

### ***9.8.7 Séquenceur et total recall***

A l'origine, le terme de "total recall" désignait une fonction destinée à mémoriser un ensemble de réglages sur une console de mixage (position des faders, des mutes...) afin d'en autoriser le rappel ultérieur (récupération instantanée d'un ancien mixage). Par extension, l'habitude fut prise de l'as-

socier à toute procédure automatisée de mémorisation/rappel d'un ensemble d'informations numériques.

Ainsi, le musicien informatisé parle également de total recall, ce qui pour lui correspond à la mémorisation et au rappel de l'ensemble du contenu des mémoires de ses instruments. Un processus généralement pris en charge par un éditeur bibliothécaire, parfois par un séquenceur (Cubase + Studio Module), voire par un éditeur bibliothécaire et un séquenceur de même marque (Vision + Galaxy), fonctionnant en étroite liaison.

Au cours de cette section, nous vous proposons d'aborder ce processus sous un angle radicalement différent, puisqu'il va s'agir de réaliser soi-même un total recall, à l'aide d'un simple séquenceur. Schématiquement, la méthode consiste à écrire les messages de dump request sur une piste, pour enregistrer sur une autre les dumps reçus consécutivement à ces requêtes (mémorisation), et les renvoyer ultérieurement aux instruments (rappel).

### ***Notions élémentaires***

Les quelques paragraphes qui suivent reprennent quasi intégralement un article publié dans un numéro hors série du magazine *Keyboards*, consacré à l'informatique musicale. Si les notions abordées sont pour partie développées plus loin au cours de cette section, ou si certaines d'entre elles l'ont déjà été, elles permettront, sous cette forme condensée, à ceux pour qui le total recall n'évoque rien de bien précis, de se familiariser avec cette procédure.

Qui n'a jamais rêvé de presser un simple bouton pour qu'automatiquement un "instantané" de son home studio soit mémorisé, permettant ainsi de retrouver ultérieurement sa séquence en l'état, avec les bons sons, les bons effets, le bon mixage ? Le point sur l'un des plus grands fantasmes du musicien informatisé...

Comme le disait Wally Badarou lors d'une interview accordée au magazine *Keyboards* "le MIDI, et plus particulièrement le séquenceur, constitue un moyen idéal de cristalliser l'inspiration, de tout conserver, mais surtout, de pouvoir réexploiter cette matière première, ce que ne permet pas un enregistrement audio. Pour le compositeur, c'est une formidable libération". Si l'on pousse ce principe à l'extrême, au moment de revenir sur un morceau (qu'il s'agisse de le remixer, de le faire écouter à un client, de poursuivre son travail...), recharger la séquence idoine ne suffira pas. Théoriquement, il faudrait aussi récupérer les sons, les échantillons, le mixage, cette fameuse réverbération sur la caisse claire, cette compression sur la voix, etc. En gros, le problème se pose dans les termes suivants : en sauvegardant une séquence, comment mémoriser parallèlement les réglages de tous les appareils de son studio, pour retrouver cet environnement intact une heure, un mois ou un an plus tard ?

Avant de nous pencher sur le matériel audio (tables de mixage, périphériques...), commençons par examiner le cas des synthétiseurs. Si vous ne les programmez pas, si vous ne chargez jamais de nouvelles banques, bref, si la mémoire de vos instruments contient toujours les mêmes sons, le processus brille par sa simplicité. A partir de votre séquenceur, vous n'aurez qu'à envoyer des program changes à chacun de vos instruments. Ainsi, lorsque vous déciderez de ressortir du placard un morceau sauvegardé quelques jours auparavant, les bons sons répondront à l'appel. Enfantin, efficace, mais néanmoins limité.

Pourquoi limité ? Parions que vos tiroirs regorgent de disquettes pleines de banques pour vos synthétiseurs, et qu'en fonction de vos goûts, des titres composés et de l'humeur du jour vous chargez différents sons dans leur mémoire. Du coup, ce fameux program change qui, naguère, sélectionnait un piano correspond peut-être aujourd'hui à un saxophone.

On pourrait fort bien imaginer que les synthétiseurs, pour sauvegarder les sons qu'ils contiennent, soient tous munis de lecteurs de disquettes ou autres

mémoires de masse (c'est d'ailleurs le cas de certains d'entre eux). Ainsi, en admettant que votre home studio héberge dix de ces machines, il vous suffirait, pour chaque morceau, de sauvegarder sur disquette les sons correspondants. Bien qu'imparable, la solution manque cruellement d'élégance, tant sur le plan des manipulations (dix opérations manuelles de sauvegarde) que sur celui du budget (dix lecteurs de disquettes là où un seul suffirait).

Mais au fait, puisque tous ces synthétiseurs sont raccordés au séquenceur par l'intermédiaire de liaisons MIDI, ne serait-il pas plus satisfaisant intellectuellement, en fin de séance, de prendre une sorte de photographie de son studio, d'aller directement chercher depuis l'ordinateur le contenu des mémoires de chacun des synthétiseurs, pour le leur renvoyer au début de la séance suivante ? Peu importe qu'entre ces deux séances les mémoires en question aient été modifiées : notre piano de tout à l'heure, après avoir remplacé le saxophone dans la RAM, répondra présent lorsque le séquenceur sollicitera ses services en transmettant au synthétiseur le program change adéquat.

Aller chercher le contenu de la mémoire d'un synthétiseur équivaut à lui transmettre via MIDI une demande qualifiée de dump request. En retour, l'instrument renvoie les données demandées sous forme de dump. Si, pour procéder à ce genre d'échanges, tous les synthétiseurs emploient le même vocabulaire (à savoir les messages exclusifs), chacun d'eux parle malheureusement sa propre langue. Ainsi, une demande adressée à un JD-800 Roland ne signifiera rien pour un SY99 Yamaha. Le logiciel en charge du total recall (rappelons que c'est ainsi que nous désignerons ce processus automatique de mémorisation/rappel du contenu des mémoires de tous les appareils du studio) devra donc parler autant de langues que d'instruments avec lesquels dialoguer. Qui plus est, pour qu'une conversation puisse avoir lieu, il faudra qu'ordinateur et synthétiseur soient raccordés de façon bidirectionnelle (de l'ordinateur vers le synthétiseur, pour l'émission du dump request depuis le logiciel, et du synthétiseur vers l'ordinateur, pour la réception du dump).

Le MIDI, par son concept unidirectionnel (un câble ne véhicule des informations que dans un seul sens), se prête difficilement à la mise en réseaux d'instruments. Dès qu'il est question, non plus d'envoyer des données d'un appareil source vers plusieurs appareils destination (cas typique du séquenceur), mais d'établir des communications plus complexes, entre n'importe quelles machines d'une configuration, cela prend des allures de casse-tête. Dommage que Lone Wolf et son MidiTap n'aient pas rencontré plus de succès (un concept dont l'étude fait l'objet de la section 3.6, dans le premier tome). Le système permettait en effet d'intégrer le MIDI à l'intérieur d'un réseau de type LAN (Local Area Network).

Avant de prétendre exploiter un logiciel de total recall, vous devrez le renseigner sur la composition de votre environnement de travail. Il suffira généralement de parcourir une liste de drivers livrés avec le programme, pour sélectionner ceux correspondant à ses synthétiseurs. Ces drivers (encore appelés profiles, templates..., ou pilotes, en français, ou encore configurations) contiennent toutes les informations nécessaires pour que le logiciel puisse converser avec les machines (syntaxe des messages de dump request à envoyer, etc.). Si par malheur l'un de vos instruments, trop rare ou trop récent, ne figurait pas dans la liste, il vous faudrait créer vous-même son driver (donc parfaitement maîtriser les SysEx), à moins d'avoir la patience d'attendre son éventuel ajout à la liste, lors d'une prochaine mise à jour du logiciel.

Vous l'aviez compris, sauf à aimer brancher et débrancher sans arrêt ses cordons MIDI, il conviendra de disposer d'autant d'entrées que de synthétiseurs, puisque, tour à tour, le programme aura besoin d'être raccordé à chacun d'eux de façon bidirectionnelle. En la matière, les solutions sont multiples. A partir d'un ordinateur équipé d'une entrée/sortie MIDI (d'origine, comme sur l'Atari, ou par adjonction d'une interface, comme sur le Mac ou le PC), on pourra faire l'acquisition d'un patch externe (Roland A-880, Sycologic, Akai ME80P...). Pour chaque instrument, on programmera l'une des mémoires du patch de façon à établir une liaison bidirectionnelle

avec l'ordinateur. Pour bien faire, le patch utilisé devra réagir aux messages de changement de programme. Le logiciel de total recall aura alors la possibilité de reconfigurer automatiquement le réseau pour chaque instrument : il transmettra par exemple un premier program change au patch, correspondant à une liaison bidirectionnelle avec le premier synthétiseur (dont il ira chercher le contenu de la mémoire), transmettra ensuite un deuxième program change, correspondant à une liaison bidirectionnelle avec le deuxième synthétiseur (dont il ira chercher le contenu de la mémoire), et ainsi de suite.

A l'issue de cette procédure automatisée, nous nous retrouverons avec un seul et unique fichier, constitué du contenu des mémoires de tous les instruments. Au moment de renvoyer ce fichier, au début de la séance suivante, signalons que les liaisons bidirectionnelles ne seront plus d'aucune utilité. A la manière d'un séquenceur, qui arrose de messages l'ensemble du réseau (chaque instrument ne réagissant qu'à ceux transmis sur le ou les canaux le concernant), le logiciel de total recall enverra les dumps à tout le monde (chaque instrument ne prenant en considération que celui le concernant).

En lieu et place d'une simple interface MIDI, dotée d'une seule entrée/sortie communiquant avec un patch externe, un ordinateur comme le Macintosh pourra se voir adjoindre un boîtier mixte, faisant à la fois office d'interface et de patch. Citons les Studio 3 et Studio 5 d'Opcode (2 entrées et 6 sorties pour le premier de ces deux modèles, 15 entrées et 15 sorties pour le second), la MIDI Time Piece de Mark of The Unicorn (8 entrées, 8 sorties), etc.

En dehors des patches externes et des interfaces/patches pour Macintosh, on rencontre un certain nombre d'interfaces exclusivement prévues pour fonctionner avec les logiciels de la même marque. Ainsi, les racks Steinberg SMP24 et SMP2, en se raccordant au port imprimante de l'Atari, ajoutent entre autres deux entrées et quatre sorties MIDI. Comme nous venons de

le dire, a priori, seuls les programmes développés par la firme pourront y accéder (Cubase, Satellite, les Synthworks...). Pour en terminer avec cette énumération des patches et interfaces, on notera au passage que rien n'empêche de mélanger plusieurs de ces éléments.

Actuellement, sauf exception, seuls certains bibliothécaires universels (peu nous importe ici qu'ils soient ou non éditeurs) disposent d'une fonction de total recall. Citons des programmes comme X\_oR (Dr.T's), Galaxy Plus (Opcode), etc. A défaut d'énumérer les caractéristiques de tous ces produits, vous savez désormais qu'avant de choisir l'élus de votre coeur il faudra d'abord vérifier qu'il existe des drivers pour chacune de vos machines (ou de armer de courage pour programmer les vôtres), puis vous assurer de la capacité du logiciel à automatiser la gestion des connexions (que ce soit au travers de patches, par envoi de program changes, et/ou d'interfaces diverses).

Pour faire communiquer plusieurs logiciels en MIDI au sein d'un même ordinateur, notamment séquenceurs et éditeurs bibliothécaires, qu'ils soient universels ou dédiés, chaque constructeur ou presque y va de son système d'exploitation multitâche : MROS chez Steinberg, SoftLink chez Emagic, MPE chez Dr.T's, OMS chez Opcode, FreeMIDI chez MOTU... Même si certains d'entre eux rivalisent d'intelligence (possibilité, alors qu'une séquence tourne, de balayer une bibliothèque de sons depuis un éditeur, puis, toujours depuis cet éditeur, d'enregistrer le son sélectionné dans le séquenceur), ne serait-il pas judicieux d'intégrer une fonction de total recall dans un Vision, un Performer, un Notator Logic ou un Cubase ?

Dans un Cubase... C'est exactement l'idée sur laquelle a planché Philippe Goutier, auteur des Synthworks, en programmant Studio Module. Ce logiciel vient se greffer sur le séquenceur, pour le moment dans sa version Atari, en attendant que les adeptes des Macintosh ou des PC puissent en bénéficier.

Une fois Studio Module dûment renseigné sur vos appareils, et après qu'il a été chercher le contenu de leur mémoire, la vie devient paradisiaque. En effet, plutôt que de régler chaque piste sur un canal et un port MIDI, vous

vous retrouverez directement confronté à des noms d'instruments, sous forme d'un menu pop-up. Par exemple, choisir le D-50 dans ledit menu aura pour effet de positionner automatiquement la piste sur le canal et le port correspondant à cet instrument. Si, sur Macintosh en particulier, d'autres logiciels sont capables d'un tel prodige (notamment chez Opcode, avec OMS), le couple Cubase + Studio Module optimise le principe. Effectivement, suite à un double clic sur la case program change d'un pattern ou d'une piste apparaîtra une fenêtre de type banque, dans laquelle il ne restera plus qu'à choisir par son petit nom le son désiré. Studio Module a bien d'autres tours dans son sac, qui dépassent largement le cadre de cet ouvrage.

Certains appareils, qu'il serait pourtant fort utile d'intégrer à une procédure de total recall, sont réfractaires au principe. C'est notamment le cas des échantillonneurs, à cause de la taille des données manipulées, trop imposantes pour être véhiculées via MIDI. A moins d'avoir la chance de posséder un sampler qui, à réception d'un program change, charge un volume à partir d'un quelconque disque dur ou optique, vous voici contraint et forcé d'écrire très lisiblement sur les étiquettes de vos mémoires de masse.

Du point de vue du total recall, tout ce qui comporte des mémoires au contenu transmissible via MIDI se gère très exactement comme les synthétiseurs, à commencer par les processeurs d'effets numériques, mais aussi certains périphériques analogiques, en particulier des égaliseurs (Digitech, Waldorf...). En fait, la grosse difficulté se situe au niveau du mixage, puisqu'en dehors des faders et des mutes, rares sont les consoles MIDI pourvues d'une mémorisation de tous les paramètres (correcteurs, départs effets, panoramiques...). Sauf à investir dans une Yamaha (DMP7, DMP11, DMP9-16, ProMix 01), une MOTU (7'S), une Fostex (DMX1000 + Mixtab), une General Music (Sound Engineer) et quelques autres, voici nos home studios fort dépourvus.

En dépit des incessants progrès de l'informatique musicale et des équipements MIDI, le véritable total recall reste encore aujourd'hui réservé à une

élite, et ce à deux titres. D'une part, les connaissances qu'il requiert en rebutent plus d'un (parfaite maîtrise de la connectique, notions avancées en matière de SysEx...), d'autre part, le matériel compatible avec cette philosophie de travail n'est pas encore suffisamment répandu dans le domaine de l'audio. Pour bénéficier d'un total recall vraiment transparent, sans doute faudra-t-il attendre que toutes les machines sans exception soient numériques, et qu'un ingénieur constructeur développe un super protocole de transfert véhiculant au sein d'un seul câble et dans toutes les directions à la fois le MIDI, l'audio numérique, les codes temporels et les autres....

### ***Introduction à la réalisation d'un total recall sur séquenceur***

Suite à ces quelques rappels, entrons dans le vif du sujet, en abordant à proprement parler la programmation d'une procédure de total recall. Celle-ci, lors d'un enregistrement sur séquenceur, va traiter les données d'un réseau d'appareils MIDI utilisés dans le cadre de la réalisation d'un morceau. Fonctionnellement, il s'agit donc d'une méthode visant à stocker le contenu des mémoires transmissibles par l'intermédiaire de messages exclusifs (sons, effets...) en les enregistrant sous forme de dump sur l'une des pistes du séquenceur. Lors de la séance de travail suivante, afin de récupérer "en l'état" son environnement initial, ces dumps seront "redirigés" vers les différents instruments.

Dans certaines limites, un résultat similaire s'obtient en transmettant en début de séquence à chaque instrument MIDI un program change en fonction du numéro de patch utilisé. Cependant, comme il a été dit précédemment, la fiabilité d'un tel système suppose que les mémoires RAM des instruments ne subissent jamais la moindre modification. Admettons, en effet, qu'un séquenceur transmette à un générateur de son le message de program change n° 10 en vue de sélectionner un timbre de piano, et qu'ultérieurement l'utilisateur décide de modifier la banque interne de ce générateur (l'éditer, la remplacer par une autre à l'aide d'un bibliothécaire, etc.).

En renvoyant la séquence, bien que le program change transmise adresse toujours le bon emplacement (c'est-à-dire la case mémoire n° 10), le son y résidant ne correspondra plus au piano initial, mais au son 10 de la nouvelle banque. Une automation basée sur un tel principe ne pourra donc s'appliquer qu'à une configuration mémoire non sujette à changements.

### ***Les concepts de base : enregistrement et lecture de SysEx***

La procédure de total recall MIDI se déroule en deux étapes : la mémorisation des données des instruments dans le séquenceur, et leur renvoi ultérieur. Sur un plan pratique, l'étape de mémorisation consiste à programmer les demandes de dump (dump request) sur une piste que nous nommerons ***piste request***. Lors de sa lecture en solo (afin d'éviter tout conflit entre les messages exclusifs et les autres messages MIDI de la séquence), une seconde piste (***piste data***) se charge alors d'enregistrer les données renvoyées par les instruments (dumps). Grâce à cela, sons et séquences sont sauvegardés conjointement. Pour retrouver ultérieurement les données d'origine (au début de la session suivante), il ne reste plus qu'à lire la piste data en solo de manière qu'elle retransmette les dumps aux instruments.

Il est impératif que le séquenceur accepte l'enregistrement de messages exclusifs en temps réel, et souhaitable qu'il en autorise la programmation pour permettre à l'utilisateur d'écrire la piste request. Cependant, dans le cas contraire, rien n'empêche de programmer cette piste à l'aide d'un séquenceur adapté, pour l'exporter via MIDI Files vers le séquenceur utilisé à des fins de total recall.

L'enregistrement de dumps s'accompagne du respect d'un certain nombre de règles élémentaires. Il conviendra, si ces options existent, de valider l'enregistrement des SysEx et de désactiver la fonction d'écho MIDI (MIDI thru), afin de ne pas renvoyer les données exclusives reçues. D'autre part, les procédures de dump étant généralement liées à un canal spécifique, nous

veillerons à ce que chaque instrument conserve le même numéro, pour des raisons évidentes d'organisation.

### ***Handshaking et patch MIDI***

Pour transmettre les demandes de dump à partir d'un séquenceur, il est nécessaire que chaque instrument y soit connecté tour à tour en mode handshake (prise MIDI-out du séquenceur vers la prise MIDI-in de l'instrument pour l'envoi des messages de request, et réciproquement pour la réception des dumps). C'est pourquoi l'utilisation d'un patch MIDI réagissant aux program changes est souhaitable. En effet, en programmant une configuration handshake par instrument, il suffira de commuter de l'une à l'autre directement du séquenceur, grâce à l'envoi du program change adéquat.

Cette manipulation n'a d'intérêt qu'en réception (lecture de la piste request et enregistrement de la piste data), puisqu'en émission (lecture de la piste data) le patch MIDI se contente d'acheminer les dumps vers l'ensemble des instruments (prise MIDI-out du séquenceur vers les prises MIDI-in de tous les instruments, chacun d'eux ne prenant en compte que ses propres SysEx). D'où la nécessité de piloter ce patch MIDI par l'intermédiaire d'une troisième piste indépendante : la *piste patch*.

### ***Une méthode d'analyse***

Dans l'élaboration d'un total recall, la première étape consiste à définir très précisément toutes les catégories de données à transférer par SysEx entre le séquenceur et les instruments, avant de s'interroger sur le meilleur moyen de programmer ces échanges. Pour illustrer cette méthode d'analyse, nous allons étudier les principes de total recall appliqués à un réseau MIDI regroupant un générateur de son monotimbral ainsi qu'un générateur de son multitimbral.

### ***L'automation d'un générateur de son monotimbral***

Admettons qu'un générateur de son monotimbral soit constitué de 64 sons en ROM (numérotés de 1 à 64) et de 64 sons en RAM (numérotés de 65 à 128). Nous savons que, dès sa sélection (et sauf exception, comme pour la série Casio CZ, mentionnée au cours de la section 9.7.2), un son est dupliqué de l'emplacement mémoire auquel il réside vers un buffer d'édition accessible via MIDI. Pour en enregistrer le dump sur la piste data du séquenceur, il suffit donc a priori de transmettre un message de dump request à l'attention du buffer d'édition par l'intermédiaire de la piste request. A l'inverse, en lecture du séquenceur, le contenu de la piste data (message de dump) est automatiquement renvoyé vers ce même buffer.

Toutefois, cette procédure n'est viable qu'à condition de n'employer qu'un seul et unique son sur toute la durée du morceau. Or, dans la réalité, il est fréquent d'en changer par incorporation de program changes à l'intérieur d'une séquence (son X pour l'introduction, son Y pour les couplets, son Z pour les refrains...). L'une des solutions à ce problème consiste à substituer au message de dump request du buffer d'édition un message de dump request de l'intégralité de la RAM, permettant ainsi d'utiliser autant de sons que nécessaire. Malheureusement, cette technique a pour inconvénient majeur de stocker des données inutiles dans la mémoire du séquenceur.

En admettant que les sons 65 à 68 de notre générateur monotimbral soient mis à contribution au cours d'un morceau, la piste request dispose de différents moyens, en fonction des messages exclusifs implantés par l'instrument, pour les transférer vers la piste data du séquenceur. A savoir :

- Envoi d'un message de dump request destiné à aller chercher en une seule passe les sons 65 à 68 (c'est par exemple le cas de certains instruments Roland, capables de transférer directement n'importe quel bloc de données caractérisé par une adresse de départ et une taille, sans qu'il transite par le buffer d'édition).

- Envoi de quatre messages de dump request successifs, chacun d'eux intégrant le numéro de son à aller chercher (c'est par exemple le cas des instruments Casio CZ, capables de transférer directement les données son par son, sans qu'elles transitent par le buffer d'édition).
- Envoi de quatre messages de dump request successifs destinés à aller chercher le contenu du buffer d'édition, le program change adéquat (65, 66, 67, 68) étant transmis préalablement à chaque demande de dump, de manière à transférer le son correspondant à l'intérieur du buffer d'édition.

Si les deux premières méthodes fonctionnent telles quelles lors de la phase de renvoi des données (les dumps enregistrés sur la piste data étant automatiquement retransmis), la troisième (ne dialoguant pas directement avec la mémoire, mais avec le buffer d'édition, utilisé à des fins de transfert en tant que mémoire temporaire) implique que l'instrument implante des messages MIDI lui intimant l'ordre d'écrire le son contenu par le buffer d'édition à tel ou tel emplacement de sa mémoire (il s'agira soit de véritables messages d'écriture, soit d'une simulation du processus de stockage par appui "virtuel" via MIDI sur les touches write, store...).

Quoi qu'il en soit, si ce concept évite d'avoir à transférer l'intégralité de la RAM, il implique une certaine organisation, à savoir un emploi constant des mêmes cases mémoire pour l'ensemble des morceaux (65 à 68 dans cet exemple), étant donné l'incapacité d'un message de dump request à deviner les numéros des sons à aller chercher.

Dans la phase de rappel du total recall, qu'il s'agisse de travailler sur tout ou partie de la RAM (et non plus sur une mémoire temporaire), les données de l'instrument sont définitivement remplacées par celles que renvoie le séquenceur sous forme de dumps. En supposant qu'entre le transfert générateur de son/séquenceur (mémorisation des données) et le transfert séquenceur/générateur de son (renvoi des données) nous ayons décidé de changer le contenu de la RAM, les sons 65 à 68 de cette nouvelle RAM seront irrémédiablement écrasés par ceux de la piste data.

Pour remédier à cet écrasement de données, il est envisageable de construire une seconde piste data (data 1), enregistrée à partir de la même piste request, dont le rôle consistera alors à stocker les sons initiaux contenus par les instruments en début de session, avant d'y substituer ceux de la piste data (c'est-à-dire ceux du total recall). A l'inverse, en fin de session, elle récupérera d'éventuelles modifications apportées aux sons de cette piste data, avant de les remplacer par les originaux (ceux de la piste data 1). Plus complexe à gérer, ce procédé préserve la mémoire vive du générateur de son de tout écrasement de données, le séquenceur officiant en tant que mémoire temporaire pendant toute la durée d'une session (à la fin de laquelle la piste data 1 pourra être effacée).

A condition de n'utiliser qu'un seul son au cours du morceau, nous traiterons les mémoires ROM de manière identique aux mémoires RAM, selon la méthode du buffer d'édition. Dans le cas contraire, lors de l'utilisation de plusieurs sons en ROM au cours du même morceau, nous aurons le choix entre l'envoi de program changes destinés à les sélectionner (1 à 64 dans cet exemple), ou la recopie préalable de ces différentes sonorités de la mémoire ROM vers la mémoire RAM, permettant de procéder comme précédemment.

En résumé, voici trois méthodes de total recall s'appliquant à un instrument MIDI à mémoire unique (par opposition aux générateurs multitimbraux, avec lesquels plusieurs mémoires sont mises à contribution simultanément) :

- Transfert du buffer d'édition (un seul son est utilisé au cours du morceau).
- Transfert de la RAM en totalité (plusieurs sons sont utilisés au cours du morceau ; l'instrument ne permet pas de transférer la RAM autrement que dans son intégralité).
- Transfert d'un certain nombre de sons situés à des emplacements mémoire fixes (plusieurs sons sont utilisés au cours du morceau ; l'ins-

trument permet de transférer successivement ou simultanément un certain nombre de sons en RAM).

### ***L'automation d'un générateur de son multitimbral***

Si tous les principes énoncés précédemment s'appliquent également aux générateurs de son multitimbraux, leur gestion est beaucoup plus complexe, du fait qu'ils mélangent de manière interactive deux types de mémoire : la mémoire de configuration (répartition des différents timbres), et les mémoires de sons (timbres utilisés par cette configuration). Cette interaction est totalement étrangère au séquenceur, incapable de deviner les numéros des sons appartenant à telle ou telle configuration. Poursuivons donc notre analyse, en nous basant sur un générateur de son multitimbral renfermant 64 configurations en RAM, chacune constituée d'un maximum de 8 sons parmi les 128 disponibles (64 en ROM, numérotés de 1 à 64, et 64 en RAM, numérotés de 65 à 128).

A condition d'utiliser une configuration unique tout au long du morceau, la méthode de transfert par buffer d'édition est viable. Toutefois, nous sommes confrontés à un ensemble composé de neuf mémoires (huit sons plus une configuration), par conséquent de neuf buffers. Or, à quelques exceptions près (de rares instruments autorisent le transfert de l'ensemble des buffers de sons en un seul bloc), aucun message de dump request ne permet de vider d'un seul coup ces neuf mémoires. En toute logique, nous transmettrons donc successivement neuf demandes de dump : la première à l'attention du buffer de configuration, et les huit suivantes à l'attention des buffers de sons.

Cependant, pour compliquer le problème, nombreux sont les instruments multitimbraux n'incluant qu'un seul et unique buffer de sons. Si l'instrument en question est capable de transférer individuellement ou en bloc un certain nombre de sons en RAM (l'inverse nous contraignant à transférer cette RAM dans son intégralité), nous déciderons arbitrairement d'associer

à la configuration huit numéros de sons fixes (par exemple, 65 à 72). Une restriction qui contraint l'utilisateur à se servir invariablement de ces mêmes emplacements.

En résumé, voici trois méthodes de total recall s'appliquant à un instrument MIDI multitimbral lorsqu'une seule configuration est sollicitée au cours du même morceau :

- Transfert du buffer d'édition de configuration et des buffers de sons associés (il existe un buffer d'édition pour chaque son lié à une configuration).
- Transfert du buffer d'édition de configuration et des mémoires de sons associées, les sons étant situés à des emplacements fixes (le buffer d'édition de son est unique, mais l'instrument permet de transférer successivement ou simultanément un certain nombre de sons en RAM).
- Transfert du buffer d'édition de configuration et de la RAM de sons en totalité (l'instrument ne permettant pas de transférer la RAM autrement que dans son intégralité).

Tout comme avec le générateur de son monotimbral, l'utilisateur pourra décider de changer de configuration en cours de morceau. Là encore, si cela est possible, nous procéderons par transfert direct en RAM, en choisissant en fonction des besoins (du nombre de changements) un certain nombre d'emplacements fixes (sur un instrument multitimbral huit voies, nous sélectionnerons par exemple les configurations 1 à 4, associées respectivement aux sons 65 à 72, 73 à 80, 81 à 88 et 89 à 96). Les procédures de transfert des configurations s'apparentent alors à celles que nous avons évoquées lors de l'étude des sons des générateurs monotimbreaux.

En résumé, voici deux méthodes de total recall s'appliquant à un instrument MIDI multitimbral, lorsque plusieurs configurations sont sollicitées au cours du même morceau :

- Transfert d'un certain nombre de configurations et de sons situés à des emplacements mémoire fixes (plusieurs configurations et plusieurs sons sont utilisés au cours du morceau ; l'instrument permet de transférer successivement ou simultanément un certain nombre de configurations ou de sons en RAM).
- Transfert de la RAM de configurations et de la RAM de sons en totalité (plusieurs configurations et plusieurs sons sont utilisés au cours du morceau ; l'instrument ne permet pas de transférer les RAM de sons et de configurations autrement que dans leur intégralité).

Ces différentes méthodes s'appliquent de manière similaire lorsque les sons de la configuration, en plus des configurations elles-mêmes, changent en cours de morceau.

Dans le sens inverse (du séquenceur vers les instruments), qu'il s'agisse d'instruments monotimbraux ou multitimbraux, les ordres d'écriture en RAM nécessitent parfois une déprotection préalable de la mémoire. Selon les cas, nous l'obtiendrons soit directement via MIDI, soit par l'intermédiaire de messages exclusifs simulant l'appui sur les touches de l'instrument. La programmation de ces différentes procédures fait l'objet d'une quatrième et dernière piste : la *piste send*.

Nous retiendrons de cette analyse que les messages exclusifs se prêtent mal à une généralisation. En réalité, chaque instrument représente un cas particulier. Les quelques directions que nous venons d'étudier sont loin d'être exhaustives, et ne sont fournies qu'à titre indicatif. De plus, chaque instrument acceptant généralement différentes "philosophies" de transfert, l'utilisateur développera son propre système d'automatisation en fonction de ses besoins. L'application commentée ci-dessous s'efforce, dans la mesure du possible, de mélanger les diverses techniques que nous venons d'aborder, afin de vous guider dans la réalisation d'un total recall.