L'usage du son dans les systèmes interactifs

Michel Beaudouin-Lafon
Laboratoire de Recherche en Informatique
Bâtiment 490
Université de Paris-Sud, U.R.A. 410 du CNRS
91 405 ORSAY Cedex
e-mail : mbl@lri.fr

Résumé : *Cet article a pour objet de montrer les usages possibles du son dans les systèmes interactifs et les moyens de sa mise en œuvre. La première partie est une analyse de l'existant. La seconde partie propose un modèle de son structuré et son implémentation dans un serveur audio, permettant d'offrir aux développeurs d'interfaces des services comparables à ceux offerts par les boîtes à outils graphiques aujourd'hui largement utilisées pour la construction d'interfaces graphiques.*

Mots-clés : interfaces homme-machine, son structuré, synthèse sonore, serveur audio.

Introduction

La majorité des systèmes interactifs en usage aujourd'hui privilégient le graphique. En quelques années, les terminaux alphanumériques ont été remplacés par des stations de travail à écran graphique couleur, et de plus en plus souvent des processeurs graphiques spécialisés sont dédiés à la gestion de l'affichage, déchargeant le processeur central et autorisant des performances graphiques toujours plus évoluées. En ce qui concerne l'audio, on perçoit les premiers signes d'une évolution similaire. De plus en plus, les stations de travail sont munies de périphériques audio qui permettent aux applications d'émettre autre chose que des "bips" agaçants.

Cependant, les usages du son restent limités aux applications musicales et au multimédia. En dehors de ces domaines spécialisés, le canal audio est pratiquement ignoré, alors que dans la vie courante il nous est si précieux. L'objet de cet article est de montrer des usages possibles du son dans les systèmes interactifs, et de proposer la notion de son structuré pour sa mise en œuvre pratique.

L'usage du son dans les systèmes interactifs

Si le domaine de l'informatique musicale est ancien, l'usage du son dans les applications non musicales a été particulièrement négligé. Quelques applications utilisent des sons échantillonnés. L'un des meilleurs exemples est le SonicFinder [Gaver89] : dans cette version sonorisée du Finder du Macintosh, les actions de l'utilisateur, comme par exemple la sélection d'un icone, sont accompagnées d'un feed-back sonore qui donne des informations non visibles graphiquement. Ainsi, chaque catégorie d'icones (dossiers, documents, applications) produit un son différent, dont la hauteur dépend de la taille du fichier représenté par l'icone. L'intérêt du SonicFinder est d'utiliser des paramètres du son pour véhiculer des informations sémantiques (type et taille des objets dans cet exemple). Lorsque l'on compare le SonicFinder à d'autres implémentations qui ont ajouté du son au Finder de façon naïve (par exemple toute sélection d'icone produit le même son), on réalise qu'un son qui ne véhicule pas d'information sémantique est un bruit, inutile et dérangeant.

Avec les « earcons » [Blattner89], les sons utilisés sont des séquences de notes dont l'application peut contrôler les paramètres : mélodie, timbre, rythme. Comme avec le

SonicFinder, l'objectif est de permettre à l'application de véhiculer une information sémantique. Si les deux approches procèdent de principes très différents (approche musicale chez Blattner, approche écologique fondée sur l'usage de sons du monde réel chez Gaver), les buts poursuivis sont identiques et les résultats apparemment comparables.

Le son a également été utilisé pour rendre des applications accessibles à des aveugles ou mal-voyants, comme par exemple dans le projet Audicones [Martial93]. Il a aussi été utilisé dans des applications de travail coopératif comme Arkola [Gaver91]. Enfin il est utilisé dans des applications scientifiques. Il a alors pour objet de faciliter l'interprétation de grandes quantités de données, de la même façon que la visualisation dans le domaine graphique. Aussi parle-t-on dans ce cas d'"auralisation" [Kramer92].

D'autres applications utilisent le son de manière « passive ». Par exemple Microsoft Word permet d'enregistrer des annotations vocales associées à une position dans le document. Un icone apparaît dans le document, permettant d'écouter l'annotation lors de la lecture du document. On peut classer dans la même catégorie les messageries électroniques qui permettent d'échanger des messages vocaux et les systèmes de téléconférence utilisant des réseaux informatiques [Vin91]. Il faut également mentionner les applications en synthèse et en reconnaissance de la parole, que l'on a tendance à oublier lorsque l'on parle d'audio. Le point caractéristique de ces applications est de traiter la voix, soit sous forme brute (annotations, messageries, téléconférence), soit sous forme interprétée (synthèse et reconnaissance de la parole).

Sur le plan des outils de mise en œuvre, on a vu apparaître récemment des serveurs audio [Neville-Neil93], plus ou moins calqués sur les serveurs de fenêtres très répandus dans le domaine du graphique, destinés à faciliter l'usage du son dans les applications. L'intérêt de ces serveurs est de rendre les applications indépendantes des périphériques effectivement disponibles, de permettre le partage des périphériques audio entre plusieurs clients, et enfin d'accèder aux ressources à travers le réseau. Cependant, ces serveurs, au contraire de leurs aînés gérant les ressources graphiques, fonctionnent à un faible niveau d'abstraction. Les informations échangées entre clients et serveur sont des flots d'échantillons qui trahissent d'une part le fait que ces serveurs ont été conçus pour des applications de type téléconférence et multimédia, d'autre part le fait que l'on ne dispose pas de modèle de son structuré comme il en existe pour le graphique.

Taxonomie des usages du son dans les systèmes interactifs

On peut classer les usages du son dans les systèmes interactifs en quatre catégories :

- *usages musicaux* : l'objet de l'application est de traiter des informations de nature musicale ou à usage musical (aide à la composition, synthèse et traitement du son, etc.) ;
- *parole et voix* : l'application peut stocker, traiter, restituer la voix et/ou reconnaître et synthétiser la parole ;
- *feed-back sonore* : l'application utilise le son comme moyen de rétroaction afin de rendre l'interaction plus "performante" ;
- *notification* : le son permet de rendre compte d'évènements se produisant de manière asynchrone (arrivée de message électronique par exemple).

De ces quatre catégories, se sont les deux dernières qui nous intéressent tout particulièrement. Les domaines de la musique, de la parole et de la voix sont largement étudiés par ailleurs. Il est clair cependant que ces utilisations peuvent, dans une application donnée, se combiner. Ainsi une application musicale peut utiliser la reconnaissance de la parole comme moyen de contrôle et la synthèse de son pour le feed - back des actions de l'utilisateur. L'analyse ci-dessous se limite aux deux dernières catégories : feed-back et notification.

Dans les interfaces homme-machine, l'objet du feed-back est double. D'une part il fournit à l'utilisateur une information lui indiquant comment l'application est en train d'interpréter l'action en cours. L'utilisateur peut ainsi savoir s'il fait bien ce qu'il croit faire. Un exemple de ce type de feed-back est le changement de la forme du curseur en fonction du mode actif. D'autre part, le feed-back a pour rôle d'assister l'utilisateur dans la spécification de l'action qu'il est en train d'exécuter. Ainsi, lorsque l'utilisateur déplace un objet à la souris, le feed - back consiste généralement en une "ombre" de l'objet, qui suit la souris et permet ainsi de poser précisément l'objet à l'endroit souhaité.

La plupart du temps, le feed-back est obtenu de manière graphique. Cela présente cependant des inconvénients, notamment la surcharge de l'écran et la limitation de l'information retournée. Ce dernier cas s'observe par exemple dans l'interface du Finder du Macintosh : lorsque l'on déplace un icone vers un autre icone, celle-ci s'"allume", indiquant qu'il est prêt à recevoir l'icone déplacée. Cependant, selon les cas, l'action résultante peut être le déplacement, la copie ou la destruction de l'icone déplacée, ou même le lancement d'une application. Il y a donc ambiguïté potentielle pour l'utilisateur.

L'usage du son permet de résoudre ces problèmes. Dans le dernier exemple, un son qui accompagne le passage de la souris sur les icones peut indiquer à l'utilisateur l'action qu'il est sur le point d'exécuter. De manière générale, le son peut contribuer efficacement à renforcer la métaphore du monde réel que la plupart des applications graphiques tentent de mettre en œuvre, comme la métaphore du bureau dans le cas du Finder : dans le monde réel les objets produisent des sons, qui participent de manière fondamentale à notre perception du monde. Le SonicFinder illustre parfaitement l'extension de cette métaphore par l'usage du son pour le feed-back. Il ressort cependant de nos expérimentations que la mise en œuvre du son pour le feed-back n'est pas simple à cause de la propension de l'ouïe à se lasser de sons répétitifs. C'est pourquoi il est indispensable de véhiculer le maximum d'information sémantique dans les sons produits, ce qui de toute évidence ne peut être obtenu par l'usage de simples sons enregistrés comme c'est le cas dans presque tous les systèmes utilisant le son dans l'interaction.

En ce qui concerne la notification, son importance devient croissante dans les systèmes interactifs modernes. L'utilisateur a en effet la possibilité de mener de front plusieurs tâches, et le changement de tâche active est souvent dirigé par les notifications qu'il reçoit. Ainsi, lorsque l'utilisateur est notifié de l'arrivée d'un message, il aura tendance à interrompre son activité (immédiatement ou lors du prochain point de rupture de son activité courante) pour aller lire le message reçu. Dans les applications de contrôle de processus (par exemple la salle de contrôle d'une centrale nucléaire), le rôle des notifications et des alarmes est primordial.

La capacité de l'oreille humaine à percevoir des sons de manière permanente fait du son un candidat parfait pour la notification. De fait, la plupart des alarmes sont sonores. Mais le son peut s'avérer utile pour des circonstances moins exceptionnelles. En particulier, le son peut être utilisé pour la surveillance (« monitoring ») d'activités parallèles. Ainsi, lorsque l'utilisateur est engagé dans une tâche qui réclame un long traitement par la machine, ce traitement peut produire un son lui permettant de surveiller l'avancement du calcul pendant qu'il est engagé dans une autre activité. Là encore, il est important que le son véhicule suffisamment d'information sur le type et l'état d'avancement de la (des) tâche(s) d'arrière-plan.

Un cas particulier d'usage de la notification est celui des applications collecticielles. Par exemple, dans un éditeur partagé qui permet à plusieurs utilisateurs de modifier, en temps réel, un document depuis leurs postes de travail respectifs, le son permet de fournir à chaque utilisateur des informations concernant l'activité des autres utilisateurs, de telle sorte que chaque utilisateur est conscient de l'activité du groupe. Là encore le son est plus adapté que le graphique, comme le montre Arkola [Gaver91], et nos propres travaux sur GroupDesign [Beaudouin-Lafon92].

Vers un modèle de son structuré

L'analyse de la section précédente montre d'une part que la plupart des applications qui utilisent le son ne s'intéressent pas à sa structure, d'autre part que l'usage du son présente un intérêt s'il véhicule une information sémantique riche. Il y a là une incohérence qu'il parait important de résoudre. Transmettre une information sémantique riche implique que l'application ait un contrôle important et de haut niveau sur les sons produits. L'objet du modèle de son structuré présenté ci-dessous est de permettre un tel contrôle.

Le modèle est fondé sur la notion de *source sonore* . Une source sonore est caractérisée par un type et un ensemble d'attributs. Les sources sonores sont organisées en un arbre, que l'on pourrait comparer à l'arbre des fenêtres d'un système de fenêtrage comme X-Windows [Scheifler91]. Comme dans les systèmes de fenêtrage, certains attributs des sources sont relatifs : leur interprétation dépend de la valeur du même attribut pour le parent de la source considérée. Ainsi, la position de la source sonore dans l'espace et son gain sont des attributs relatifs, au même titre que la position d'une fenêtre est relative à sa fenêtre parente. Ceci permet de déplacer un ensemble de sources ou de changer leur gain en agissant simplement sur l'attribut de leur ancêtre commun.

La production de sons est obtenue en créant des *interactions* avec les sources. Par exemple, l'interaction avec une source représentant un instrument de musique contient les notes à jouer, tandis que l'interaction avec un synthétiseur de parole spécifie la phrase à prononcer. Selon le type d'interaction, une même source peut produire différents types de sons. Ainsi une source qui représente un objet sonore peut subir des interactions telles que l'impact, le frottement, le bris. Enfin, une source peut produire plusieurs sons qui se recouvrent temporellement. En particulier, les nœuds de l'arbre ne produisent en général pas de son directement, mais font produire des sons par leurs descendantes en contrôlant leur synchronisation temporelle. Par exemple, une source "chef d'orchestre" détenant une partition contrôle des sources "instruments". Par le même mécanisme, une source peut encapsuler une interaction complexe comme le rebondissement d'un objet, qui consiste à faire produire par une sous-source un ensemble de sons d'impact avec des intervalles de temps et des gains décroissants.

On peut à nouveau comparer cette approche à celles des systèmes graphiques. Les sons produits par les sources correspondent aux objets graphiques (cercles, polygones, etc.) que l'on peut afficher dans des fenêtres. Le contrôle de l'organisation temporelle des sons par les sources correspond dans les systèmes graphiques au contrôle de l'organisation spatiale des objets (gestion de la géométrie). Cependant, il ne faut pas chercher le parallèle avec le graphique à tout prix. Il est mis en évidence ici simplement pour montrer que la démarche est similaire. Par ailleurs, d'autres travaux, comme par exemple le système Formes [Rodet84] ont déjà fait usage de modèles structurés pour le son. L'originalité ici réside dans l'arbre des sources et dans le fait que le modèle est destiné à des applications non musicales.

Mise en œuvre expérimentale : ENO

Afin de valider le modèle esquissé ci-dessus, j'ai développé un prototype de serveur sur station de travail ainsi qu'un ensemble d'applications de démonstration. Le serveur est une version modifiée du système AudioFile [Levergood93]. Il permet à plusieurs clients, à travers le réseau, de manipuler l'arbre des sources et de produire des sons. Quatre types de sources sont implémentées :

- Les *échantillons* sont des sources qui produisent un son stocké sous forme d'échantillons dans un fichier.
- Les *machines* sont des sources qui produisent des sons par synthèse FM, selon l'algorithme décrit par Bill Gaver [Gaver93]. On peut contrôler la taille de la

machine (fréquence de la porteuse), sa vitesse (fréquence de la modulation), et l'intensité du travail fourni (profondeur de la modulation).

- Les *objets sonores* sont des sources qui représentent des objets définis par une forme, une taille et un matériau (caractérisé par sa résonnance : mat, comme du bois, jusqu'à brillant, comme du verre ou du métal). Les sons produits peuvent être des sons d'impact et des sons de frottement. Les algorithmes de synthèse sont aussi ceux inventés par Bill Gaver [Gaver93].
- Les *groupes* sont des sources qui ne produisent pas de son mais servent de nœuds dans l'arbre des sources. Nous n'avons pas encore implémenté de source qui contrôle les sons produits par les sous-sources.

Pour toutes ces sources, les valeurs des attributs sont contrôlables en temps réel, avec effet immédiat sur les sons produits. Trois applications de démonstration utilisent le serveur audio (baptisé ENO). Ces applications peuvent fonctionner simultanément.

Adraw est une mini-application de dessin qui utilise le son pour le feed-back. À chaque objet graphique est associé un objet sonore dont la taille et le matériau dépendent respectivement de la taille de l'objet graphique et de sa couleur. Les actions comme la sélection, le déplacement, le changement de taille utilisent des interactions d'impact et de frottement avec contrôle en temps réel des paramètres. Ainsi, la vitesse du frottement est coordonnée à la vitesse de déplacement de la souris.

Amake est une version de *make* (utilitaire Unix de gestion de configuration) qui utilise des sons de machine pour informer l'utilisateur de la progression des opérations. Chaque commande lancée par *amake* produit un son de machine dont les caractéristiques dépendent de la commande et évoluent au cours du temps. L'expérience montre que, très rapidement, l'oreille "apprend" le son d'une session *amake*, ce qui permet à l'utilisateur de mieux paralléliser ses tâches.

Atwm est une version sonorisée du gestionnaire de fenêtres *twm* . Le principe est similaire à celui de *adraw* , et l'on retrouve quelques principes inspirés du Sonic Finder [Gaver89]. Comme il s'agit d'une application "réelle" (au contraire de *adraw* qui est une application de démonstration), on peut étudier la pertinence de l'usage du son sur de longues sessions ainsi que les problèmes liés à l'usage simultané du son dans plusieurs applications. À cet effet, *atwm* dispose de mécanismes permettant de spatialiser les sons des applications en fonction de la position des fenêtres à l'écran et de leur ordre de superposition.

Conclusion et Perspectives

Le son mérite mieux que ce que lui offre le multimédia. Le modèle de son structuré présenté dans cet article et sa mise en œuvre dans un serveur sont une première tentative pour faciliter le travail des développeurs d'interfaces en leur fournissant des outils appropriés. L'implémentation d'ENO est encore loin d'être parfaite. Cependant, les fonctionnalités présentes et la facilité de développement d'applications permettent déjà de tirer de valider l'approche choisie. La combinaison du modèle de son structuré avec les algorithmes de synthèse de Bill Gaver permet de contrôler les sons par des grandeurs physiques (taille, matériau, vitesse, etc.) et non par des paramètres de synthèse (fréquence, largeur de bande, etc.).

Cette approche apparaît très prometteuse, et appelle d'ores et déjà d'autres développements. En premier lieu, d'autres algorithmes de synthèses sont nécessaires. Nous avons commencé à explorer des approches de synthèse par modèles physiques [Roads93]. Ces algorithmes doivent être efficaces pour permettre une synthèse et un contrôle des paramètres en temps réel sur une station qui doit aussi exécuter des applications. Ensuite, il faut compléter la synthèse par des traitements permettant la spatialisation du son. Enfin, il faut continuer les expérimentations visant à identifier les meilleurs moyen d'utiliser le son dans les interfaces.

Remerciements

Ces travaux ont débuté lors de mon passage au laboratoire Rank Xerox EuroPARC à Cambridge (Angleterre). Je remercie Bill Gaver pour son accueil et son aide précieuse.

Références

[Beaudouin-Lafon92] Beaudouin-Lafon, M. and Karsenty, A., *Transparency and Awareness in a Real-Time Groupware System*, in Proc. ACM Symposium on User Interface Software and Technology (UIST'92, Monterey, November 1992), ACM Press, New York.

[Blattner89] Blattner, M., Sumikawa, D., and Greenberg, R., *Earcons and Icons: Their Structure and Common Design Principles*, Human-Computer Interaction, volume 4, pp 11 - 44, Lawrence Erlbaum Associates, 1989.

[Gaver89] Gaver, W., *The Sonic Finder: An Interface That Uses Auditory Icons*, Human-Computer Interaction, volume 4, pp 67-94, Lawrence Erlbaum Associates, 1989.

[Gaver91] Gaver, W., Smith, R.B., and O'Shea, T., *Effective Sounds in Complex Systems: The ARKola Simulation*, in Proc. ACM Conference on Human Factors in Computing Systems (CHI'91, New-Orleans, April 1991), ACM, New York, 1991.

[Gaver93] Gaver, W., *Synthesizing Auditory Icons*, in Proc. ACM Conference on Human Factors in Computing Systems (INTERCHI'93, Amsterdam, April 1993), pp 228-235, ACM, New York, 1993.

[Kramer92] Kramer, G. and Lane, N., *Some Organizing Principles for Representing Data with Sound*, in Proc. First International Conference on Auditory Display (ICAD'92), SFI Studies in the Sciences of Complexity, volume 18, Addison-Wesley, 1993.

[Levergood93] Levergood, T., Payne, A., Gettys, J., Treese, W., and Stewart, L., *AudioFile: A Network-Transparent Audio Server for Distributed Audio Applications*, in Proc. Summer Usenix Conference, Usenix Association, 1993.

[Martial93] Martial, O. and Dufresne, A., *Audicon: Easy Access to Graphical User Interfaces for Blind Persons - Designing for and with People*, in Proc. HCI'93 (Orlando), volume 19B, pp 808-813.

[Neville-Neil93] Neville-Neil, G., *Current Efforts in Client/Server Audio*, The X Resource, Issue 8, 1993.

[Roads93] Roads, C., *Initiation à la Synthèse par Modèles Physiques*, Les Cahiers de l'IRCAM, Numéro 2, La synthèse sonore, 1993.

[Rodet84] Rodet, X. and Cointe, P., *Formes : Composition and Scheduling of Processes*, Computer Music Journal, Vol. 8, Numéro 3, 1984, pp.32-50.

[Scheifler91] Scheifler, R. and Gettys, J., *XWindow System*, Digital Press, Bedford, 1991.

[Vin91] Vin, H., Zellweger, P., Swinehart, D., and Rangan, V., *Multimedia Conferencing in the Etherphone Environment*, IEEE Computer, october 1991, pp 69-79.