# RÉALISATION LOGICIELLE DES SYSTÈMES MULTIMODAUX : UN MOTEUR DE FUSION GÉNÉRIQUE

**Laurence Nigay**
**Laboratoire de Génie Informatique-Institut IMAG**
**BP 53, 38041 Grenoble Cedex 9**
**Tél. 76 51 44 40, E-mail : Laurence.Nigay@imag.fr**

**Résumé :** Cet article a trait à la réalisation logicielle des systèmes multimodaux. Pour ce faire, nous partons de propriétés qui caractérisent l'usage des modalités : la complémentarité, l'assignation, la redondance et l'équivalence des modalités. Ces propriétés constituent un canevas remarquable pour la conception logicielle. Nous proposons un modèle d'architecture PAC-Amodeus en symbiose avec un moteur de fusion pour la réalisation logicielle de ces propriétés. Grâce à sa généricité et en tant que service "d'utilité publique" pour les systèmes multimodaux, le moteur de fusion est un bon candidat comme composant réutilisable d'un squelette d'application.

**Mots clés :** Modalité, utilisabilité, modèle d'architecture logicielle, fusion, parallélisme.

## 1. Introduction

Dans cet article, nous considérons la réalisation logicielle des systèmes multimodaux qui offrent plusieurs techniques d'interaction innovantes à l'utilisateur. Notre approche se fonde sur les types de coopération entre modalités exposés par J.C. Martin [MARTIN 94]. Nous définissons ces quatre types de coopération comme des propriétés que nous notons propriétés CARE. Les propriétés CARE (Complémentarité, Assignation, Redondance, Equivalence) décrivent les possibilités offertes à l'utilisateur et donc l'usage possible des modalités. L'analyse et les incidences des propriétés CARE sur l'utilisabilité d'un système sont développées dans [NIGAY 94]. Nous définissons formellement ces propriétés dans [COUTAZ 94].

Il convient ici d'apprécier ces propriétés comme des éléments directeurs en conception logicielle d'interfaces. Nous considérons les implications des propriétés CARE, et en particulier la complémentarité et la redondance, sur la réalisation logicielle. Schématiquement :

- La complémentarité implique l'usage combiné de plusieurs modalités pour spécifier une commande ou une partie d'une commande. Notons qu'une commande est le terme usuel qui désigne une procédure atomique que le système fournit à l'utilisateur pour accomplir une tâche élémentaire.
- La redondance autorise l'utilisateur à spécifier plusieurs fois une même commande (ou une même partie de commande), simultanément selon plusieurs modalités.

Nous proposons d'intégrer la complémentarité et la redondance au sein d'un moteur de fusion. Ce moteur générique opère en corrélation avec notre modèle d'architecture PAC-Amodeus. L'objet de cet article est de présenter le fonctionnement de ce moteur et le point d'ancrage dans PAC-Amodeus.

Nous avons appliqué PAC-Amodeus à la réalisation de plusieurs systèmes multimodaux et en particulier MATIS. MATIS est un système d'information sur les voyages aériens, développé sur machine NeXT avec le système Sphinx. Ce système autorise des énoncés de commandes tels que la phrase dite <I would like a Usair flight from this city to this city> conjuguée à deux sélections-souris pour spécifier les villes de départ et d'arrivée. MATIS est utilisé pour étayer notre discussion.

## 2. Le modèle PAC-Amodeus

Comme le montre la figure 1, PAC-Amodeus [NIGAY 93] est un modèle d'architecture logicielle préconisant cinq composants logiciels. Le Contrôleur de Dialogue (CD), clé de voûte du système interactif, prend en charge l'enchaînement des tâches et gère chaque fil de dialogue au moyen d'une hiérarchie d'agents PAC [COUTAZ 87].

Chaque niveau dans la hiérarchie d'agents PAC correspond à des niveaux d'intervention dans les transformations successives de l'information (processus de fusion et de fission). La fusion dite syntaxique [NIGAY 94] revient au Contrôleur de Dialogue. Elle consiste à combiner des unités informationnelles pour obtenir la description complète d'une commande comme la spécification d'une requête dans le système MATIS. L'organisation non séquentielle et hiérarchique du Contrôleur de Dialogue est un support à la réalisation de la fusion. Les données sont combinées en parallèle et de façon incrémentale à chaque niveau dans la hiérarchie d'agents. L'exemple d'un éditeur graphique supportant la commande multimodale "Mets cà là" est détaillé dans [NIGAY 93]. Cet exemple montre l'intérêt de la hiérarchie d'agents PAC.

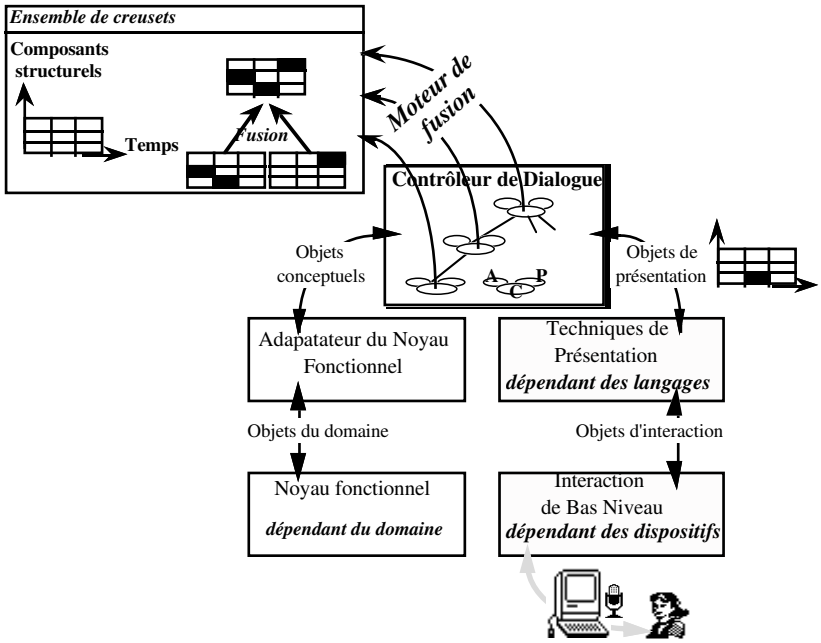


**Figure 1 :** Le modèle PAC-Amodeus et ses liens avec le moteur de fusion.

Nous décrivons maintenant la réalisation logicielle du processus de fusion des informations. Pour cela nous présentons notre moteur de fusion selon la grille d'analyse élaborée lors des journées de travail d'IHM'92 [IHM 92] puis nous allons dans les détails en expliquant les règles régissant le comportement du moteur.

## 3. Présentation du moteur de fusion

La fusion s'appuie sur une représentation uniforme : le creuset. Le creuset sert de représentation commune aux unités informationnelles émises par le Composant Techniques de Présentation et aux unités informationnelles qui résultent du processus de fusion. Ainsi de la fusion de deux creusets, résulte un creuset. Comme le montre la figure 2, un *creuset* est un objet structuré à deux dimensions. L'un des axes définit les composants structurels que le Contrôleur de Dialogue et l'Adaptateur du Noyau Fonctionnel savent reconnaître. Par exemple, chaque constituant de la signature d'une commande occupe un emplacement sur l'axe vertical "composants structurels" du creuset. L'axe des temps identifie les dates de spécification par l'utilisateur des unités informationnelles incluses dans le creuset. La présence de plusieurs valeurs pour un même composant peut révéler une redondance ou des incohérences. Le creuset doit être rempli à l'intérieur d'une fenêtre temporelle (appelée aussi "épaisseur du présent"), sinon il est abandonné.

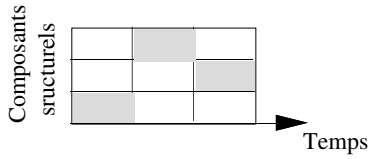


**Figure 2** : Le creuset comme structure de représentation commune pour la fusion.

L'organisation non séquentielle du Contrôleur de Dialogue autorise des fusions en parallèle. Son organisation hiérarchique permet une fusion incrémentale et contextuelle. La figure 1 désigne le point d'ancrage du moteur de fusion dans PAC-Amodeus. On constate que la partie Contrôle de chaque agent fait appel au moteur. Chaque agent a la possibilité d'ajouter des unités informationnelles issues de son état interne qu'il souhaite soumettre au moteur comme candidat à la fusion. Dans une implémentation à objet selon laquelle un agent serait réalisé par un exemplaire d'objet, le moteur serait une méthode héritée par toutes les classes d'agent. Le mécanisme de fusion serait alors commun mais exécuté de manière répartie par une société d'agents.

Une sous-hiérarchie d'agents PAC correspond, à une tâche ou à une grappe de tâches. Dans notre analyse du mécanisme de fusion, une structure hiérarchique permet d'organiser l'espace des creusets en sous-ensembles ou contextes. Ce procédé permet d'accroître l'efficacité du moteur de fusion : deux creusets qui n'appartiennent pas au même ensemble ne peuvent être fusionnés. Les creusets de deux ensembles distincts peuvent avoir ou non un nombre différent de composants structurels.

Par leur Présentation, chaque agent feuille de la hiérarchie, reçoit du Composant Techniques de Présentation (CTP) les creusets pour lesquels ils ont exprimé leur intérêt. Chaque case d'un creuset peut être déclarée intéressante par un ou plusieurs agents. Un creuset émis par le CTP peut être complet ou incomplet. L'agent traite localement l'information reçue : même si le creuset est incomplet, il peut cependant spécifier un retour d'information qui est alors partiel.

Si le creuset reçu par un agent feuille est incomplet ou s'il est complet mais intéresse l'agent père, l'agent feuille le transmet au père. Ce dernier fait appel au moteur de fusion en lui fournissant comme paramètre le creuset nouvellement reçu. Si aucune fusion n'a été effectuée, le creuset incomplet est renvoyé à l'agent. Mais si le creuset est fusionné, le creuset résultant est envoyé à l'agent le plus bas dans la hiérarchie et père des agents responsables des cases remplies du creuset.

La stratégie du moteur de fusion est précoce (*"eager"*). Le moteur ne se met jamais en attente d'une information complémentaire. L'avantage est la possibilité de produire des retours d'informations lexicaux et syntaxiques immédiats. L'inconvénient est le risque d'avoir à défaire des fusions incorrectes. Le creuset, formalisme unique de représentation, facilite cette activité. Le moteur élabore une solution par fusions successives : sa stratégie est donc progressive et précoce. Il opère en profondeur d'abord avec retour arrière en cas de fusion incorrecte. La fusion est effectuée sur la base de trois critères : la structure des objets à combiner, le temps, le contexte. A chacun de ces critères, correspond un traitement spécifique de fusion que nous appelons respectivement *microfusion*, *macrofusion* et *fusion contextuelle*. En complément, il convient de considérer l'anticritère suivant : si les structures logiques des unités informationnelles sont incompatibles, leur fusion est impossible. La complémentarité structurelle est donc une condition nécessaire mais non suffisante à la fusion. Nous présentons ces trois conditions de fusion par ordre de priorité décroissante :

- Le *critère de microfusion* s'appuie sur le temps et combine des informations, structurellement complémentaires ayant des intervalles de temps qui s'entrelacent (parallélisme ou pseudo-parallélisme) ;
- le *critère de macrofusion* s'appuie sur le temps et combine des informations structurellement complémentaires qui appartiennent à une fenêtre temporelle (épaisseur du présent ou proximité temporelle) ;

• le *critère de fusion contextuelle* utilise uniquement le contexte et fusionne des informations structurellement complémentaires. Dans le système MATIS par exemple, la requête la plus récente en cours de construction est une information maintenue par le contexte et est utilisée par le moteur comme critère d'intégration.

A titre d'illustration, la figure 3 montre comment le moteur de fusion vérifie les propriétés de complémentarité et de redondance au sein du système MATIS.

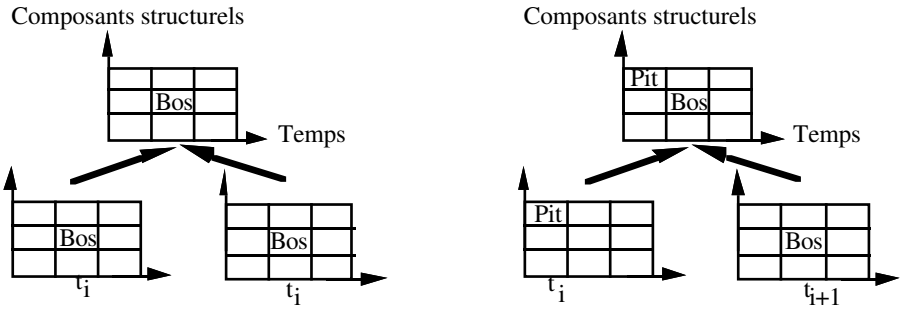


**Figure 3 :** La redondance et la complémentarité au sein du moteur de fusion.

## 4. Le mécanisme de fusion et ses règles

Dans [NIGAY 94], nous formalisons les règles qui régissent le comportement de notre moteur. Nous présentons ici les règles correspondant à la microfusion. Pour cela, nous introduisons tout d'abord les métriques associés à un creuset :

- $c_i = (p_1, p_2, ..., p_j, ..., p_n)$ : le creuset i constitué de n composants structurels.
- $info_{ij}$ : l'unité informationnelle rangée dans le composant j du creuset i.
- $Tinfo_{ij}$ : la date de l'unité informationnelle rangée dans le composant j du creuset i.
- $Tmax_i$ : la date de la case la plus récente de creuset $c_i$.
- $Tmin_i$ : la date de la case la plus ancienne du creuset $c_i$.
- $Fen_temp_i$ : la fenêtre temporelle du creuset $c_i$. $Fen_temp_i = [Tmin_i - \Delta t , Tmax_i + \Delta t]$
- Tempo : la temporisation du creuset $c_i$ déclenchée à sa complétude.

Lorsqu'un creuset est nouvellement ajouté à la liste des candidats, la première étape du moteur consiste à le combiner en appliquant le critère de *microfusion*. La microfusion considère la date des colonnes d'un creuset. Une colonne correspond au contenu structurel d'une unité informationnelle. Par exemple dans MATIS l'énoncé de la phrase *"USAir flights from Pittsburgh"* à l'instant t, conduit à la création du creuset à une seule colonne (de date t) contenant les valeurs PIT et USAir sur des lignes distinctes.

Le moteur recherche si des colonnes complémentaires ont des dates qui sont très proches à Δmicrot près (Δmicrot << Δt) de la date du nouveau creuset. Dans le cas positif, les informations sont fusionnées, entraînant parfois une dé-fusion. La microfusion garantit que des informations très proches dans le temps soient combinées. Nous schématisons la microfusion à la figure 4.

Règle **La microfusion**

Soient :

- $col_{it} = (p1, p2, ..., pj, ..., pn)$ :
  une colonne à l'instant t du creuset $c_i$ de composants structurels p1, ..., pn.
- $col_{jt'} = (p'1, p'2, ..., p'j, ..., p'n)$
  une colonne à l'instant t' du creuset $c_j$ de composants structurels p'1, ...,p'n.
- $i \neq j$

$col_{it}$ et $col_{jt'}$ sont combinées :

- si elles sont complémentaires. Complémentaire($col_{it}$ , $col_{jt'}$ ) est défini par :

$$\forall k \in [1..n] : \exists info_{ik} \ \Lambda \ \neg (\exists \ info_{jk})$$

• et si leurs dates sont très proches à Δmicrot près. Proche($col_t$, $col_{t'}$) est défini par :

$t' \in [t-\Delta microt, t+\Delta microt]$

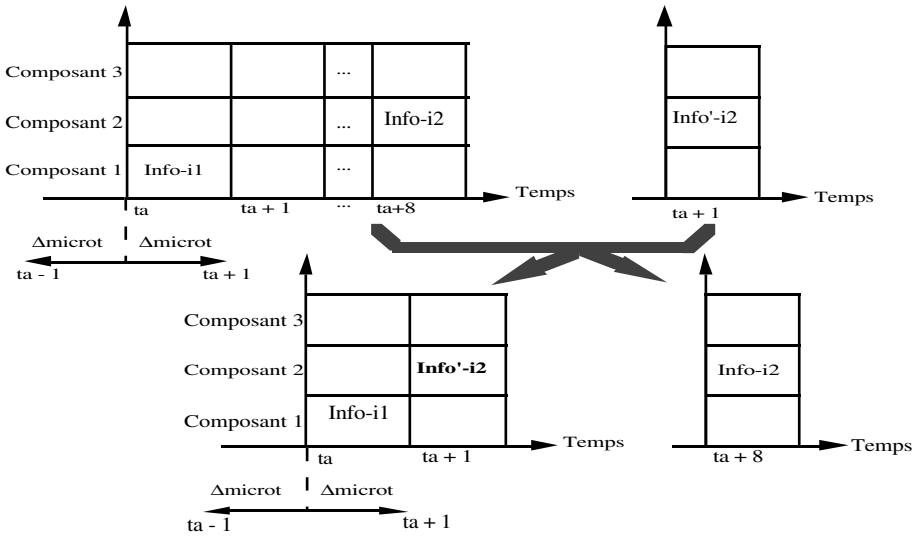


**Figure 4 :** Microfusion avec dé-fusion. (Δmicrot = 1 unité de temps)

La redondance est analysée en pré-traitement de la microfusion. Avant de décider s'il y a lieu de pratiquer une fusion, le moteur vérifie que les informations ne sont pas redondantes. Si c'est le cas le nouveau creuset est ignoré.

Règle **Redondance détectée avant la microfusion**

Soient

• $col_{it}$ = (p1, p2, ..., pj, ..., pn) :
une colonne à l'instant t du creuset $c_i$ de composants structurels p1, ..., pn.

• $col_{jt'}$ = (p'1, p'2, ..., p'j, ..., p'n)
une colonne à l'instant t' du creuset $c_j$ de composants structurels p'1, ..., p'n.

• $i \neq j$

$col_{it}$ et $col_{jt'}$ sont déclarées redondantes :

• si elles contiennent les mêmes informations pour les mêmes composants structurels. Redondant($col_t$, $col_{t'}$) est défini par :

$$\forall k \in [1..n] : \exists info_{ik} \ \Lambda \ \exists info_{jk} \ \Lambda \ info_{ik}=info_{jk}$$

$$\Lambda \quad \forall k' \in [1..n] : \neg(\exists info_{ik'}) \ \Lambda \ \neg(\exists info_{jk'})$$

• et si leurs dates sont très proches à Δmicrot près. Proche($col_t$, $col_{t'}$) est défini par :

$t' \in [t-\Delta microt, t+\Delta microt]$

## 5. Conclusions et extensions envisagées

Nous avons proposé un modèle d'architecture, PAC-Amodeus, qui, en symbiose avec notre moteur de fusion, permet de supporter les propriétés CARE. Nous avons ainsi tenté de jeter un pont au-dessus du gouffre qui existe entre les propriétés d'utilisabilité des systèmes et leurs réalisations logicielles. Notre contribution à la réalisation logicielle des systèmes est multiple :

• nous proposons un modèle d'architecture pour la réalisation logicielle des systèmes multimodaux,

• nous avons intégré dans un modèle d'architecture les phénomènes de fusion et de parallélisme, deux caractéristiques essentielles des systèmes multimodaux,

• nous proposons un mécanisme de fusion indépendant du domaine de l'application du système.

Nous souhaitons souligner le fait que notre moteur de fusion est générique : il ne dépend pas d'un système particulier. Ses critères de fusion sont la proximité temporelle et la complémentarité structurelle. Les objets ou unités informationnelles qu'il manipule, notés creuset, sont génériques : un creuset correspond à un ensemble de cases à compléter par fusion. La sémantique attachée à chaque case n'est pas connue du moteur. Seule la composition structurelle doit être déclarée de manière externe au moteur. Cette technique, qui permet au moteur d'engendrer des formats de creuset à façon sans en modifier les traitements, permet d'envisager l'intégration du moteur dans un squelette d'application pour systèmes multimodaux. Nous avons organisé les traitements du moteur en explicitant trois types de fusion de priorités distinctes. La *microfusion* permet un usage parallèle et combiné des modalités ; la *macrofusion* autorise l'usage séquentiel néanmoins proche dans le temps (fenêtre temporelle) et combiné des modalités ; la *fusion contextuelle* permet d'ignorer la notion de temps et de prendre en compte les changements de fils de dialogue. Nous avons montré l'apport du modèle PAC-Amodeus et notamment celui de la hiérarchie d'agents PAC du Contrôleur de Dialogue comme support à la fusion. Les services du moteur de fusion sont accessibles par la partie Contrôle de chaque agent PAC. La hiérarchie d'agents permet de structurer l'espace des creusets et par conséquent d'augmenter l'efficacité du moteur. L'organisation hiérarchique du Contrôleur de Dialogue permet des retours d'information immédiats et partiels dans le cas de fusions successives. Grâce à sa généricité et en tant que service "d'utilité publique" pour les systèmes multimodaux, le moteur de fusion est un bon candidat comme composant réutilisable d'un squelette d'application.

Nous envisageons plusieurs extensions possibles que nous proposons de discuter lors des journées IHM'94 dans l'atelier "Interfaces Multimodales". Une première classe d'extensions consiste à introduire de nouveaux paramètres qui permettraient d'agir de manière générique sur le comportement du moteur. Un premier paramètre est la largeur de la fenêtre temporelle. De la même manière que l'utilisateur du Macintosh définit la vitesse du double-clic (Très lent, Lent ou Rapide), nous proposons de définir la largeur de la fenêtre temporelle qui pourrait être calculée dynamiquement par le système en fonction du contexte. Un deuxième paramètre envisagé permettrait de spécifier les types de fusions possibles. Par exemple, un système qui n'autorise que l'usage synergique des modalités ne requiert pas la *microfusion*.

Outre l'ajout de paramètres, il nous semble crucial d'associer un facteur de confiance à chaque case de creuset. Cette notion a été introduite de manière ad hoc pour le système MATIS. Il faudrait le généraliser. Typiquement lorsque l'utilisateur sélectionne une ville dans une fenêtre, le Composant Techniques de Présentation crée un creuset qui contient deux fois le nom de la ville dans des cases différentes. En effet le CTP ne peut décider s'il s'agit d'une ville de départ ou d'arrivée. Les cases contenant les noms de ville ont alors un facteur de confiance très faible, qui permet au moteur de fusionner le creuset en ignorant l'une des cases. Le creuset de droite de la figure 5 illustre le cas. Les facteurs de confiance modélisent l'incertitude des informations dans chaque case. Ils permettent aussi de spécifier un ordre de remplissage des cases d'un creuset, très utile à la résolution des coréférences. Par exemple l'utilisateur de MATIS articule la phrase *"Flights from Boston to this city"* puis sélectionne une ville avec la souris. Comme le montre la figure 5, le CTP crée un creuset (à gauche) correspondant à la phrase dont l'une des cases est vide avec un haut facteur de confiance. Le moteur de fusion sera alors capable de fusionner les deux creusets en s'appuyant sur les facteurs de confiance. Un très haut facteur de confiance dans une case non remplie, fonctionne comme un aimant et va influencer le moteur à fusionner des informations dans cette case.

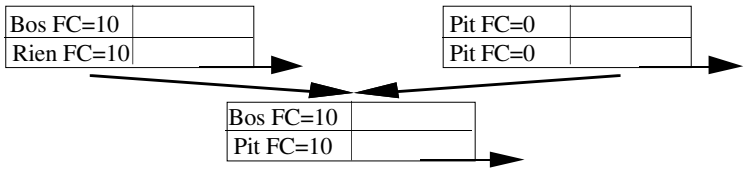

**Figure 5 :** Facteur de confiance associé à chaque case de creuset.

## 6. Remerciements

Ce travail a été partiellement financé par le projet ESPRIT BRA AMODEUS2 7040 et le PRC CHM, pôle IHM multimodales.

## 7. Références

[COUTAZ 87]
J. Coutaz, PAC: an Implementation Model for Dialog Design, Proceedings of Interact'87, North Holland, Stuttgart, (Septembre 1987), 431-436.

[COUTAZ 94]
J. Coutaz, L. Nigay, Les propriétés “CARE” dans les interfaces multimodales, Actes de IHM'94, Lille, (Décembre 1994).

[MARTIN 94]
J.-C Martin, Cadre d’étude de la multimodalité fondé sur les types et buts de coopération entre modalités, Actes de Montpellier'94,, (Fév. 1994), 97-106.

[NIGAY 93]
L. Nigay, J. Coutaz, A design space for multimodal interfaces: concurrent processing and data fusion, Proceedings of InterCHI'93, Amsterdam, (Avril 1993), 172-178.

[NIGAY 94]
L. Nigay, Conception et modélisation logicielles des systèmes interactifs, thèse de l’Université Joseph Fourier, Grenoble, (Janvier 1994), 350 pages.

[IHM 92]
Atelier IHM'92, ENST, Interfaces Multimodales et Architecture Logicielle, (30 Nov.-2 Déc. 1992).