L'EDITEUR DE SIGNAL"A DISTANCE" MAPSIGNAL

Alain de Cheveigné

Laboratoire de Linguistique Formelle, CNRS - Université Paris 7, France.

ABSTRACT

MapSignal is a program that allows the display and manipulation of numerical data over a serial line or LAN. The data to be manipulated resides on a host computer (mainframe, mini, etc.), the user interface is on a microcomputer. Advantages are a) flexibility in the choice of processing hardware and software, and b) the high-quality user interface available on microcomputers. It is argued that the concept of *separate machines* for processing and user interface compares favorably with the more familiar concept of signal-processing *workstation* in which these elements are combined.

1. INTRODUCTION

De nouveaux éditeurs de signal sont proposés régulièrement. Ce foisonnement est le reflet des problèmes que posent la *portabilité* des outils de manipulation graphique du signal. Un éditeur de signal doit gérer à la fois l'*affichage graphique*, la *conversion numérique/analogique* et le *traitement du signal*. Ces trois éléments subissent une évolution technologique rapide que les logiciels ont du mal à intégrer. Chaque fois qu'apparaît un nouveau matériel, les contraintes de la compatibilité logicielle viennent compromettre ses avantages de coût et de performance.

Le présent article aborde ce problème sous un angle inhabituel, en faisant l'hypothèse qu'un des trois élements (l'interface graphique) se trouve sur une machine distincte. Il décrit une réalisation concrète fondée sur cette hypothèse: l'éditeur de signal MapSignal.

2. LA STATION DE TRAVAIL

2.1. Une solution idéale sur le plan technique...

L'affichage graphique nécessite le transfert de données entre le processeur et la mémoire d'image. L'acquisition du signal requiert une communication rapide entre l'interface de conversion analogique/numérique et la mémoire de la machine. Le traitement des données suppose l'accès du CPU ou des processeurs spécialisés aux données en mémoire et sur disque. L'ensemble de ces contraintes amène tout naturellement à grouper les fonctions dans une machine unique: la *station de travail*.

2.2. ...mais pas forcément sur le plan écologique.

La station de travail pour traitement du signal a l'inconvénient d'être une machine spécialisée. Associant calcul, graphique et entrées-sorties numériques, elle se trouve à l'intersection de leurs marchés respectifs (Fig. 1).

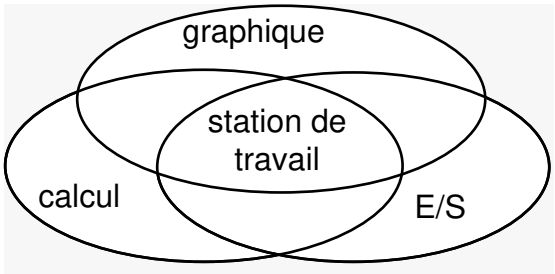

Fig. 1. Marchés des trois composantes constitutives de la station de travail de traitement du signal.

Il en résulte que ces matériels sont chers et d'évolution lente. Une configuration qui se trouve être optimale à un instant donné a peu de chances de le rester, face à l'évolution plus rapide de matériels moins spécialisés et dont le marché est plus vaste.

2.3. La compatibilité logicielle.

Le problème de la compatibilité logicielle face à l'évolution des matériels n'est pas nouveau. Il se pose cependant de façon plus aigüe avec le développement des machines à interface graphique interactive. Un programme qui, autrefois, aurait assuré un simple calcul, tend aujourd'hui à comporter une part graphique et interactive. Il devient ainsi plus complexe et plus étroitement lié au matériel, ou au système de fenêtrage.

2.4. Les standards ?

X-windows apporte un élément de réponse au problème précédent, mais cette solution est peut-être moins satisfaisante qu'elle ne paraît. D'une part du fait qu'elle n'est pas unique ("the nice thing about standards is that there are so many to choose from..."). D'autre part du fait de son évolution (certains ont en mémoire le passage de X-10 à X-11, versions incompatibles du même "standard"). Enfin du fait que les bibliothèques offertes par certains constructeurs pour compléter la fonctionnalité du standard restent leur propriété, et compromettent ainsi de façon sournoise la portabilité des logiciels qui les incorporent.

2.5. Les micros

Les micro-ordinateurs bénéficient d'une évolution plus rapide, ce qui fait qu'ils offrent aujourd'hui une interface-utilisateur de meilleure qualité. Les logiciels sont plus nombreux et moins chers que sur les stations de travail. L'intégration logicielle qui permet le "couper-coller" entre éditeurs de texte, de dessins, les tableurs, etc. est particulièrement utile, surtout dans la mesure ou elle permet aussi d'inclure les résultats expérimentaux. Si l'on conçoit le métier de chercheur ou d'ingénieur comme une *production de documents* (articles, rapports, etc.), l'utilité de la "chaîne de production" menant de l'expérimentation à la publication apparaît clairement.

Il faudrait donc, ou que les stations de travail offrent des fonctionnalités d'intégration logicielle (une évolution qui se dessine, mais avec retard), ou que l'interface-utilisateur du logiciel d'expérimentation se trouve sur micro-ordinateur. Cette dernière solution implique que la machine sur laquelle se font les traitements soit distincte de celle qui assure l'interface avec l'utilisateur. C'est la solution qui est retenue ici pour MapSignal.

3. MAPSIGNAL

MapSignal est un logiciel sur Macintosh qui permet la visualisation et manipulation de données se trouvant sur une autre machine.

3.1. Architecture

MapSignal dissocie les fonctions d'acquisition/traitement de signal et d'interface utilisateur, et les assigne à des machines différentes. La communication entre machines se fait par liaison série ou LAN. Il en résulte que le choix de la machine d'acquisition/traitement peut se faire sur les seuls critères de performance, coût, ou facilité de programmation, et que l'interface-utilisateur peut profiter des avantages qu'offrent un micro-ordinateur.

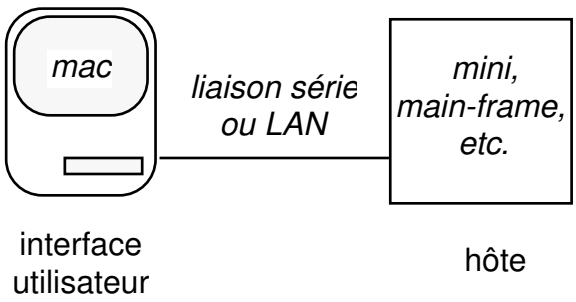

Fig. 2. Architecture d'édition "à distance" de MapSignal.

3.2. Protocole

Les protocoles de communication entre machines sont souvent source d'ennuis. MapSignal fait appel à un certain nombre de techniques pour les éviter.

1. relation maître-esclave

L'hôte est sous le contrôle exclusif de l'interface-utilisateur, qui est lui-même sous le contrôle de l'utilisateur. En aucun cas l'état de l'interface ne peut se trouver modifiée par une initiative de l'hôte (comme il peut arriver par exemple avec X-windows).

2. protocole "sans états"

L'hôte demeure dans un état unique qu'il ne quitte que transitoirement pour satisfaire une requête de l'interface. Il n'y a pas de multiples états, donc pas de risque de désynchronisation entre hôte et interface.

3. interpreteur de commande

S'il n'y a pas d'états, il n'y a pas non plus besoin d'un processus serveur tournant en permanence sur l'hôte. En pratique les requêtes de l'interface sont servies dans le cadre de l'*interpréteur de commande* habituel de l'hôte ("shell", interpréteur lisp, etc.). Un avantage est que l'interface de commandes de l'hôte demeure accessible à l'utilisateur (à travers une fenêtre de terminal incorporée à MapSignal). Un autre est que le protocole est robuste et facile à déboguer.

4. format des échanges

Les échanges d'information se font à l'aide de caractères ascii affichables (pas de caractères de contrôle, pas de bit de parité). Les requêtes et réponses utilisent un format lisible par l'homme (nombres en ascii), sauf pour les données graphiques.

5. code graphique "min-max"

Un signal est affiché sur l'écran du Mac sous la forme de son *enveloppe*. Celle-ci se matérialise graphiquement par un certain nombre de traits verticaux, à la résolution de l'écran. Ces traits sont définis par les ordonnées de leurs extrémités, qui dépendent simplement des *min* et *max* de groupes de k échantillons de signal (k est fonction de l'échelle horizontale):

$$a_n = \max_{i \in [kn,\, k(n+1)[} S_i$$

$$b_n = \min_{i \in [kn,\, k(n+1)[} S_i$$

Un algorithme spécial permet des échelles fractionnaires. Pour assurer la continuité visuelle (rompue lorsque $a_n < b_{n+1}$ ou $b_n > a_{n+1}$) ces valeurs sont remplacées avant affichage par:

$$A_n = \max\left(a_n, \frac{a_n+b_{n-1}}{2}, \frac{a_n+b_{n+1}}{2}\right)$$

$$B_n = \min\left(b_n, \frac{b_n+a_{n-1}}{2}, \frac{b_n+a_{n+1}}{2}\right)$$

Ces données graphiques sont calculées sur la machine hôte, codées sur 4 octets et envoyées à l'interface utilisateur qui les décode et s'en sert pour tracer l'enveloppe du signal. Le résultat graphique est identique à celui qui serait obtenu si tous les échantillons étaient dessinés, mais le volume de données à transmettre est beaucoup plus faible.

Ainsi, ce ne sont pas les données numériques qui transitent entre l'hôte et l'interface, mais seulement leur *représentation graphique*. Cela permet à MapSignal d'offrir un temps de réponse tout à fait honorable malgré la capacité limitée d'une liaison série. La gestion des fenêtres (réaffichage, etc.) est purement locale et ne nécessite aucun transfert de données.

3.3. Mécanisme d'extension

MapSignal n'offre en propre que des fonctions de visualisation, mais il possède un mécanisme qui permet de d'incorporer de nouvelles fonctions. Un "éditeur de commandes" permet de constituer un menu de commandes. A chacune d'elles correspond une chaîne de caractères qui est envoyée à l'hôte lorsque la commande est choisie dans le menu. La chaîne de caractères peut comporter des *mots-clé* qui sont interprétés au moment de l'envoi, ce qui permet d'inclure dans la chaîne des informations dynamiques telles que les coordonnées du curseur, le nom du fichier signal affiché, etc.. L'utilisateur peut ainsi diriger interactivement l'exécution de programmes disponibles sur l'hôte, en particulier ceux qu'il aura réalisés lui-même. Puisque l'affichage et l'interaction sont pris en charge par MapSignal, ces programmes peuvent être simples et portables.

3.4. Mecanisme d'interrogation de paramètres

Avant l'affichage d'un signal, MapSignal interroge l'hôte pour obtenir des informations sur ce signal (nombre d'échantillons, etc.). L'hôte envoie les informations qu'il possède: celles qui manquent sont remplacées par des valeurs par défaut. L'hôte a aussi le choix de la méthode par laquelle il obtient une information. Par exemple, la taille en échantillons du fichier signal peut provenir d'un "header", du système d'exploitation, ou d'un programme qui mesure explicitement la taille du fichier. MapSignal peut ainsi accomoder des formats de données très divers.

3.5. Moniteur d'acquisition

Une fonction originale mérite d'être signalée. Il s'agit d'une *fenêtre-moniteur d'acquisition de données analogiques.* Associée à un programme d'acquisition continue sur l'hôte, qui utilise un *buffer circulaire* (buffer contenant en permanence les n dernières secondes de données), elle

affiche son contenu. Ce contenu est en constant renouvellement au rythme de l'acquisition de nouvelles données, ce qui se reflète par l'image du signal qui "flotte" de droite à gauche à travers l'écran. L'utilisateur peut ainsi "geler" l'acquisition au moment opportun. On évite ainsi les problèmes de "timing" dus au mauvais alignement temporel, ou les distortions dues au mauvais réglage du niveau.

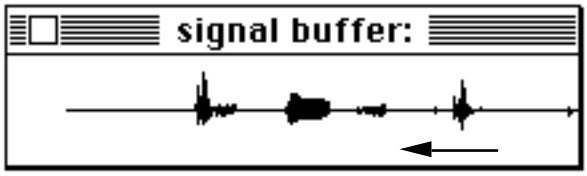

Fig. 3. Moniteur d'acquisition montrant le contenu d'un buffer circulaire.

L'acquisition continue est plus conviviale pour un locuteur que celle, discontinue, qu'imposent les solutions classiques.

3.6. Implémentation

MapSignal est écrit en C sur le Macintosh. Il nécessite deux commandes simples sur l'hôte, écrites elles aussi en C. Le programme est disponible gratuitement à fins de recherche. La distribution comprend le programme Macintosh, les sources des commandes hôte en C, et la documentation.

4. COMPARAISON AVEC X-11

L'idée principale défendue dans cet article est qu'on a intérêt à dissocier interface-utilisateur et traitement (calcul, entrées-sorties) et à les attribuer à des machines différentes. Cette idée sous-tend celle de "frontal" utilisée pour les bases de données ou le calcul scientifique (Mathématica). Elle se retrouve aussi dans X-windows, qui permet à un programme tournant sur une machine d'interagir indifféremment avec toute console du réseau. X-windows est de plus en plus répandu, mais il souffre des inconvénients suivants:

a) On ne peut associer que des machines possédant une version compatible de X-windows. Une mise à niveau partielle peut tout désorganiser, par ailleurs certaines machines de calcul n'offrent pas X. On perd ainsi dans la pratique une part de la liberté de choix qu'est censée apporter la dissociation des fonctions.

b) X-windows est gourmand en ressources (mémoire, disque).

c) X-windows ne permet n'offre pas un environnement logiciel aussi complet que ceux disponibles sur les micro-ordinateurs (cette situation change avec l'apparition des émulateurs X pouvant s'intégrer dans l'environnement logiciel d'un micro).

En comparaison, le modèle de communication sur lequel est fondé MapSignal pose peu de conditions quant à la machine hôte ou la nature de la liaison. Il faut cependant reconnaître que les contraintes de protocole que sont la relation maître-esclave stricte et l'absence d'états impliquent une communication moins riche que celle offerte par X-windows.

CONCLUSION

Le problème de la portabilité des logiciels de manipulation du signal peut trouver un élément de réponse dans la dissociation des fonctions de traitement du signal et d'interface utilisateur entre machines distinctes. La communication entre machines que nécessite cette solution peut se faire de façon simple et robuste dans le contexte de l'interpréteur de commande (shell) de la machine hôte. Conçu selon ces principes, le logiciel MapSignal permet la visualisation et la manipulation de données de signal se trouvant sur des machines très diverses. Sa conception fait appel à des idées originales qui peurvent être utiles dans d'autres contextes.

REMERCIEMENTS

Une partie de ce travail à été effectué lorsque l'auteur était à *ATR Interpreting Telephony Research Laboratories* dans le cadre d'une bourse du programme STP des Communautés Europénnes. Il tient à remercier ATR de son hospitalité.

BIBLIOGRAPHIE

[1.] de Cheveigné, A., M. Abe and S. Doshita (1985), "The human interface of a speech work-station", Studia Phonologica. *XIX*, 18-26.

[2.] de Cheveigné, A. (1989), "The MapSignal remote speech editor", *Proc. Acoust. Soc. Japan Automn meeting*, 383-384.

[3.] de Cheveigné, A. (1990), "The MapSignal remote speech editor", ATR Technical report TR-I-0137.