Outils et pratiques du sonore dans le spectacle vivant

Groupe de travail interdisciplinaire

missionné par
l'Association Française d'Informatique Musicale

À l'initiative de la Compagnie Incidents Mémorables, un groupe de travail s'est constitué début 2006 afin de mener une étude visant à faire un premier état de l'art des pratiques et des outils rencontrés dans le domaine du spectacle vivant, et en particulier dans l'intersection de ce champ avec celui des dispositifs numériques temps réel pour la composition musicale.

Ses **objectifs** initiaux étaient de :

- Mettre en oeuvre un questionnement des pratiques, des outils et des métiers du sonore dans le contexte du spectacle vivant (théâtre, danse, concert étendu ou multimédia) pour en dégager les particularités.

- Établir un état-de-l'art des environnements logiciels existants pour la composition et l'interprétation du sonore dans le cadre du spectacle vivant.

- Dégager les prospectives de développement futur de tels environnements.

Cette étude a été rendue possible grâce au financement du groupe de travail par L'association Française d'Informatique Musicale.

L'étude s'est clôturée par la communication des activités du groupe de travail lors des Journées d'Informatique Musicale 2007 organisées en avril 2007 à Lyon.

périodes de création sur le plateau et en amont) de pouvoir avoir accès à toutes les fonctions disponibles dans le moteur, et de pouvoir fixer, puis modifier, les relations entre entrées (sources sonores, entrées physiques...etc...), processus de génération et de transformation, et sorties.

C'est à ce moment qu'intervient le *mapping*, qui consiste à établir une cartographie de l'évolution du matériau sonore, notamment par la mise en relation des gestes du régisseur et des interprètes avec les processus de transformation et de génération disponibles dans les moteurs audio utilisés. Cette phase est primordiale dans l'élaboration d'un environnement sonore en relation étroite avec les autres éléments du spectacle. C'est le *mapping* qui permettra le jeu, c'est-à-dire l'interprétation fine d'une écriture du son.

L'interface de création rend donc possible l'écriture de l'évolution temporelle du son et l'établissement des interactions, et permet ainsi de tisser des réseaux complexes de relations entre le sonore, les interprètes sur le plateau et les autres éléments du spectacle (images, lumière, scénographie, actionneurs, etc.)

Interface d'interprétation ou de conduite

L'interface de conduite ou d'interprétation doit alors, après le travail effectué sur l'interface de création, présenter le minimum d'information nécessaire au bon déroulement de la performance ou de l'exécution de l'œuvre.

Ici, l'ergonomie doit se concentrer sur la présentation des bonnes informations et des accès utiles, uniquement au moment opportun, pour ne pas noyer l'interprète ou le régisseur sous une masse de données dont la disponibilité est certes nécessaire durant la phase de création, mais dont celui-ci doit se passer durant l'exécution, pour ne se concentrer que sur les paramètres directement prégnants.

Cela implique donc que cette interface de conduite présente un système de pages différentes pour les différents moments du spectacle, dont chacune présentera les accès et les retours d'informations strictement nécessaires à l'interprétation. L'interprète/régisseur pourra alors se concentrer sur l'utilisation sensible de son medium en phase avec l'activité du plateau.

Il paraît également productif de proposer un retour d'information des paramètres de plus haut niveau que ceux employés dans les moteurs, présentant ainsi un caractère davantage descriptif qu'opérationnel, permettant alors une appréhension plus intuitive des actions générées et des paramètres contrôlés, un saut du niveau technique au niveau artistique.

On retrouve ici le rapport à l'instrument dont nous parlions plus haut, qui implique la fixation d'un rapport entre geste et production sonore, pour affiner la sensibilité et l'expressivité de l'interprète et, en quelque sorte, "maîtriser la technique pour la dépasser".

Protocole de communication

Il est indispensable, pour communiquer entre ces divers niveaux : moteur, interface de création, interface d'interprétation, d'établir un même protocole.

Ce protocole doit être suffisamment formalisé pour permettre aux divers niveaux de transmettre leurs informations tout en faisant une description, même sommaire, des fonctions auxquelles elles correspondent.

Il doit se baser sur une norme suffisamment courante pour être supportée par des logiciels différents, et ainsi permettre la communication avec autant de systèmes que possible.

Techniquement, deux protocoles, à notre connaissance, répondent à cette dernière condition :

- le MIDI, dont la popularité n'est plus à démontrer, mais qui accuse un grand nombre de limitations en terme de débit, de précision ou de description.
- le standard OSC, relativement répandu dans les logiciels et les interfaces audio professionnelles, efficace, rapide et permettant une description précise, mais dont l'ouverture est telle qu'elle pose le problème inverse du MIDI, à savoir que chaque implémentation génère son propre langage et ne peut donc plus communiquer qu'avec elle-même. À noter la structure hiérarchique (basée sur le système XML) permettant de décrire l’arborescence fonctionnelle des moteurs.

Outils existants

Il appert des réflexions précédentes qu'une telle pratique requiert des outils spécifiques, qui ne sont pas encore aboutis pour l'instant, voire n'existent pas.

Nous avons donc établi un tour d'horizon des logiciels et environnements audio utilisés dans le spectacle vivant. Ils se divisent en deux grands groupes : les logiciels commerciaux, fixés, et les environnements évolutifs.

Logiciels

- Dans le premier groupe, on pourrait citer comme le plus efficace et plus répandu le logiciel Live d'Ableton, dont la popularité croissante montre bien la pertinence dans le traitement en direct du sonore. Il semble cependant que, si ce logiciel est idéal pour le déclenchement de séquences sonores et dans une certaine mesure pour le traitement en direct, il montre assez rapidement ses limites dès lors que l'on cherche à mettre en oeuvre des dispositifs complexes de génération sonore ou d'interaction, par son absence de système sophistiqué de *mapping* et son utilisation exclusive du protocole MIDI.

À noter qu'à l'origine, le logiciel Live a été prototypé dans Max/MSP, puis ensuite refondu en C pour des raisons de performances.

- Il existe également plusieurs logiciels dédiés à la conduite sonore pour le spectacle vivant et destinés à un public de régisseurs : du plus simple au plus complexe : cricketsound, SeqCon, CueStation, Séquenceur d'Histoire de l'Odéon.

Tous ces logiciels ont des qualités certaines, présentent une très bonne fiabilité et sont particulièrement adaptés pour une régie "classique" de théâtre, consistant en un enchaînement de séquences déclenchées par des cues successifs correspondant à des événements du plateau.

Ils montrent cependant très rapidement lorsque l'on veut employer des dispositifs interactifs ou génératifs, que ce soient des processus *ad hoc* de génération sonore algorithmique, ou bien des interactions avec le plateau par caméra ou capteurs, ou encore des modes d'interfaçage alternatifs à la console MIDI.

- Jade serait peut-être le logiciel le plus adapté aux besoins définis plus haut. En effet, ce logiciel, développé sous Max/MSP, fonctionne avec une librairie de modules standard, à laquelle peuvent s'ajouter des modules développés par les utilisateurs. Il permet un accès configurable à tous les paramètres et des mappings assez précis. Cependant, la gestion des événements (cues) est un peu lourde à programmer, puisque basée entièrement sur un langage de script. De plus, il n'est plus maintenu.

Par contre, son architecture et ses éléments de base ont été refondus et "open-sourcés", donnant ainsi naissance au projet Jamoma, dont nous reparlerons plus bas.

Environnements

- Environnements modulaires "bas-niveau" : Max/MSP/Jitter et Pure Data. À notre connaissance, la plupart des productions professionnelles impliquant des processus génératifs et interactifs ont été développés dans ces environnements, dont la caractéristique principale est de permettre à-peu-près tout type d'interaction ou de relation entre un medium et un autre. La contrepartie de cette ouverture est le temps nécessaire au développement du moindre processus, et la tendance à une spécificité d'écriture qui rend difficile la réutilisation de processus génériques que l'on retrouve souvent d'un projet à l'autre.

- Un certain nombre d'environnements modulaires de plus haut niveau, donc plus simples et plus rapides à mettre en oeuvre, sont disponibles sur le marché : Plogue Bidule, AudioMulch, Usine, Processing.

Tous ont leurs particularités, points faibles et points forts, mais sont généralement orientés vers la production musicale expérimentale ou électronique, et montrent rapidement leurs limites lorsqu'il s'agit d'écrire l'interaction dans le temps et avec le plateau, ou de construire des conduites fiables.

- Enfin, les environnements développés dans Max ou Pure-data, dont on trouve pléthore de réalisations sur le web : UBC Toolbox, Theater Max, Quintet.net, MaWe, Roseß, MPTools, lloop.

- Sans oublier ceux développés par les membres du groupe de travail Peralvino (Olivier Pfeiffer), Z (Pascal Baltazar), Tape (Tom Mays) et l'environnement de Mathieu Chamagne.

Ici, de même, chacune de ces réalisations a ses spécificités, ses qualités et ses défauts, mais aucune d'entre elles ne peut prétendre, en l'état, à devenir un outil fiable et performant pour la création sonore pour le spectacle vivant. On constate en effet, pour quasiment tous ces environnements, un manque de généricité, une absence de protocole de communication avec l'extérieur du système et/ou un mode d'utilisation assez cryptique et éminemment personnel, puisque généralement lié à la pratique singulière du développeur.

Convergence et prospectives de développement

Protocole d'échange

Nous avons donc déduit de cet ensemble de réflexions qu'il semblait plus productif de s'orienter vers l'établissement d'un protocole plutôt que d'un produit fini. Le développement devrait privilégier un environnement ouvert, évolutif, tout en proposant des éléments d'assez haut niveau pour ne pas exiger de l'utilisateur des compétences trop élevées de programmation.

Ainsi peut-on éviter le risque majeur du produit fini : devenir obsolète au moment même où il est produit, puisque ne pouvant suivre l'évolution vertigineuse des pratiques.

Nous nous sommes donc orientés vers des communautés de travail ayant des objectifs similaires :

- Jamoma, dont l'objectif est de "produire une structure pour l'échange, la réutilisation et la transmission de patches Max". L'une des particularités intéressantes de ce système est son usage du protocole OSC.

Nous avons donc organisé une rencontre avec Trond Lossius, un des fondateurs de cette initiative, qui nous en a expliqué le fonctionnement et l'esprit, qui se trouvaient converger sur de nombreux points avec notre démarche.

- Nous avons également participé à une réunion de travail du groupe Integra, dont certains des objectifs sont similaires aux nôtres, à savoir la réflexion sur le développement d'un environnement audio basé sur le couple interface/moteur, mais dans un champ différent : la pérennisation et la création d'oeuvres musicales contemporaines "mixtes" (instrumentales et électroniques).

À noter que Integra et Jamoma réfléchissent ensemble à l'élaboration d'un espace de nommage générique, visant à la description des processus audio et des modes d'interaction, et basé sur le protocole OSC

- Nous avons enfin assisté à la réunion des développeurs OSC qui a eu lieu à l'IRCAM, à la fin du NIME'06. Cette réunion évoquait notamment l'élaboration d'un OSC 2.0 (sur le modèle du web 2.0), dont l'une des particularités intéressantes consisterait à ce que les systèmes soient dotés de dispositifs permettant de s'identifier lorsqu'ils apparaissent sur un réseau, et de signaler leur arborescence et, partant, les langages qu'ils comprennent et produisent c’est-à-dire leurs modes de description des données et leurs structures d'adressage. Cette évolution permettrait notamment de régler le principal problème actuel du protocole OSC, à savoir sa trop grande ouverture.

Niveaux d'utilisateurs

Il nous a semblé pertinent, dans la perspective d'un développement futur d'un tel système, de dégager trois niveaux d'utilisateurs :

- Utilisateur final : Régisseur, compositeur n'ayant pas ou peu de connaissance de la programmation informatique, et dont l'intérêt se situe dans la mise en oeuvre du sonore pour la composition et la régie. Cet utilisateur est cependant censé avoir consacré un minimum de temps à se former à l'utilisation de l'environnement, qui doit rester suffisamment ergonomique et clair pour en permettre une prise en main en quelques heures, même si l'exploitation de toutes les possibilités implique forcément d'avantage d'exploration.
- Développeur de modules : Compositeur ou régisseur ayant une certaine expérience de la programmation, et souhaitant intervenir sur les processus et algorithmes mis en jeu. Il faut alors fournir à cet utilisateur "intermédiaire" un "Software development kit" simple d'utilisation et clair, lui permettant dans un temps relativement court d'intégrer ses algorithmes comme de nouveaux éléments de la librairie de modules, sans forcément avoir à connaître ou comprendre le fonctionnement interne de l'environnement.
- Développeur d'environnement ou de moteur : Membre d'une équipe de développement d'environnement, il est censé connaître la majeure partie des tenants et aboutissants de la programmation de l'environnement et en assurer la maintenance et l'évolution. Il est en dialogue avec les développeurs de modules et leur fournit une "feuille de route" pour l'intégration de leurs modules.

Développement de la Maquette EVE
(Environnement Versatile Expérimental)

Plusieurs membres du groupe de travail ont donc consacré un temps de développement à élaborer une première maquette, visant à expérimenter concrètement et à mettre à l'épreuve par la pratique les prospectives de développement dégagées par l'étude.

La Maquette EVE se présente comme un logiciel-hôte pouvant accueillir des modules développés selon le protocole Jamoma. La partie commune de EVE présente un certain nombre de fonctionnalités génériques :

- mise en page des modules et classement dynamique par page
- gestion des états des modules (presets)
- gestion des « cues » (messages temporels de commande des paramètres des modules)
- gestion des sorties audio (traitement et multi-diffusion)
- gestion du matriçage des signaux audio et video

Fig. 2 : Un exemple de configuration de EVE.

D'autre part, la plate-forme propose d'établir une bibliothèque de modules, classée par développeur et appelée à croître au fur et à mesure des contributions. Une liste non exhaustive comprendrait les types de modules suivants :

- génération AUDIO : lecteurs de fichiers son, entrées externes (micros, sources live...), processus de synthèse divers (analogique, FM, granulaire...)
- traitements AUDIO : filtres, effets divers, plug-ins, traitements dynamiques... etc.
- analyse AUDIO : suivi d'enveloppe, analyse de timbre et de hauteur...

- génération VIDEO : lecteurs de media (films et images fixes), entrées externes (caméras, sources diverses...), synthétiseurs vidéo...
- traitements VIDEO : filtres, processus de composition d'image...
- analyse VIDEO : détection de formes et de gestes, suivi de couleurs et de mouvement...
- sorties VIDEO : gestion des sorties physiques (écrans) et virtuelles (plans 3D). La gestion de la vidéo utiliserait la technologie GPU...
génération et manipulation 3D : objets virtuels, dessin et écriture spatialisés, systèmes de particules...
- acquisition de données PHYSIQUES : capteurs de toutes sortes, interfaces de contrôle (controleurs MIDI, tablette graphique, Lemur...) ...
- commande d'actionneurs PHYSIQUES : commande de moteurs, lumières, électrovannes... etc.

Fig. 3 : L'interface de spatialisation

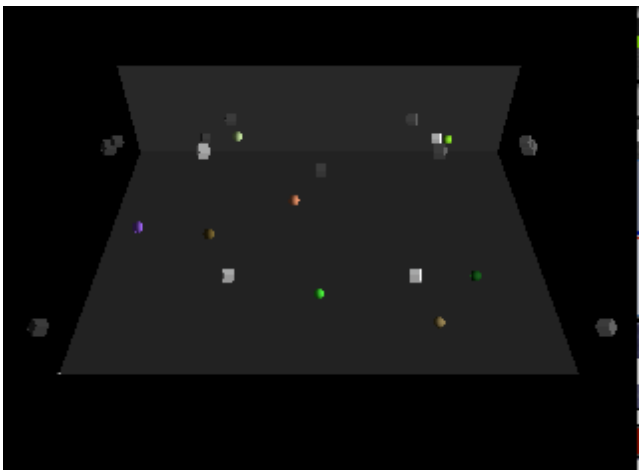
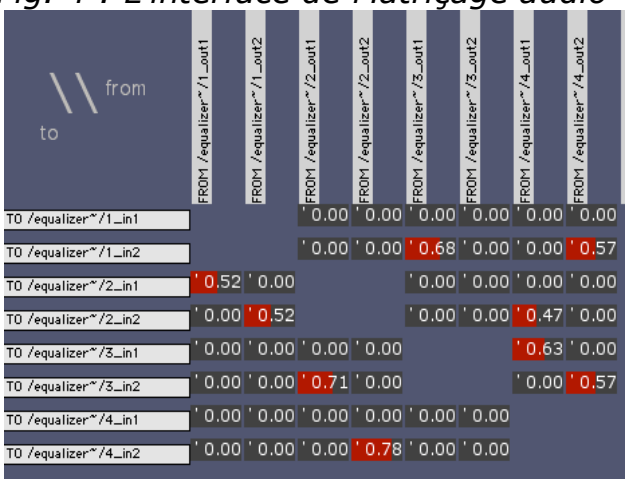


Fig. 4 : L'interface de Matriçage audio



En résumé, EVE permettrait, grâce à l'intégration dans Jamoma, de transformer tout patch Max/MSP/Jitter en un module directement utilisable dans un environnement générique, et ce en un temps réduit (une à quelques heures pour un utilisateur habitué), permettant ainsi de relier ses données à d'autres processus, ainsi qu'à d'autres systèmes (grâce notamment au protocole OSC).

Cela dans le but, d'une part, de permettre à des utilisateurs souhaitant profiter des ressources des technologies temps réel de pouvoir le faire sans avoir à acquérir une expertise de programmation longue et laborieuse.

D'autre part, une personne possédant quelques bases de programmation pourra ainsi créer ses propres modules (ou modifier les modules existants) sans se soucier de toute la couche de programmation concernant l'architecture, qui est de loin la plus difficile à mettre en place.

L'utilisateur pourrait ainsi générer à sa guise autant de configurations différentes, pour chacun de ses projets, à partir des modules présents dans la librairie, ou de ceux qu'il aura créés pour l'occasion. Pour chaque configuration, il pourra générer autant de fichiers de « conduite », ou de « partitions interactives » qu'il le souhaite, et passer de l'une à l'autre en cours d'utilisation.

Bilan de la Maquette

Cette maquette réalisée, ses principes ont pu en être présentés concrètement aux acteurs du métier, afin de recueillir leurs point de vue sur ce développement particulier, et plus généralement sur la pertinence d'une telle démarche. Il faut toutefois signaler que le développement s'étant fait principalement sur un temps libre estival de certains membres du groupe de travail, et donc majoritairement sur une économie générée en marge de l'intermittence du spectacle, il eût été illusoire de compter sur un résultat pleinement opérationnel.

Les évaluations critiques de cette maquette, par des utilisateurs ou artistes-programmeurs l'ayant essayée ou ayant assisté à une séance de présentation, ainsi que celles de l'équipe l'ayant développée, ont été recueillies à diverses occasions :

La principale critique émise à propos de la maquette, à savoir sa fragilité et la difficulté de sa mise en œuvre, interdisant sa mise en œuvre en situation réelle « de plateau » (laquelle constituerait pourtant une étape nécessaire de test), tient avant tout aux conditions dans lesquelles elle a été produite.

Pour les mêmes raisons, l'ergonomie a été perçue comme restant à un stade de développement, très éloigné des principes d'interface (création/conduite) énoncés dans l'étude préalable.

Malgré ces points négatifs, ce prototype a cependant permis d'éprouver certains des principes directeurs mis en œuvre lors de son développement :

En premier lieu, l'utilisation du protocole Jamoma a notamment été discutée et parfois critiquée par l'équipe de développement, notamment pour sa lourdeur d'utilisation. Il faut toutefois signaler qu'au moment du développement de la maquette, Jamoma n'en était qu'à un stade embryonnaire et a été nettement optimisé depuis.

L'utilisation de Jamoma lors du développement de la maquette a également permis de sensibiliser les développeurs à l'importance et à la complexité d'un protocole d'échange et de communication. La question a été posée de savoir s'il ne serait pas plus simple et plus efficace de recréer un protocole moins lourd, *ex-nihilo*. Cette question reste ouverte, dans la mesure où la mise au point d'un nouveau protocole aurait toutes les chances de se voir confrontée aux mêmes besoins de généralisation croissante, et en conséquence d'alourdissement incessant, qui est le verrou principal de toute démarche visant la généricité.

Une nouvelle tentative de développement de EVE demanderait alors de faire un nouvel examen de Jamoma pour interroger la pertinence de son utilisation renouvelée.

Cet examen a partiellement été accompli par certains membres du groupe de travail, plusieurs développeurs ou « artistes-programmeurs » intéressés par cette démarche ainsi que toute l'équipe de développement de Jamoma, lors d'un atelier de développement autour de Jamoma ayant eu lieu au mois de mars 2007 (financé par deux structures participant au groupe de travail : Incidents Mémorables et le GMEA).

Un réel progrès a pu être mesuré en l'espace d'un an, mais le protocole Jamoma ne semblait alors pas encore suffisamment au point pour être employé de façon fiable dans l'immédiat. Cet atelier a cependant permis de procéder à un « alpha-testing » de la nouvelle version (fraîchement finalisée) de Jamoma, et d'en détecter et corriger la majeure partie des « bugs ».

À l'issue de l'atelier, l'ensemble des participants a toutefois salué les principes et le sérieux du développement et considérait globalement ce projet comme très prometteur. Nous continuerons donc de suivre son évolution, au cas où les conditions seraient réunies pour lancer une deuxième phase de développement de EVE.

Concernant la maquette EVE, la question du public potentiel d'un tel dispositif a été également posée :

Les tests d'utilisabilité ont notamment révélé d'une part la nécessité d'un travail conséquent d'ergonomie pour qu'un tel dispositif puisse être adopté par un public significatif, non-spécialiste et pourtant confronté à l'utilisation de processus spécifiques. D'autre part, ils ont posé la question d'une « quatrième catégorie » d'utilisateurs (en prenant pour référence la répartition des niveaux d'utilisateurs exposée ci-dessus), ne nécessitant pas l'apprentissage de la configuration du dispositif (qui reste, en l'état, relativement complexe et peu ergonomique), mais seulement de son utilisation : la tâche de la mise au point de configurations spécifiques pour chaque projet artistique restant alors confiée à un développeur, ou du moins à un utilisateur plus confirmé.

En outre, un certain nombre d'artistes-programmeurs spécialistes de Max/MSP, à qui la maquette a été présentée, ont considéré ce prototype comme trop fermé pour leurs besoins, préférant s'en tenir aux spécificités de leurs propres outils et défendant « le minimalisme de la page blanche comme principal outil de programmation ».

Ce public représentait cependant un champ d'utilisateurs à la limite de l'objet de notre étude, dans la mesure où celle-ci concernait finalement davantage les questions de transmission et d'appropriation - en condition de production - de dispositifs complexes mis au point par des tiers, que l'utilisation de ces dispositifs par leurs créateurs, en soi beaucoup moins problématique.

En définitive, la question de la transmission de la pensée créative et des moyens de la réaliser d'un créateur (en l'occurrence créateur de dispositifs génératifs, sous forme de patches Max/MSP) vers un interprète est ici un premier point central que cette étude a adressé à travers de questionnements techniques principalement. Cette question est, par exemple, réglée dans le cas de l'interprétation musicale « traditionnelle » par le moyen de la partition et de la culture d'interprétation transmise par les conservatoires.

En revanche, il est évident que ce domaine de création, relevant des « mutations numériques » dans le spectacle vivant, est en pleine émergence, et ne peut certainement encore trouver de réponse à cette question. Si réponses il doit y avoir, elles émergeront par la pratique et par les choix politiques, technologiques et culturels qui se dessineront dans les années à venir.

Le développement et l'examen de la maquette a permis également de valider certains choix présidant au développement :

- Tout d'abord la logique modulaire a été validée comme justification principale d'un tel dispositif face aux logiciels « commerciaux » fermés : l'ajout de fonctions spécifiques – indispensable pour les projets incluant une part importante d'expérimentation – restant alors possible, tout en gardant l'avantage d'une infrastructure générique pour la réactivité en situation de création « de plateau ».
- L'aspect collaboratif du développement et l'esprit de mutualisation ont également été appréciés par les membres de l'équipe de développement (même si le constat a été fait que ce travail collectif était mené par un noyau dur moteur des choix et de l'organisation sous-tendant le travail de développement). En effet, un tel processus de développement collaboratif est relativement rare dans une culture de travail plutôt « individualiste » puisque issue des domaines de la composition musicale et des arts numériques, arts majoritairement pratiqués de façon solitaire par leurs créateurs.

Ce dernier point – la complexité d'un développement collectif dans des domaines où les frontières entre artistique et technique restent floues – est certainement un second nœud de la problématique adressée par cette étude, avec la confrontation entre deux cultures de travail : une culture de travail solitaire de « l'artiste numérique », et celle , éminemment collective, du spectacle vivant.

De même, la question de la création expérimentale, hautement spécifique, et de son insertion dans un contexte générique reste entière, même si l'expérience de ce développement et de sa mise à l'épreuve de l'usage a permis d'ouvrir quelques pistes et de poser plus précisément les questions.

En effet, il est évident que la pensée artistique –et en particulier dans le cadre des « arts numériques » - interagit en permanence avec le développement de ses outils et/ou l'intégration d'outils prévus à d'autres fins. Cependant, l'insertion de la création numérique « expérimentale », fonctionnant selon ce paradigme, dans un cadre aussi contraint que celui de la production de spectacle vivant – pour lequel l'urgence est bien souvent la règle, et dont le temps de travail est fortement limité par toute une organisation du travail relativement lourde – révèle avec force le besoin de travailler à une amélioration des conditions.

Conclusion et travail futur

Nous avons ici esquissé des directions de travail et de recherche, inévitablement partiales et discutables, en vue de répondre à ce besoin par le développement de l'aspect générique des outils et dispositifs employés, à travers notamment la mise au point de protocoles et d'architectures communes et la mutualisation des moyens.

En outre, la dynamique contradictoire entre deux aspirations opposées et également indispensables à la création, que l'on pourrait résumer au conflit entre les aspects « artistique et technique », n'a cessé de traverser notre étude durant les discussions, développements collectifs et mises en oeuvre. Il serait toutefois bien présomptueux de prétendre y avoir répondu, tant cette contradiction est un des moteurs de la création artistique, et en particulier lorsqu'elle fait un usage de moyens techniques ou technologiques poussés.

Dans la continuité de cette étude, et en vue de permettre le développement d'environnements de régie numérique (audio, vidéo, lumière, et…), dans le contexte de la création artistique, un consortium a été mis en place, regroupant plusieurs des acteurs ayant participé au groupe de travail, en l'augmentant d'autres laboratoires et entreprises complémentaires, afin de créer une plate-forme de recherche autour des interfaces de contrôle et d'écriture multimodales et multimédia pour la création artistique et les industries culturelles, associant acteurs artistiques, entreprises et laboratoires académiques.

Ce projet, du nom de Virage, a été déposé au mois de mars 2007 auprès de l'Agence Nationale de la Recherche, section Audiovisuel et Multimédia.

Dans l'hypothèse où il aura été retenu, il constituera de fait une suite logique à cette étude et au groupe de travail, rassemblé grâce au soutien financier de l'Association Française d'Informatique Musicale, et à l'apport en industrie des structures y ayant participé.

Remerciements

Le groupe de travail tient à remercier Thierry Coduys qui, en donnant l'impulsion initiale à cette étude au sein de l'AFIM a permis que ce projet puisse se constituer.

Il remercie aussi toutes les personnes ayant participé aux tables rondes, rencontres et réunions, et qui ont bien voulu lui accorder leur temps, principalement de manière bénévole, pour en faire avancer les contenus.