

Le modèle de cohérence perceptive pour les applications multimédia interactives et distribuées

Nicolas Bouillot
CEDRIC-CNAM 292 rue St Martin
75141 Paris cedex 03 France
bouillot@cnam.fr

Résumé

Les Applications Multimédia Interactives et Distribuées (AMID) permettent la collaboration en temps réel entre plusieurs utilisateurs, à travers un système qui met en relation un groupe d'individus. La réalité mixte, la réalité virtuelle, les jeux vidéo, les performances musicales en réseau, le théâtre virtuel, les groupware interactifs sont des exemples d'AMID où les ambiguïtés temporelles liées à l'échange de messages provoquent des incohérences entre les états des répliques.

Pour de telles applications, nous proposons un modèle de cohérence orienté utilisateur, dont les critères sont basés sur la perception du temps physique, lorsque des personnes sont proches. Ensuite, grâce à l'utilisation de spécifications basées sur des horloges physiques locales, nous montrons que notre modèle est libre de conflits d'ordonnancement des opérations.

1 Introduction

De façon générale, la contrainte forte des AMID est de provoquer chez les utilisateurs un sentiment de co-présence [MRWJ03]. Cet objectif peut-être obtenu avec des astuces d'interface homme machine, par exemple en utilisant des interfaces à taille humaine, ou encore la spatialisation du son. En fait, nous ressentons ce sentiment de présence dans notre

vie de tous les jours d'une part grâce à l'interprétation des événements que nous percevons (odeurs, couleurs, dimensions...), mais aussi car la physique y est propice, en terme de perception du temps et de l'espace. En effet, pour des personnes proches, l'ordonnancement des événements et la progression dans le temps sont "identiques". C'est sous cet aspect que nous étudions la cohérence pour les AMID.

Comme dans les systèmes de Mémoires Réparties Partagées (MRP), les AMID sont composées de répliques distantes qui communiquent. Dans les deux types de systèmes, les critères de cohérence sont orientés par la sémantique du système simulé, i.e. l'ordre du programme pour les MRP et la perception physique des utilisateurs dans les AMID. C'est à dire que la conception d'une AMID doit tenir compte de la rapidité de retour à l'utilisateur d'une action qu'il a lui-même effectué, du rapport temporel qu'il entretient avec des événements dont il n'est pas à l'initiative, et enfin des délais de propagation de l'information. De plus, contrairement aux MRP, une AMID constitue virtuellement un environnement qui est naturellement parallèle.

Dans le paragraphe 2, nous allons proposer une adaptation du modèle d'historiques introduit par Raynal et Schiper dans [RS96], en vue d'établir un modèle pour les AMID. Dans le paragraphe 3, nous exposerons les contraintes temporelles que nous utilisons comme critères de cohérence, ainsi

que leurs répercussions sur l'utilisation des AMID. Nous présentons ensuite la cohérence perceptive et ses caractéristiques, avant de conclure au paragraphe 5.

2 Un modèle pour les AMID

Dans les MRP, la valeur d'une donnée n'a d'intérêt que lorsqu'un appel à la fonction *read* est effectué. Cela s'oppose aux AMID, où les utilisateurs partagent des données média, non seulement issus de la numérisation de leurs actions, mais aussi perçus continuellement. De ce point de vue, l'accès aux données n'est pas initié par une demande explicite (opération *read*), mais par une notification des mises à jours (opération *handle*). Plus précisément, dans les AMID, la communication est de type "publish/subscribe". C'est pour ces raisons que dans notre modèle, l'utilisation de l'opération *handle* est préférable à l'opération *read*.

Cependant, nous ne perdons pas en généralité, car l'opération *read* peut-être vue comme une simplification de l'opération *handle*. Par exemple, en procédant comme ci-après : localement à une réplique, une lecture sur un objet renvoie la valeur retournée par la dernière opération *handle* locale. Sachant que le mot *dernière* est non ambigu car il sous entend une vision locale, celle de l'ordre du processus.

Notation Un système d'AMID est composé d'un ensemble fini de processus séquentiels $P = (p_1, \dots, p_n)$ qui interagissent par l'intermédiaire d'un ensemble fini d'objets $X = (x_1, \dots, x_m)$. Chaque objet de X est une instance de média¹ qui est accédée par des opérations de type *write* et *handle*.

L'écriture d'une valeur v sur un objet x par le processus p_i est notée $w_i(x)v$, où chaque valeur écrite v est considérée comme unique. De la

¹Nous considérons qu'une *instance de média* correspond à l'utilisation spécifique d'un média. Par exemple le son est un média, tandis que des flux audio dans une application de téléphone sont des instances de média.

même façon, l'opération *handle* associée à cette écriture est notée $c_i(x)v$. Afin de simplifier la notation, *op* sera utilisé pour noter soit w , soit c .

Au niveau de chaque réplique, une horloge physique locale permet d'attribuer une date locale aux opérations. Nous savons, d'après les travaux de David Mills sur NTP qu'une synchronisation parfaite et absolue des horloges sur un réseau de type IP est impossible [IEE04]. En conséquence, il n'est pas possible, dans un système distribué, d'utiliser une horloge globale pour déduire des relations d'ordre entre les opérations, sans introduire une marge d'erreur.

C'est la raison pour laquelle nous utilisons des horloges physiques locales pour définir les contraintes temporelles. Plus, formellement, nous notons t_i l'horloge physique locale du processus p_i . En conséquence, $t_i(w_i(x)v)$ date localement et physiquement l'écriture de la valeur v sur l'objet x . Notons que $t_i(w_j(x)v)$ n'a pas de sens puisque t_i ne peut dater que des opérations locales à p_i . Nous utiliserons donc le plus possible la notation t à la place de t_i , car il n'y a pas d'ambiguïté sur l'horloge utilisée.

Précédemment introduit par [RS96], l'historique local \hat{h}_i de p_i est la séquence d'opérations exécutée par p_i . Si $op1$ et $op2$ sont exécutées par p_i et $op1$ est exécutée en première, alors $op1$ précède $op2$ dans l'ordre du processus p_i . Cela est noté $op1 \rightarrow_i op2$. Soit h_i l'ensemble des opérations exécutées par p_i ; l'historique local \hat{h}_i est l'ordre total (h_i, \rightarrow_i) .

Un historique \hat{H} d'une AMID est l'ordre partiel $\hat{H} = (H, \rightarrow_H)$ tel que :

- $H = \bigcup_i h_i$
- $op1 \rightarrow_H op2$ si :
 - i) $\exists p_i : op1 \rightarrow_i op2$ (dans ce cas, la relation \rightarrow_H est appelée *process-order*)
 - ou ii) $op1 = w_i(x)v \wedge op2 = c_j(x)v$ (*handled-from*)
 - ou iii) $\exists op3 : op1 \rightarrow_H op3$ et $op3 \rightarrow_H op2$

L'ordonnement des opérations Le formalisme d'historique permet de modéliser de façon efficace les systèmes de répliques tels que

les MRP et les AMID. Cependant, les propriétés d'ordre attendues sur ces historiques diffèrent selon le type d'applications. En effet, les MRP sont destinés à l'exécution de processus programmés, où le programmeur suppose que l'état du système est mis à jour après chaque instruction. Dans ce cas, l'exécution répartie doit tenir compte de cette dépendance, qui est formalisée par la légalité causale [Lam79, RS96].

Les AMID constituent virtuellement un environnement naturellement parallèle où les données sont modifiées dynamiquement selon le bon vouloir des utilisateurs. En d'autres termes, les actions des utilisateurs ne sont pas fonction de la toute dernière opération effectuée sur l'instance de média, mais par l'ensemble des actions effectuées "récemment". Nous pouvons alors dire que la causalité, telle qu'elle est décrite dans [Lam79, RS96], est un critère abusif pour les AMID.

Cependant, les AMID ne sont pas libres de contraintes liées à l'ordre, car il est nécessaire d'éviter les conflits causés par des ordonnancements locaux différents, parmi les répliques. Nous définissons que deux opérations $op1$ et $op2$ sont conflictuelles si $\exists i, j : c_i(x)v \rightarrow_i c_i(x)u$ et $c_j(x)u \rightarrow_j c_j(x)v$. Le fait d'éviter ce type de conflits permet aux répliques de prendre des décisions immédiatement et consensuellement sur l'effet qu'une action va avoir sur l'environnement, comme une décision de succès dans un jeu, ou encore la couleur d'un cercle dans une application de tableau blanc.

3 Les contraintes temporelles

Dans notre vie de tout les jours, lorsque nous interagissons avec d'autre personnes, nous possédons chacun notre propre perception de l'écoulement du temps. Dans ce référentiel, nous y percevons nos propre actions, comme un retour de nous même, mais aussi les actions venant d'éléments extérieurs. Ce sont ces caractéristiques que nous essayons de capturer avec les notions d'instantanéité, de Δ légalité et de simultanéité.

L'instantanéité Les utilisateurs d'applications multimédia sont habitués à un *effet immédiat* d'une action sur l'application. L'application doit donc avoir un temps de réponse faible. Cet effet caractérise la notion d'instantanéité. Formellement, une opération $handle\ c_i(x)v$ est instantanée si :

$$t(c_i(x)v) - t(w_i(x)v) \leq \tau_{resp}(x)$$

Où τ_{resp} est le délai toléré par un utilisateur entre une action et sa perception. Par exemple, dans les applications audio, en fonction de l'utilisateur, un délai de 5ms à 40ms peut être toléré. Ainsi, un historique \hat{H} est *instantané* si toutes les opérations *handle* sont instantanées.

Cependant, nous pouvons noter que selon les applications, certaines astuces permettent de relâcher la contrainte d'instantanéité. Par exemple, dans le cas de la musique, où le plus important est d'avoir un tempo commun, nous avons utilisé [Bou03] un délai local exagérément grand, mais conservant le tempo des différents flux. Pour cela, les musiciens configurent l'application en donnant un délai qu'ils expriment en nombre de noires dans un tempo spécifié en Battement Par Minute (BPM). Le retard est alors supportable, car "en rythme".

La Δ légalité Dans certaines AMID, les délais entre deux écritures locales doivent être répercutés au niveau des opérations *handle* associées. Par exemple pour des flux de type audio/vidéo, ou encore des positions dans un jeu de voiture, où la vitesse doit être conservée malgré l'échange de positions absolues entre répliques. Plus formellement, un historique \hat{H} est Δ légal si :

$$\forall x \forall u, v \forall i, j : t_i(w_i(x)v) - t_i(w_i(x)u) = t_j(c_j(x)v) - t_j(c_j(x)u)$$

Nous pouvons remarquer que cela définit une latence constante entre les différents processus. En effet si nous supposons que les horloges t_i et t_j sont parfaitement synchronisées, alors elles renvoient une date universelle qui permet de comparer leurs dates respectives. Nous avons alors : $\forall u, v : t_j(c_j(x)u) - t_i(w_i(x)u) = t_j(c_j(x)v) -$

$t_i(w_i(x)v)$

En d'autre terme, la latence entre p_i et p_j doit rester constante au cours du temps (la gigue doit être rendue nulle). Nous notons $\delta_{x,i,j}$ une telle constante, qui correspond à une contrainte sur le délai réseau entre p_i et p_j .

C'est principalement cette propriété qui va poser problème aux protocoles de cohérence perceptive, car la latence est variable sur les réseaux IP. Les protocoles de cohérence perceptive seront donc optimistes car ils supposeront que les messages vont arriver avant un certain délai (qui devra être calibré en fonction de l'état du réseau, cf paragraphe 4).

La simultanéité A notre connaissance, les premiers travaux sur l'étude de la perception du temps par un être humain sont ceux de A. Einstein dans [Ein05]. Dans cet article, il propose une étude fine de la notion de simultanéité. Plus particulièrement, dans la "partie cinématique", il propose la définition suivante :

Si [...] je dis que "le train arrive à 7 heure", cela veut dire : "Le fait que la petite aiguille de ma montre pointe sur 7 heure et l'arrivée du train sont deux événements simultanés"

Dans le cas des AMID, la simultanéité peut être de la première importance. Par exemple dans un jeu de course automobile, il est important que plusieurs joueurs perçoivent certains événements simultanément, notamment lors du franchissement de la ligne d'arrivée. En effet, l'expression "lorsque la voiture A traverse la ligne d'arrivée, la voiture B est 2 mètres derrière" est équivalente à l'expression "la voiture A gagne la course et la voiture B est à 2 mètres de l'arrivée sont deux événements simultanés". De plus cela est vrai du point de vue de B, mais aussi et de celui de A.

Dans la partie suivante, "De la relativité du temps et de l'espace", il est montré que la simultanéité est propre à chaque référentiel temporel, et non à une définition globale du temps. Cela montre qu'un être humain perçoit deux événements simultanés uniquement dans son propre

système temporel, et que la perception simultanée et partagée par plusieurs humains n'est possible que pour des personnes proches. A partir de ce constat, nous pouvons définir formellement le concept de simultanéité, pour des personnes proches :

Dans un historique \hat{H} , les événements $c_j(x_k)v$ et $c_j(x_l)u$ (n'ayant pas forcément le même écrivain) sont simultanés si :

$$\forall p : t_j(c_j(x_k)v) - t_j(c_j(x_l)u) = t_p(c_p(x_k)v) - t_p(c_p(x_l)u)$$

En fait, cela définit la perception de la simultanéité au cours du temps, et en des lieux différents. En effet la perception de deux événements simultanés $c(x_k)v$ et $c(x_l)u$ par deux personnes proches (p_j et p_p) est un cas spécial de la définition précédente, et est défini comme suit :

$$t_j(c_j(x_k)v) = t_j(c_j(x_l)u) \text{ et } t_p(c_p(x_k)v) = t_p(c_p(x_l)u)$$

Nous pouvons maintenant définir le concept d'historique simultané. \hat{H} est *simultané* si :

$$\forall p \forall c_{p1}, c_{p2} \in H : c_{p1} \text{ et } c_{p2} \text{ sont des événements simultanés.}$$

4 La cohérence perceptive (PC)

Dans sa première définition formelle [Qin02], PC fut appelée "cohérence absolue" et spécifiée avec une horloge universelle. Cependant, de par notre définition de la simultanéité, nous déduisons que cette appellation ne convient pas.

Définition *Un historique \hat{H} est perceptif cohérent si il est simultané et Δ légal.*

PC et les conflits Si un historique \hat{H} est perceptif cohérent, alors il n'admet pas de conflit. En effet, comme nous utilisons des horloges physiques locales, nous pouvons transformer la relation de précédence temporelle en relation d'ordre, i.e.

$$t_j(c_j(x)v) < t_j(c_j(x)u) \Leftrightarrow c_j(x)v \rightarrow_j c_j(x)u$$

Donc, si un historique \hat{H} est simultané :

$$\forall c_j(x)v, c_j(x)u \in H \forall k : (c_j(x)v \rightarrow_j c_j(x)u) \Leftrightarrow$$

$$(c_k(x)v \rightarrow_k c_k(x)u)$$

Par définition, les conflits sont évités. Cependant, les spécifications temporelles de la Δ légalité permettent l'exécution d'opérations *handle* à la même date locale, comme le montre le cas suivants, où $\delta_{x,i,j}$ et $\delta_{x,k,j}$ sont les constantes exprimant les latences réseau et où T est l'horloge universelle (voir la définition de la Δ légalité paragraphe 3) :

$$\begin{aligned} \delta_{x,i,j} + T(w_i(x)v) &= \delta_{x,k,j} + T(w_k(y)u) \\ \Rightarrow T(c_j(x)v) &= T(c_j(y)u) \\ \Rightarrow t_j(c_j(x)v) &= t_j(c_j(y)u) \end{aligned}$$

Dans ce cas, les conflits peuvent être évités en donnant la priorité d'exécution à l'opération *handle* dont l'écrivain à l'identifiant le plus faible [VM01]. Cela permet d'éviter les conflits sans échange de messages.

PC et l'instantanéité Nous allons maintenant analyser l'effet de la cohérence perceptive sur la notion d'instantanéité, et montrer que dans un historique conforme à PC, l'instantanéité est fonction des latences réseau entre les répliques.

Supposons deux écrivains p_i et p_j sur la même instance de média x , alors de la Δ légalité nous avons (T est l'horloge universelle) :

$$\begin{aligned} \text{(i)} \quad T(c_j(x)v) &= T(w_i(x)v) + \delta_{x,i,j} \text{ et } T(c_i(x)v) = \\ &T(w_i(x)v) + \delta_{x,i,i} \\ \text{(ii)} \quad T(c_i(x)u) &= T(w_j(x)u) + \delta_{x,j,i} \text{ et } T(c_j(x)u) = \\ &T(w_j(x)u) + \delta_{x,j,j} \end{aligned}$$

De la simultanéité, nous avons :

$$\text{(iii)} \quad T(c_j(x)v) - T(c_j(x)u) = T(c_i(x)v) - T(c_i(x)u)$$

De (i) et (ii) dans (iii), nous avons :

$$\begin{aligned} T(w_i(x)v) + \delta_{x,i,j} - T(w_j(x)u) - \delta_{x,j,i} &= \\ T(w_i(x)v) + \delta_{x,i,i} - T(w_j(x)u) - \delta_{x,j,i} & \end{aligned}$$

$$\text{alors } \delta_{x,i,i} + \delta_{x,j,j} = \delta_{x,i,j} + \delta_{x,j,i}.$$

Il est donc préférable que les latences locales $\delta_{x,i,i}$ et $\delta_{x,j,j}$ soient choisies un peu plus grande que les latences réseau entre p_i et p_j . Sinon, la plupart des opérations *handle* risquent d'arriver tardivement, et donc d'être illégales. En d'autres termes, le délai $\delta_{x,i,i} + \delta_{x,j,j}$ doit être plus grand que le délai aller-retour entre p_i et p_j . Cela rend l'ins-

tantanéité avec la simultanéité possible uniquement lorsque les latences sont telles que :

$$\delta_{x,i,j} \leq \tau_{resp}(x) \text{ et } \delta_{x,j,i} \leq \tau_{resp}(x)$$

Le dimensionnement des AMID ne pourra donc se faire qu'avec une connaissance précise des seuils de réponse des différents types de média. Dans la littérature, nous pouvons trouver un nombre important d'études sur la perception des délais par des utilisateurs. Citons notamment [PW02, PK99].

Les protocoles Nous pouvons trouver des protocoles proches de la cohérence perceptive dans [VM01, GD98, AY96]. Dans [VM01, GD98], les protocoles utilisent une horloge universelle afin de déterminer une latence fixe et identique entre tous les participants, et pour toutes les instances de média. Dans [AY96], c'est grâce à une estimation du délai aller retour que le système décide de la date à laquelle une valeur doit prendre effet.

Le principal défaut de ces protocoles est l'utilisation d'une horloge universelle pour obtenir la propriété de simultanéité. L'erreur de synchronisation des horloges se répercute directement dans le protocole : soit par des erreurs temporelles, soit par des conflits.

Finalement, [Bou03] propose le protocole le plus fidèle à la cohérence perceptive car il utilise une synchronisation de flux isochrones basée sur les horloges locales des processus. Il s'agit d'effectuer un consensus sur un vecteur de dates locales d'opérations *handle* qui doivent être jouées simultanément.

5 Conclusion

Les Applications Multimédia Interactives et Distribuées (AMID) sont naturellement soumises à des contraintes temporelles fortes. Dans cette article, nous avons proposé, pour ce type d'applications, un modèle de cohérence basé sur la perception des événements par les utilisateurs, et au cours du temps physique. En réutilisant

le modèle des historiques de [RS96] et en y modifiant la sémantique de l'opération de lecture, nous avons montré que la causalité est un critère abusif pour les AMID, et que l'utilisation de spécifications basées sur des horloges locales permet à notre modèle de cohérence d'être libre de conflits d'ordonnancements d'opérations.

Nous espérons pouvoir bientôt tester des protocoles de cohérence perceptive, et les confronter à des utilisateurs afin de valider de façon "pratique" ce modèle.

Références

- [AY96] I. Akyildiz and W. Yen. Multimedia group synchronization protocols for integrated services networks. *IEEE Journal on selected area in communications*, Vol. 14(No. 1) :p. 162, 1996.
- [Bou03] N. Bouillot. Un algorithme d'auto synchronisation distribuee de flux audio dans le concert virtuel reparti. In *Proc. of The Conference Francaise sur les Systemes d'Exploitation (CFSE'3)*, La Colle sur Loup, France, October 2003.
- [Ein05] A. Einstein. On the electrodynamics of moving bodies. *Annalen der Physik*, 17, June 1905. translated from german.
- [GD98] L. Gautier and C. Diot. Design and evaluation of MiMaze, a multi-player game on the internet. In *International Conference on Multimedia Computing and Systems*, pages 233–236, 1998.
- [IEE04] Precision clock synchronization protocol for networked measurement and control systems. IEEE standard, 2004. IEC 61588 First edition 2004-09 ; IEEE 1588.
- [Lam79] L. Lamport. How to make a multiprocessor computer that correctly executes multiprocess programs. *IEEE Transaction on Computers*, C28(9) :690–700, 1979.
- [MRWJ03] M. Meehan, S. Razzaque, M. C. Whitton, and F. P. Brooks Jr. Effect of latency on presence in stressful virtual environments. In *IEEE Virtual Reality*, page p. 141, Los Angeles, CA, March 22 - 26 2003.
- [PK99] K. Shin Park and R. V. Kenyon. Effects of network characteristics on human performance in a collaborative virtual environment. In *Proceedings of the IEEE Virtual Reality*, page 104. IEEE Computer Society, 1999.
- [PW02] L. Pantel and L. C. Wolf. On the impact of delay on real-time multiplayer games. In *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, pages 23–29. ACM Press, 2002.
- [Qin02] X. Qin. Delayed consistency model for distributed interactive systems with real-time continuous media. *Journal of Software*, Vol.13(No.6) :1029–1039, June, 2002.
- [RS96] M. Raynal and A. Schiper. A suite of formal definitions for consistency criteria in distributed shared memories. In *Proceedings Int Conf on Parallel and Distributed Computing (PDCS'96)*, pages 125–130, Dijon, France, September 1996.
- [VM01] J. Vogel and M. Mauve. Consistency control for distributed interactive media. In *ACM Multimedia*, pages 221–230, 2001.