LOTOS Eucalyptus. Cet environnement est en phase d'intégration et de stabilisation mais a montré sa puissance. En particulier, *caesar*, *caesar.adt*, *aldebaran* et *exhibitor* ont été utilisés de façon intensive avec succès. Néanmoins, ceux-ci étant limités par le phénomène d'explosion d'états, il incombe au concepteur de modifier sa spécification et d'utiliser au mieux ces outils de manière à contrôler ce phénomène.

Cette étude nous montre une fois de plus l'importance des méthodes formelles pour le développement et la mise au point de protocoles de communication. En effet, si ce protocole avait été formalisé et vérifié avant d'être implémenté dans certains appareils, l'impact de la correction des erreurs identifiées se serait limité à la seule spécification du système.

Remerciements

Nous tenons à remercier M. Celiktin et J. Roca d'Eurocontrol pour nous avoir soumis ce cas d'étude intéressant et pour les interactions fructueuses que nous avons pu avoir avec eux.

Références

[BB87] Tommaso Bolognesi and Ed Brinksma. Introduction to the ISO specification language LOTOS. *Computer Networks and ISDN Systems*, 14(1):25–29, January 1987.

[Eur96] Eurocontrol. Transition guidelines for initial air/ground data communications services. Technical report, EATCHIP Project, Eurocontrol, October 1996.

[Fer90] Jean-Claude Fernandez. An implementation of an efficient algorithm for bisimulation equivalence. *Science of Computer Programming*, 13(2–3):219–236, May 1990.

[FGM+91] Jean-Claude Fernandez, Hubert Garavel, Laurent Mounier, Anne Rasse, Carlos Rodríguez, and Joseph Sifakis. Une boîte à outils pour la vérification de programmes lotos. In Omar Rafiq, editor, *Actes du Colloque Francophone pour l'Ingénierie des Protocoles CFIP'91 (Pau, France)*, pages 479–500, Paris, September 1991. Hermès.

[Gar96] Hubert Garavel. An overview of the eucalyptus toolbox. In Z. Brezočnik and T. Kapus, editors, *Proceedings of the COST 247 International Workshop on Applied Formal Methods in System Design (Maribor, Slovenia)*, pages 76–88. University of Maribor, Slovenia, June 1996.

[GS90] Hubert Garavel and Joseph Sifakis. Compilation and verification of lotos specifications. In L. Logrippo, R. L. Probert, and H. Ural, editors, *Proceedings of the 10th International Symposium on Protocol Specification, Testing and Verification (Ottawa, Canada)*, pages 379–394. IFIP, North Holland, June 1990.

[GT93] Hubert Garavel and Philippe Turlier. Cæsar.adt : un compilateur pour les types abstraits algébriques du langage lotos. In Rachida Dssouli and Gregor v. Bochmann, editors, *Actes du Colloque Francophone pour l'Ingénierie des Protocoles CFIP'93 (Montréal, Canada)*, 1993.

[ISO88] ISO/IEC. Lotos — a formal description technique based on the temporal ordering of observational behaviour. International Standard 8807, International Organization for Standardization — Information Processing Systems — Open Systems Interconnection, Genève, September 1988.

[Mas93] Thierry Massart. A Collision Problem in OSI Standard Formal Specification. *Computer Networks and ISDN Systems*, 26:233–238, 1993.