

ERATO, un système expert d'aide au contrôle du trafic aérien

Marcel Leroux

Ecole Nationale de L'Aviation Civile

Jean-Marc Alliot

Centre d'Etudes de la Navigation Aérienne

Institut de Recherche en Informatique de Toulouse (laboratoire modélisation du raisonnement)

4 janvier 1991

Résumé

ERATO (En Route Air Traffic Organizer) est un système expert d'aide au contrôle du trafic aérien. Le but d'ERATO est de fournir au contrôleur des outils pour l'assister dans sa tâche, et non de réaliser une automatisation du contrôle. Afin d'être plus facile à intégrer, ERATO tente de modéliser au mieux le raisonnement du contrôleur pour lui fournir une aide directement en rapport avec ses méthodes de résolution.

ERATO est un projet développé conjointement par le Centre d'Etudes de la Navigation Aérienne (CENA) et l'Ecole Nationale de l'Aviation Civile (ENAC).

1 Introduction

1.1 Les buts

Depuis 1980, le trafic aérien connaît une progression de l'ordre de 10% par an, et il semble désormais acquis qu'il doublera dans les dix prochaines années. De plus, le nombre de données à traiter augmentera encore avec la mise en place de nouveaux moyens de communication entre le sol et les avions (FMS, DATA-LINK). Or, si une automatisation totale du contrôle est sans doute possible sur le long ou le très long terme, il semble irréaliste d'envisager une telle opération sur une aussi courte période. D'autre part, le système français de contrôle arrive dès aujourd'hui à saturation et les moyens classiques pour augmenter la capacité (redécoupage des secteurs ...) semblent avoir été exploités au maximum.

Il faut donc envisager le développement d'outils coopératifs permettant d'augmenter la capacité de traitement du contrôleur. Pour que ces outils puissent devenir effectivement opérationnels, il faut qu'ils soient efficaces, mais également acceptables. Par **efficaces**, nous entendons que ces outils doivent réaliser plus rapidement et mieux certaines tâches que le contrôleur assume aujourd'hui. Par **acceptables**, nous signifions qu'ils doivent s'intégrer dans le système actuel. En particulier, les choix, jugements et informations que ces outils fourniront devront être immédiatement compréhensibles par le contrôleur, qui reste l'élément fondamental de la boucle¹.

¹ Rappelons à ce sujet une anecdote rapporté par Daniel Michie ([Mic90]). Lors d'un tournoi d'échecs pour ordinateurs, un programme joua un coup tellement aberrant que ses programmeurs passèrent une nuit blanche à chercher le "bug" dans le programme, avant de se rendre compte qu'il avait en fait joué le coup le moins mauvais, grâce à une analyse tactique d'une profondeur inaccessible à un être humain. On peut en déduire

La seule solution raisonnable consiste donc dans la création d'outils respectant les modes opératoires du contrôleur. En effet, une première expérience ([eML87]) a montré la nécessité d'un modèle global du contrôleur. Un traitement dissocié des différentes tâches (coordination entrante, tenue de strips, détection de conflits, résolution de conflits, coordination sortante, ...) est difficile, voire impossible, car les différentes tâches sont étroitement imbriquées. D'autre part, le traitement partiel d'une tâche ne permet pas de développer un outil opérationnel: en effet, un tel outil est incapable de modéliser la charge de travail globale de l'opérateur humain, la complexité technique de la situation à gérer et ne saura donc pas adapter son mode de fonctionnement aux besoins effectifs du contrôleur.

ERATO a donc pour buts:

- D'analyser *en suivant une approche globale* les processus permettant au contrôleur de planifier son activité.
- De développer des outils *coopératifs* permettant de l'aider dans cette tâche de planification.

1.2 Problématique

L'espace aérien français est divisé en secteurs de contrôle qui sont des entités géographiques gérées par des équipes de deux ou trois contrôleurs. L'un de ces contrôleurs (le contrôleur radar) est en contact radio avec tous les avions se trouvant dans les limites du secteur. Il décide en temps réel des mesures à prendre pour qu'à aucun moment la séparation entre deux avions quelconques de son secteur ne tombe en dessous des normes requises. **Cependant, le contrôle du trafic aérien n'est pas une affaire de réflexes pour faire s'éviter les avions au dernier moment.** L'accumulation de telles décisions en de courts intervalles de temps peut créer des situations dangereuses et non-maitrisables.

L'autre contrôleur (opérateur de la fonction de planification ou contrôleur organique ou planificateur) est en contact avec tous les secteurs adjacents. Il organise le trafic qui va entrer dans son secteur en fonction de la situation à laquelle le contrôleur radar devra faire face. Il doit également négocier avec le secteur suivant les conditions de sortie de chaque avion, afin que le planificateur de ce secteur puisse à son tour vérifier la compatibilité avec le trafic de son secteur (coordination). Il doit également préparer le travail du contrôleur radar en mettant en évidence les paires d'avions susceptibles d'engendrer des conflits lors de la traversée du secteur.

Le contrôleur organique a donc une tâche importante et complexe, puisqu'il doit analyser et prévoir la situation future, ainsi que la charge de travail qu'elle induira sur le contrôleur radar. Les coordinations ayant lieu dix minutes avant l'entrée dans le secteur, le travail du planificateur consiste à prendre des décisions à partir de prévisions de conflits qui pourraient avoir lieu dans dix à vingt minutes. Le même contrôleur remplissant alternativement les fonctions de contrôle radar et organique, il utilise son savoir faire de contrôleur radar pour remplir la fonction de planification.

Les données sur lesquelles il opère sont floues, incertaines et dépendantes du temps. D'autre part, le modèle du contrôleur radar n'est pas déterministe: il existe plusieurs solutions possibles pour un conflit particulier, ce qui renforce considérablement les aspects flous et incertains du problème. Pour plus de détails sur le sujet, on peut se reporter à [Vil89].

qu'un programme, pour être pleinement acceptable, doit être capable de présenter ses résultats sous une forme directement accessible à l'opérateur humain.

2 Modélisation du contrôleur

2.1 Représentation de la connaissance

2.1.1 Les acquis en psychologie cognitive

Des études menées depuis 1973 ([BN73], [Bis79a], [BG73], [Bis79b], [Fal79]) montrent que le contrôleur réalise une “précompilation fonctionnelle”.

L’assimilation des données par le contrôleur n’est pas un simple exercice de mémoire. Par exemple, il transforme directement une distance de quarante nautiques pour un boeing 747 en cinq minutes de vol, alors que ces mêmes quarante nautiques sont compris comme onze minutes pour un Fokker 27. De même, des contrôleurs à qui on demande de dessiner de mémoire la carte de leur secteur, la représentent de façon déformée: il existe un lien direct entre la représentation de l’espace et les problèmes que le contrôleur à l’habitude d’y rencontrer.

ERATO s’appuie sur de semblables résultats pour construire son modèle du contrôleur organique.

2.1.2 Les logiques non-classiques

En raison de la complexité du problème (données floues, incertaines, dépendantes du temps), le calcul des prédicats ne pouvait suffire comme support à ERATO. ERATO utilise un certain nombre de propriétés de logiques non-classiques: logique temporelle, logique des défauts, logique possibiliste. Ces propriétés sont utilisées de façon purement opératoire². Examinons rapidement l’utilisation de chacun de ces outils:

Une logique temporelle floue: L’activité du contrôleur peut être décrite comme un processus mettant en oeuvre des tâches pouvant se produire dans des intervalles de temps définis de façon possibiliste, avec des butées impératives. Nous n’avons trouvé aucun modèle théorique correspondant au modèle du contrôleur tel que nous souhaitons le représenter. Le système que nous utilisons est un “croisement” entre la logique trivaluée de S. C. Kleene et une logique temporelle sur les intervalles du type de celle d’Allen où les bornes d’intervalle seraient floues. Nous allons examiner plus en détail les problèmes de recouvrement d’intervalles.

Tout évènement élémentaire E peut-être décrit par un intervalle temporel comprenant quatre (et non deux) valeurs: $[t_0, t_1, t_2, t_3]$. Les relations que nous avons entre les différentes valeurs sont: $t_0 < t_1 < t_3$ et $t_0 < t_2 < t_3$. E commence entre t_0 et t_1 , sans qu’il soit possible de savoir à quel instant précis. La valeur de vérité de E est u sur cet intervalle. Entre t_1 et t_2 si l’intervalle existe ($t_1 < t_2$), E a lieu de façon certaine: sa valeur de vérité est t . E se termine entre t_2 et t_3 . Sa valeur de vérité est u sur cet intervalle. En dehors de $[t_0, t_3]$, E est f . La proposition E se produit est t si l’intervalle $[t_1, t_2]$ n’est pas vide. Considérons maintenant deux évènements E et F que l’on peut décrire par $[t_0, t_1, t_2, t_3]$ et $[m_0, m_1, m_2, m_3]$. Nous allons essayer de déterminer la valeur de vérité des quatre propositions:

²Nous sommes parfaitement conscients de la vanité qu’il y aurait à essayer de démontrer la consistance de l’ensemble des axiomes logiques de base utilisés par ERATO. Au reste, Turner ([Tur86] fait déjà remarquer que même dans des cas plus simples et mieux formalisés, une telle opération peut être extrêmement difficile, voire impossible.

proposition	valeur	condition
E commence avant que F commence	t	$m_1 < t_0$
	f	$m_0 > t_1$
	u	$(m_0 \leq t_1) \wedge (m_1 \geq t_0)$
E finit avant que F finisse	t	$m_3 < t_2$
	f	$m_2 > t_3$
	u	$(m_2 \leq t_3) \wedge (m_3 \geq t_2)$
E commence avant que F finisse	t	$m_1 < t_2$
	f	$m_0 > t_3$
	u	$(m_0 \leq t_3) \wedge (m_1 \geq t_2)$
E finit avant que F commence	t	$m_3 < t_0$
	f	$m_2 > t_1$
	u	$(m_2 \leq t_1) \wedge (m_3 \geq t_0)$

Table 1: Valeurs de vérité

- E commence avant que F commence.
- E finit avant que F finisse.
- E commence avant que F finisse.
- E finit avant que F commence.

Elle dépend des positions relatives dans le temps de $t_0, t_1, t_2, t_3, m_0, m_1, m_2, m_3$. Nous pouvons calculer la table 1.

A partir de la table de ces quatre propositions, il est possible de calculer la valeur de vérité de propositions plus complexes, comme E et F se produisent simultanément, qui est équivalente à $((E \text{ commence avant que } F \text{ commence}) \wedge (F \text{ commence avant que } E \text{ finisse})) \vee ((F \text{ commence avant que } E \text{ commence}) \wedge (E \text{ commence avant que } F \text{ finisse}))$.

La valeur u correspondra, au moment de la détection de conflit, au diagnostic “doute” du contrôleur³.

La logique des défauts: La logique classique est inadéquate à modéliser correctement le raisonnement du contrôleur. En effet, la logique classique modélise des raisonnements de type monotone. Or le raisonnement du contrôleur est essentiellement non-monotone. Face à un problème, le contrôleur commence par considérer un certain nombre de paramètres comme prenant une valeur standard. S’il reçoit par exemple un Boeing-747 en coordination en montée, il va supposer que son taux de montée est compris entre 800 et 1500 ft/mn, et bâtir l’intégralité de ses déductions sur cette valeur. Or, si cette hypothèse se révèle inexacte (si, par exemple, le Boeing est très lourdement chargé ou s’il connaît un problème de moteur), il lui faudra reprendre l’ensemble de son raisonnement, ainsi que l’ensemble des résultats qui en ont découlé. La méthode de raisonnement utilisée n’est donc pas de type purement déductif.

³La logique de Lukasiewicz ne pouvait nous convenir. En effet, pour nous, deux propositions ayant l’une et l’autre la valeur u ne sont pas équivalentes.

La logique des défauts ([Rei87b], [Rei87a]) nous a semblé bien adaptée à ce type de mode opératoire et peut être considérée comme le coeur du raisonnement d'ERATO. Le SE ne considère que la plage de variation du paramètre en question (ici le profil de montée) pour laquelle la fonction de croyance de possibilité est 1 (ou proche de 1). Cette plage de variation est la seule prise en compte par le système pour la suite du raisonnement. Les autres plages sont prises en compte indirectement sous la forme de paramètres bloquants (au sein d'une logique des défauts semi-normale) qui relient les conclusions aux hypothèses faites. La simple vérification que les valeurs des paramètres bloquants restent à l'intérieur de la plage normale suffit à valider toutes les analyses et toutes les décisions qui auront été faites à partir des profils de montée par défaut. En revanche, la variation d'un de ces paramètres en dehors de la plage normale déclenchera une remise en cause complète du raisonnement tenu pour cet avion.

Les mécanismes de type Truth Maintenance System sont connus pour être lourds à gérer. Ici, la précompilation fonctionnelle du flot d'informations permet, comme nous allons le voir, de restreindre considérablement la taille de la base de connaissances à maintenir, rendant le système viable.

La théorie des possibilités: La théorie des possibilités (largement exposée dans [DP88]), nous permet d'associer à un événement incertain ou flou une fonction de croyance de possibilité. Il est ensuite possible de définir des opérations ensemblistes floues spécifiques afin de rendre compte de la façon dont l'opérateur combine les informations floues.

2.1.3 Pattern-matching

Plutôt que de travailler directement sur tous les paramètres relatifs à un conflit, ERATO utilise des techniques de pattern-matching permettant de réduire le nombre de classes de conflit. Cette méthode est principalement utilisée pour la détection et la résolution de conflit. Ainsi, trois caractéristiques discrètes (trajectoire horizontale relative (23 cas différents), vitesse relative, ordre de passage) suffisent pour décrire n'importe quel conflit pour la détection et l'analyse initiale.

La résolution demande un typage plus fin; il faut ajouter trois autres caractéristiques (la trajectoire horizontale relative affinée et les trajectoires verticales des deux avions), mais on peut cependant avec un nombre limité de règles décrire plus de 150000 conflits différents.

En reconnaissant et décrivant les conflits à partir d'un nombre limité de caractéristiques, nous pouvons développer simplement des raisonnements de type heuristique sans avoir à réaliser des calculs longs et compliqués, peu en rapport avec le mode de raisonnement du contrôleur.

2.2 Architecture d'ERATO

On peut considérer qu'ERATO se compose de quatre blocs principaux:

- La création du monde par défaut à partir des données initiales.
- La détection et l'analyse de conflits.
- La résolution de conflits.
- Les ajustements de trafic.

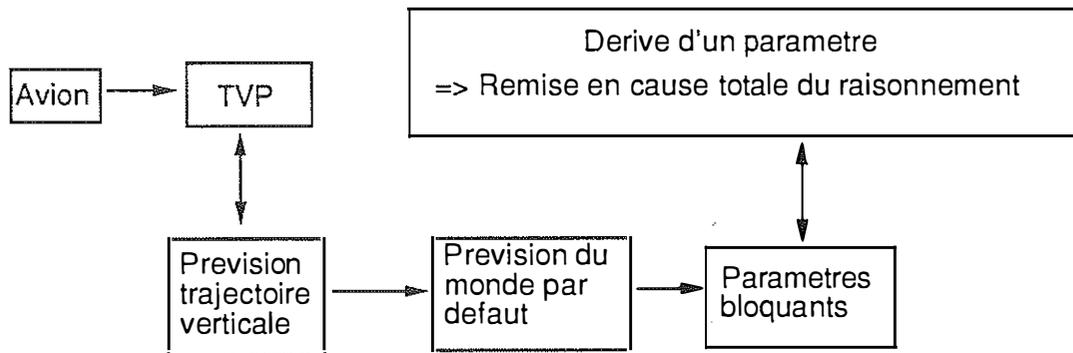


Figure 1: Création du monde par défaut

Nous allons détailler chacun de ces blocs.

2.2.1 La création du monde par défaut

Dans une première étape (figure 2.2.1, ERATO va créer le *monde par défaut* du contrôleur. Ce monde par défaut est basé sur les **TVP: Trajectoire Verticale Prévisible** et les *bloquants*, qui contrôlent la validité des TVP.

Pour chaque avion ERATO reçoit une proposition de coordination. Cette proposition contient l'ensemble des éléments du plan de vol:

- L'indicatif du vol
- Le type de l'avion
- La vitesse de l'avion
- L'aérodrome de départ
- L'aérodrome d'arrivée
- Le niveau plan de vol
- Le niveau effectif
- Le niveau autorisé
- La liste des balises
- La route plan de vol avec les estimés au passage de balise

C'est à partir de ces données qu'ERATO construit le TVP. Le TVP est une prévision heuristique de la trajectoire verticale des avions dans le secteur. *Il ne s'agit en aucun cas d'un calcul précis de trajectoire.* ERATO associe à chaque paire (ou triplet si la distance entre deux balises est non-signifiante) de balises consécutives un intervalle de niveaux de vol dans lequel se situera l'avion "selon toute vraisemblance". Ces niveaux de vol sont considérés comme occupés pendant l'intégralité du transfert entre les deux balises. Le TVP servira de base à la détection de conflit.

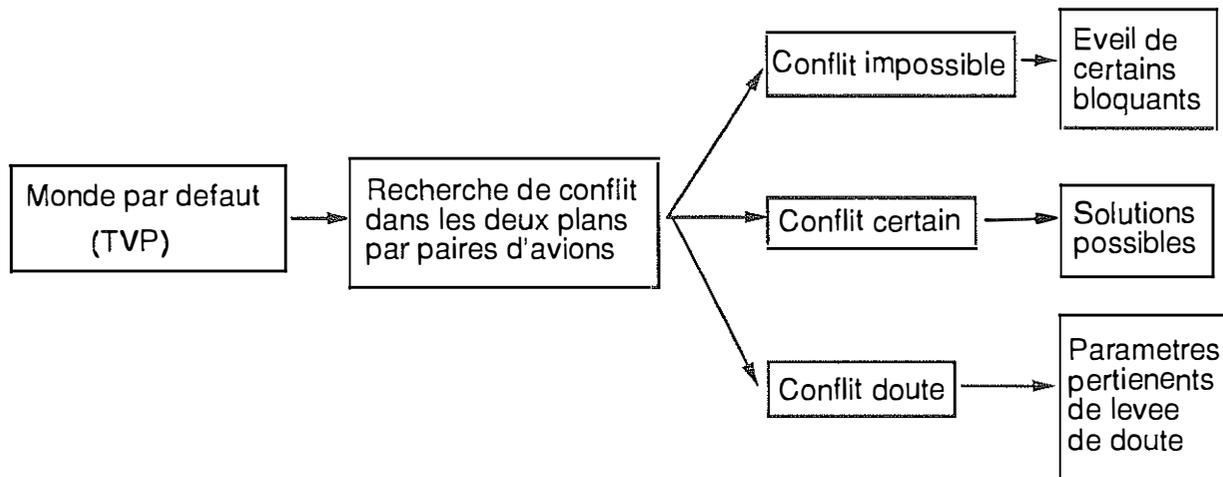


Figure 2: Détection et analyse de conflits

Les bloquants sont les paramètres que le contrôleur doit surveiller pour garantir que les prévisions du TVP restent valables. La liste des bloquants est inscrite dans un agenda que le système expert vérifie régulièrement.

Nous allons détailler ceci sur un exemple. Considérons un Airbus A300 que nous recevons au niveau 110 en montée vers le 290 suivant la route : (MAN, TEVX, LILAN, LILAR, BULOM, ANG). ERATO sait qu' un Airbus A300 monte en général à 2000 pieds/minute. Il construit donc le TVP suivant:

MAN,TEVX,LILAN : [110,290]

LILAN,LILAR,BULOM : [190,290]

BULOM,ANG : [290,290]

Quel est l'élément déterminant? C'est bien sur le taux de montée de l'avion. Si celui ci se révélait, par exemple, plus faible que le taux de montée prévue, il faudrait revoir l'occupation des niveaux sur le troisième segment. Le taux de montée est le bloquant de ce TVP. Ce taux de montée sera donc surveillé par le contrôleur. En cas de variation en dehors des plages normales du taux de montée, il faudra reconsidérer l'ensemble du raisonnement qui a été tenu pour cette avion.

2.2.2 Détection et analyse

Le second bloc réalise la detection et l'analyse de conflit (figure 2.2.2). Il faut immédiatement remarquer un point très important. ERATO réalise la détection de conflits sur chacune des *paires* d'avions présentes dans le système. Il s'agit d'une hypothèse fondamentale: pour ERATO, l'élément atomique de résolution est la paire d'avions. La détection de conflit se fait en deux étapes:

- Dans un premier temps, le système expert va comparer deux à deux les TVP des avions. Il va en alors en déduire quels sont les avions susceptibles d'interférer dans le plan vertical, dans quelle tranche de niveaux, et dans quel intervalle temporel.

- Dans un second temps, il va, pour chacune des interférences verticales constatées, faire une analyse des interférences horizontales possibles.

La recherche de l'interférence horizontale est relativement complexe. Nous disons qu'il y a interférence horizontale si, en cumulant les erreurs du système (imprécision des données, incertitude sur l'extrapolation) dans le sens le plus défavorable, la séparation minimum devient inférieure à la norme réglementaire.

Or, l'effet des erreurs du système dépend essentiellement de la trajectoire horizontale des deux avions (vingt-trois types différents), éventuellement de leur ordre de passage, et de leur vitesse relative. Ces trois paramètres constituent un premier typage du conflit et définissent une technique heuristique particulière de recherche de l'interférence horizontale. Cette technique fournit alors la plage d'interférence dans le plan horizontal. Il reste alors à rechercher si les deux types d'interférences peuvent se produire simultanément en utilisant les techniques décrites précédemment. Le diagnostic de conflit peut donc être de trois types:

Non-conflit: Le SE initialise alors une tâche de surveillance des bloquants qui garantissent ce diagnostic.

Conflit: Le SE active le module de résolution de conflits.

Doute: Le système expert dégage les *paramètres pertinents de levée de doute*. Le SE doit surveiller l'évolution de leur valeur afin de déterminer s'il y a ou non conflit. Il initialise donc une tâche de surveillance de ces paramètres mais également un plan de résolution du conflit potentiel.

L'ensemble des éléments nécessaires à la description des conflits pour un avion donné est regroupé dans l'*EAI (Ensemble des Avions Interférant)* avec cet avion. L'EAI comprend donc la description de l'*ensemble* des conflits certains ou potentiels relatifs à cet avion. Pour chaque conflit, on trouve dans l'EAI:

- La classification du conflit.
- Le caractère du conflit (certain ou doute).
- L'intervalle temporel où le conflit peut (ou va) se produire.
- La tranche de niveaux associés.
- La liste des balises décrivant la portion de la route où le conflit peut (ou va) se produire.

2.2.3 Résolution

La résolution de conflits soulève différents problèmes: il faut tout d'abord savoir résoudre les conflits élémentaires (qui ne font intervenir que deux avions), puis les conflits à plus de deux avions, et enfin assurer la cohérence de la base de connaissance, en recalculant un certain nombre de facteurs, *si et seulement si cela est nécessaire*, qui peuvent avoir été modifiés par la résolution d'un conflit (par exemple, un avion que l'on a dévié de sa trajectoire peut entrer en interférence avec un avion avec lequel il n'était pas en conflit jusque là).

Conflits élémentaires: Pour pouvoir effectuer une résolution du conflit, le SE doit tout d'abord affiner le typage dégagé lors de l'étape précédente. Pour ce faire, il utilise deux caractéristiques supplémentaires (les trajectoires verticales de chacun des avions) et il affine la description de la trajectoire horizontale. Il peut alors déterminer la gravité du conflit. La gravité d'un conflit est liée au type de "solution" envisagée: il peut s'agir d'un guidage radar, d'une surveillance (cas de doute). Parfois, l'existence de l'interférence ne fait que créer une contrainte au cas où un des deux avions devrait subir un guidage radar pour résoudre un autre conflit. Si le conflit détecté nécessite (ou risque de nécessiter) un guidage radar, cette phase détermine, à partir d'heuristiques liées au typage précédent, le sens des altérations de cap conformes aux règles de l'art, et au savoir-faire du contrôleur, leur plage d'initialisation, et l'intervalle de temps pendant lequel le contrôleur radar sera mobilisé par la résolution de ce conflit. La signature du conflit rend compte du non-déterminisme du modèle du contrôleur en mettant en évidence tous les types de solution radar possibles pour le résoudre.

Conflits complexes: Le conflit élémentaire est un conflit théorique, isolé de son contexte, c'est à dire des contraintes imposées par le reste du trafic. Ces contraintes peuvent rendre impossible une ou plusieurs solutions théoriques. Il s'agit donc dans cette phase de détecter les cas de conflits complexes (recouvrement des plages temporelles et spatiales de résolution de deux ou plusieurs conflits élémentaires) et de trouver la ou les solutions compatibles avec l'ensemble des conflits élémentaires.

Dans certains cas, il peut même se trouver qu'il n'existe aucune solution commune compatible avec l'ensemble des conflits. Ici, l'opérateur doit choisir un des conflits (le plus pénalisant) et le supprimer (il peut choisir de faire monter ou descendre un avion, ou même refuser un avion lors de la coordination s'il estime qu'il s'agit de la seule action raisonnable).

Interaction des solutions avec le système existant: Un problème sérieux se pose: comment garantir que la solution d'un conflit ne va pas engendrer un ou plusieurs nouveaux conflits? Faut-il chaque fois que l'on résout un conflit reprendre l'intégralité du raisonnement? Un des éléments fondamentaux qui garantissent un fonctionnement raisonnable du mécanisme de TMS est la ZIP: *Zone d'Interférence Potentielle*. La ZIP découle d'une notion utilisée par les contrôleurs lorsqu'ils font du contrôle aux procédures. Comme le contrôleur organique ne peut, dix ou quinze minutes à l'avance prévoir le comportement du contrôleur radar, il construit pour chaque avion une enveloppe de trajectoires, la ZIP⁴. Tout avion se trouvant dans cette enveloppe est ajouté dans l'EAI. Ainsi, l'EAI contient non seulement l'ensemble des avions conflictuels, mais aussi les avions pouvant générer des contraintes ou des conflits en cas de résolution d'un autre conflit. Ainsi, considérons l'exemple où l'avion AV1 est en conflit certain avec AV2, avec de plus un avion AV3 présent dans l'EAI d'AV1 qui crée une contrainte de résolution. Le problème est traité comme un conflit multiple entre AV1, AV2 et AV3. La solution de ce conflit multiple ne génèrera *de façon certaine*, aucun autre conflit avec quelque avion que ce soit.

⁴La construction de la ZIP est en fait un peu plus évoluée que cette phrase lapidaire ne le laisse croire.

2.2.4 Charge de travail et ajustements de trafic

Nous pensons que la représentation de la charge de travail prévisible du contrôleur radar est très proche d'une charge de tâches.

Sous cette hypothèse (qui s'appuie sur l'expérience), notre but dans cette phase est d'énoncer les heuristiques permettant à l'opérateur de la fonction de planification de prévoir les situations dans lesquelles l'accumulation des tâches dépasse la capacité de traitement du contrôleur radar. Certains paramètres de la signature des conflits seront directement utilisés par ces heuristiques (plage de début de résolution de conflit, fonction de croyance de possibilité de la manoeuvre pour les résolutions de conflit), certains paramètres doivent être affinés. Ainsi, dans le cas de la surveillance radar, il faut estimer la gravité du conflit potentiel: il semble plus lourd d'avoir à surveiller deux avions dont l'écart minimum sera "probablement" 20Nm que si l'écart supposé est plutôt de 40Nm.

Cette phase est encore en cours de développement.

3 Les outils coopératifs

Comme nous l'avons dit en introduction, le but d'ERATO est de construire un modèle du contrôleur permettant de développer des outils efficaces et acceptables. Rappelons que nous ne voulons en aucun cas réaliser une automatisation complète du contrôle, ni même automatiser les processus de décision, mais bien fournir à l'opérateur des outils lui permettant d'améliorer sa capacité de travail.

A partir de l'analyse de ce modèle, nous pouvons dès aujourd'hui envisager la réalisation des outils suivants:

Filtrage de l'information: L'information est actuellement présentée au contrôleur de façon systématique, indépendamment du contexte. Or, l'opérateur n'utilise qu'une faible partie de ces informations à un instant donné. La connaissance des processus de raisonnement du contrôleur doit permettre un affichage fonctionnel des différents paramètres.

Filtrage du trafic: Comme nous le disions en introduction, le contrôleur organique doit connaître parfaitement les modes opératoires du contrôleur radar, pour écouler au mieux le trafic. Lorsque le contrôleur organique détecte une situation trop complexe à gérer pour le contrôleur radar, il doit modifier les niveaux d'entrée pour un ou plusieurs avions. Ainsi, le contrôleur organique doit concilier deux impératifs contradictoires: opérer un nombre suffisant de modifications afin de conserver une situation gérable par le contrôleur radar (sécurité), et pénaliser le moins possible les avions pour assurer un bon écoulement du trafic.

Avec le modèle du contrôleur radar contenu dans ERATO, il est possible de concevoir des outils d'aide à la décision pour le contrôleur organique:

- Un vérificateur permettant de tester la compatibilité d'une configuration de sortie du secteur pour un groupe d'avions donné avec la charge de travail et le savoir-faire du contrôleur radar.
- Un organisateur pouvant proposer (à la demande) au contrôleur des solutions planning.

Surveillance de paramètres bloquants: La surveillance des paramètres bloquants est une tâche fastidieuse qui accapare une large part de l'énergie du contrôleur, alors que dans la plupart des cas, ces paramètres resteront dans la plage normale d'utilisation. Il semble donc judicieux de faire assurer ce travail par un outil spécifique, lorsque le contrôleur en exprime le désir. Cet outil déclenchera une alarme si la valeur d'un de ces paramètres s'éloigne de l'intervalle par défaut.

Surveillance de paramètres pertinents: De la surveillance des paramètres pertinents dépend la levée du doute sur le diagnostic d'un conflit ou sur la résolution à y apporter. Comme pour la surveillance des paramètres bloquants, il s'agit d'une tâche contraignante. Lorsque le contrôleur estime ne plus être en mesure de l'assurer convenablement en raison d'une charge de travail trop importante, il préfère supprimer le conflit potentiel, pénalisant ainsi inutilement le trafic.

Ce système surveillera à la demande du contrôleur les paramètres pertinents et rendra son diagnostic avant le début de la plage normale de résolution du conflit afin de permettre au contrôleur de prendre une décision en temps normal.

Simulation de résolution: Dans le cas de situations très complexes, le contrôleur a besoin de planifier très tôt la résolution du conflit; afin de l'aider dans cette tâche, un simulateur de résolution permettra de visualiser les avions en conflit, les avions susceptibles d'interférer et de simuler les différentes solutions proposées par le contrôleur.

Exécution différée de solutions: L'accumulation de conflits à résoudre simultanément oblige le contrôleur à en supprimer certains, pénalisant ainsi le trafic. A l'aide de cet outil, il pourra programmer l'exécution différée d'un ensemble de manoeuvres résolvant un conflit; le logiciel s'assurera de la compatibilité de cette solution avec l'environnement courant.

Résolution d'un conflit: Le contrôleur pourra demander au résolveur de prendre totalement en charge un conflit donné. Le résolveur calculera et exécutera une solution compatible avec les règles de l'art et le savoir-faire du contrôleur.

L'agenda: Le but de l'agenda est de fournir au contrôleur une représentation visuelle de l'urgence quant à l'exécution de tâches impératives (exécution d'une résolution).

4 Etat d'avancement

Le projet ERATO entre dans sa quatrième année. Il est le résultat de la collaboration entre contrôleurs et ingénieurs. Développé sur stations de travail SUN en Quintus-Prolog, il comprend environ 3000 règles. Les blocs 1 et 2 (construction du monde par défaut et détection de conflit) sont terminés. Le bloc 3 est largement avancé et la conception du bloc 4 se poursuit actuellement. La validation des hypothèses psychologiques qui sont à la base d'ERATO (représentation anisotrope de l'espace, existence d'une représentation par défaut des trajectoires verticales, mécanisme de surveillance de paramètres bloquants et de paramètres de levée de doute, ...) est en cours, en collaboration avec l'Université de Toulouse-Mirail (laboratoire de personnalisation et changements sociaux, équipe de psychologie du travail), et devrait s'achever en Avril 91. Diverses interfaces graphiques ont été développées ([Tha90], [LL90]) afin de faciliter les expérimentations et la validation du système.

Le projet semble avoir aujourd'hui atteint une maturité suffisante pour être présenté en dehors du monde de la Direction Générale de l'Aviation Civile.

Bibliographie

- [BG73] A. Bisseret et Y. Girard. Le traitement des informations par le contrôleur du trafic aérien. Rapport technique, IRLA CO 7303 R37, 1973.
- [Bis79a] A. Bisseret. Les informations utilisables pour la résolution de conflit. Rapport technique, IRIA CO 7301 R36, 1979.
- [Bis79b] A. Bisseret. Utilisation de la théorie de la détection de signal pour l'étude des décisions opératives: effets de l'expérience des opérateurs. Rapport technique, IRIA CO 79011 R60, 1979.
- [BN73] A. Bisseret et J. Nobel. Estimation de la fréquence des conflits sur trois secteurs de contrôle du centre de contrôle régional nord. Rapport technique, IRIA CO 7301 R34, 1973.
- [DP88] Didier Dubois et Henry Prade. *Théorie des possibilités*. Masson, 1988.
- [eML87] Philippe Genthon et Marcel Leroux. Un système expert pour l'aide du trafic aérien. Rapport technique, Ecole Nationale de l'Aviation Civile, Janvier 1987.
- [Fal79] P. Falzon. Première expérimentation de la méthode d'apprentissage par exercices auto-programmés. Rapport technique, ENAC, 1979.
- [LL90] Rémy Larice et Ling Wee Lee. Interface homme-machine pour la validation du système expert erato. Rapport technique, Ecole Nationale de l'Aviation Civile, 1990.
- [Mic90] Donald Michie. *Réflexions sur l'Intelligence des machines 25 ans de recherche*. Masson, 1990.
- [Rei87a] R. Reiter. Non monotonic reasoning. *Annual Review of Computer Science*, 1987.
- [Rei87b] R. Reiter. *Readings in knowledge representation*, chapitre On reasoning by defaults. Morgan Kaufman, 1987.
- [Tha90] Daniel Thaysse. Présentation graphique des résultats du système expert erato. Rapport technique, CNAM, 1990.
- [Tur86] Raymond Turner. *Logiques pour l'Intelligence Artificielle*. Masson, 1986.
- [Vil89] Jacques Villiers. Le mur de la capacité. Rapport technique, DGAC, Décembre 1989.