

Music Composer's Workbench : vers un environnement intelligent de l'enseignement de la composition tonale

Pierre Jouvelot¹
en collaboration avec Narcis Bonet²

¹CRI, Ecole des mines de Paris

²Schola Cantorum, Paris

jouvelot@cri.ensmp.fr

Résumé

Music Composer's Workbench (Atelier du Compositeur de Musique) est un projet, en cours de développement, d'un environnement logiciel complet pour l'enseignement à distance de la composition musicale tonale. Il est le fruit d'une collaboration entre le Centre de Recherche en Informatique de l'École des mines de Paris et l'école d'expression artistique et musicale Schola Cantorum.

Contrairement aux approches essentiellement syntaxiques utilisées jusqu'à présent dans les environnements de composition, l'accent est mis sur une prise en compte aussi poussée que possible par le logiciel des caractéristiques sémantiques de la musique tonale aux niveaux de l'harmonie, du contrepoint, de l'instrumentation, de l'orchestration, des formes ou des rythmes musicaux. Un premier prototype écrit en Java, et disponible sur Internet, permet, en se limitant pour l'instant aux bases essentielles de l'harmonie tonale, d'en valider les premiers concepts.

Mots-clés: enseignement assisté par ordinateur, composition, musique tonale, MIDI, client-serveur, Java, Internet, MCW

1 Introduction

Si la musique a été, de tous temps, une application privilégiée de l'informatique [13], ce n'est qu'avec le développement de la norme MIDI de commande des appareils musicaux dans les années 1980 et l'accroissement de puissance des ordinateurs personnels que la diffusion de l'informatique musicale a connu un impact significatif dans le grand-public. Celle-ci se développe selon essentiellement trois grands axes :

- la recherche en composition musicale contemporaine, avec des machines matérielles et/ou logicielles comme la mythique 4X à l'IRCAM, Csound [16] au MIT ou Common Music [9] au CCRMA. L'objectif est ici d'accroître l'expression du langage musical dont peuvent tirer parti les compositeurs contemporains, que cela soit au niveau des nouvelles représentations de l'univers musical ou de nouveaux matériaux sonores.
- l'édition de partitions musicales, avec des logiciels commerciaux comme Finale, Sibelius, Cubase ou Logic. Ici, la puissance de l'informatique est utilisée pour développer des systèmes de traitement de texte spécifiques pour la mise en forme des partitions.
- l'élaboration d'oeuvres musicales pour des applications audiovisuelles comme le multimédia, le cinéma ou la vidéo. Ces pièces musicales peuvent être jouées sur des échantillonneurs MIDI spécialisés comme les samplers proposés entre autres par Kurzweil ou Roland, ou, depuis peu, des systèmes audionumériques entièrement logiciels comme Protools ou Cubase Audio.

L'objectif du projet Music Composer's Workbench (MCW, pour Atelier du Compositeur de Musique) est de participer au développement d'un quatrième axe d'applications de l'informatique à la musique : l'enseignement, éventuellement à distance, de la composition musicale par le développement d'un environnement "intelligent" adapté.

Les cloisons entre les domaines cités ci-dessus sont, bien sûr, rarement complètement étanches ; de nombreux environnements de composition permettent l'élaboration de partitions ; les systèmes de traitement de partitions offrent souvent la possibilité de générer un fichier MIDI pour permettre une première écoute. Nous retrouverons ces interdépendances dans l'environnement MCW.

Dans le reste de cet article, nous motiverons ce nouvel axe de recherche et la nécessité d'un nouvel environnement de composition musicale (section 2). La section 3 décrit la conception et l'implémentation d'un prototype de cet environnement ; celui-ci est disponible sur Internet à l'adresse <http://www.cri.ensmp.fr/~pj>. Nous aborderons dans la section 4 les perspectives offertes par cette nouvelle approche, avant de conclure (section 5).

2 Motivations

Le projet Music Composer's Workbench a pour principal objectif de profiter de la puissance qu'offrent les nouvelles technologies informatiques et de télécommunication pour développer un nouvel axe de l'informatique musicale peu exploité jusqu'à présent : l'enseignement assisté par ordinateur de la composition tonale.

L'environnement MCW se distingue des environnements de composition existants par l'utilisation concomitante de deux aspects novateurs : (1) l'accent mis sur les caractéristiques sémantiques de la composition musicale, et donc de son enseignement, et (2) l'approche distribuée permettant d'instaurer une vraie relation maître-élève via Internet, même en l'absence de contact physique.

Séparément, ces deux idées ne sont pas neuves. L'aspect sémantique est prépondérant dans les systèmes automatiques de composition ou d'harmonisation tonales comme celui de BackTalk [11] basé sur MusES [12]. Nous tenons cependant, ici, à intégrer ces connaissances dans un système mixte, couplant automatisme et intervention d'un professeur, à des fins d'enseignement. Sur l'aspect distribué, des projets comme Elody [7] ou l'interface HTTP développée pour Common Music [10], bien que visant à développer des environnements pour de la musique essentiellement contemporaine, mettent également en exergue l'intérêt de cette distribution sur Internet ; elle est évidemment encore plus manifeste pour un système d'enseignement comme c'est, en partie, le cas ici.

2.1 Sémantique

Les nombreux environnements de composition musicale existants ont essentiellement pour fonction de proposer des moyens de manipulation de représentation d'objets sonores. Ces représentations peuvent être standards, à base de portées, notes et silences dans les logiciels d'édition de partitions musicales évoqués ci-dessus, ou plus abstraites, comme dans les environnements de recherche de la musique contemporaine comme Elody [7] de Grame, ou OpenMusic [1] et JavaMusic [8] de l'IRCAM. Bien que superficiellement différents, ces environnements s'attachent tous à fournir des moyens de création, de manipulation ou d'expression de formes sonores, sans en *comprendre* fondamentalement le sens ; nous dirons qu'ils sont essentiellement *syntactiques*.

Si l'informatique fournit ainsi au compositeur des moyens d'exprimer ses intentions musicales, elle est rarement utilisée pour critiquer les choix faits par l'artiste, que ceux-ci soient conscients ou non. Cette critique n'est possible que sous réserve que le système possède un modèle global permettant de valider telle ou telle des décisions prises par l'artiste. Notre propos est donc ici de développer dans l'environnement MCW une base de connaissance *sémantique* permettant, à des fins d'aide ou d'apprentissage, de capitaliser les règles qui sous-tendent le processus de composition, que ce soit dans le domaine de l'harmonie [4], du contrepoint [5], de l'instrumentation [2], etc. La maîtrise de ces règles est nécessaire à tout compositeur, que ce soit pour les appliquer... ou les enfreindre en connaissance de cause.

Pour illustrer cette différence entre syntaxe et sémantique, il est intéressant de faire un parallèle rapide avec les environnements de programmation informatique. D'un côté, on trouve les systèmes de traitement de texte comme Emacs : ils traitent les programmes comme des suites de caractères ASCII, avec, éventuellement, un minimum de connaissances sur leur syntaxe pour permettre de distinguer les mots-clés des commentaires ou des instructions. Sur le plan sémantique, on trouve le Programmer's Apprentice de Waters [17] qui étend Emacs avec un riche module sémantique et pragmatique permettant d'indiquer au programmeur des possibilités de code incorrect ou peu efficace, avec des conseils du genre "cette fonction récursive n'a pas d'alternative de sortie".

Pour revenir à la musique, notre ambition, modeste, n'est pas d'élaborer un modèle général d'analyse critique ou d'aide générale à la composition musicale, dont on peut facilement douter de l'existence. Le domaine d'application auquel nous nous limitons dans le projet MCW est l'enseignement de la composition musicale tonale. La musique est un art et son apprentissage relève, encore de nos jours, plus de la relation maître-élève ou, même, "maître-disciple" que de l'enseignement traditionnel. L'informatique n'a pas été encore largement utilisée pour faciliter le processus d'apprentissage des règles de base de la musique, qui demande de mettre en oeuvre le type de base de connaissance évoquée dans le paragraphe précédent. Ces bases de connaissances recouvrent de nombreux domaines, comme l'harmonie, le contrepoint ou l'instrumentation. Que l'on observe un cours d'harmonie [4] ou un traité d'instrumentation comme celui d'Hector Berlioz [2], et l'on se rend compte du volume de connaissances amassées au cours des siècles et que devra maîtriser l'apprenti compositeur.

Si l'on observe, plus concrètement, l'enseignement classique de l'harmonie tonale, les règles de base comme "les octaves parallèles sont interdites", "éviter les intervalles augmentés", "par défaut, mettre la sixte au soprano", sont relativement faciles à énoncer ; les mettre en pratique par l'élève relève encore du domaine d'un apprentissage long et progressif, largement basé sur la réalisation des exercices que sont les basses ou chants donnés [3]. L'activité du maître est ici triple :

- introduire et justifier par des raisonnements esthétiques ou physiques les diverses règles de composition qui seront utilisées lors des exercices par l'élève.
- souligner les erreurs, ou écarts à la règle, faites par l'élève : "la sensible ne monte pas vers la tonique", "le ténor et la basse font des quintes parallèles" ou "entrent en fausse relation", "la septième devrait descendre", etc.
- proposer d'autres réalisations conformes aux règles, dont le nombre et les spécificités varient au fur et à mesure de la progression de l'élève.

S'il va de soi que l'apport pédagogique du maître est primordial pour les première et dernière étapes, la seconde, rarement intéressante et peu formatrice, pourrait être avantageusement remplacée par un environnement informatique qui maîtriserait les bases de l'harmonie (l'aspect sémantique). Cela permettrait au professeur de se concentrer sur la vraie valeur ajoutée pédagogique et à l'élève de progresser plus rapidement en lui permettant d'éviter de se fourvoyer dans des voies sans issue. Ce qui vient d'être dit pour l'harmonie se généralise évidemment à l'enseignement du contrepoint ("telle note est étrangère", "deux harmonies dans la même mesure sont interdites", "telle rupture mélodique s'oppose au cantus firmus") ou de l'instrumentation ("telle trille de clarinette est difficile à jouer", "cette note sort de la tessiture du basson"). Le prototype du MCW se limite pour l'instant à n'aborder que les aspects harmoniques.

2.2 Accès à distance

La seconde caractéristique du MCW est sa capacité à opérer en mode distant. Comme on l'a vu, l'enseignement des bases de la composition nécessite un contact quasi-permanent avec un maître ; c'est évidemment difficile pour un élève qui n'habite pas près d'un centre musical adéquat. Les nouvelles technologies, et Internet en particulier, permettent de pallier cette déficience en offrant la possibilité d'une relation virtuelle entre le professeur et ses élèves, éliminant du coup un des freins majeurs, à la fois pratique (temps de transport) et économique (coût des transports), à la diffusion de la connaissance du processus compositionnel.

MCW est basé sur un modèle distribué de l'apprentissage. L'élève a la possibilité de travailler, en mode local sur sa station, les divers exercices qui lui sont fournis par le professeur, de profiter du module de connaissance sémantique évoqué dans la section précédente pour déterminer et, éventuellement, corriger les fautes les plus évidentes, et de soumettre, enfin, une version aussi correcte que possible de sa réalisation. Le professeur peut alors étudier l'exercice réalisé et communiquer, selon des modes divers (courrier électronique, tableau noir distribué, discussion), ses commentaires à l'élève. Bien évidemment, cette interaction peut être étendue à un groupe d'élèves ou aux élèves entre eux, indépendamment de leur localisation géographique, grâce au mécanisme de communication Internet.

Le prototype d'environnement MCW présenté dans la suite de cet article est entièrement implémenté en Java [6], permettant ainsi une disponibilité immédiate du système à toute personne munie d'un browser Netscape ou Internet Explorer. Son implémentation est décrite plus avant dans la sous-section 3.2.

3 Prototype

Le prototype du Music Composer's Workbench est un environnement d'aide et d'enseignement à la composition correspondant aux deux premières années d'une classe typique d'harmonie. Il est à même d'aider l'élève dans la réalisation de basses et chants donnés contenant, dans l'état actuel du projet, des accords à l'état fondamental, ainsi que les deux renversements, mais ne prend pas en compte les septièmes de dominante.

Nous examinerons tout d'abord un exemple de réalisation permettant d'apprécier l'interface utilisateur, avant d'aborder son implémentation.

3.1 Utilisation

L'interface utilisateur de MCW illustrée dans la figure 1 est centrée autour de deux fenêtres principales : la première affiche¹ l'exercice à réaliser, ici une basse donnée de Challan [3], tandis que la seconde permet d'interagir avec l'utilisateur aussi bien en entrée (pour demander des explications sur un terme inconnu) qu'en sortie pour indiquer des conseils ou signaler des erreurs. Les actions sur ces deux fenêtres sont pilotées à partir de deux séries de menus et de boutons situées entre ces fenêtres.

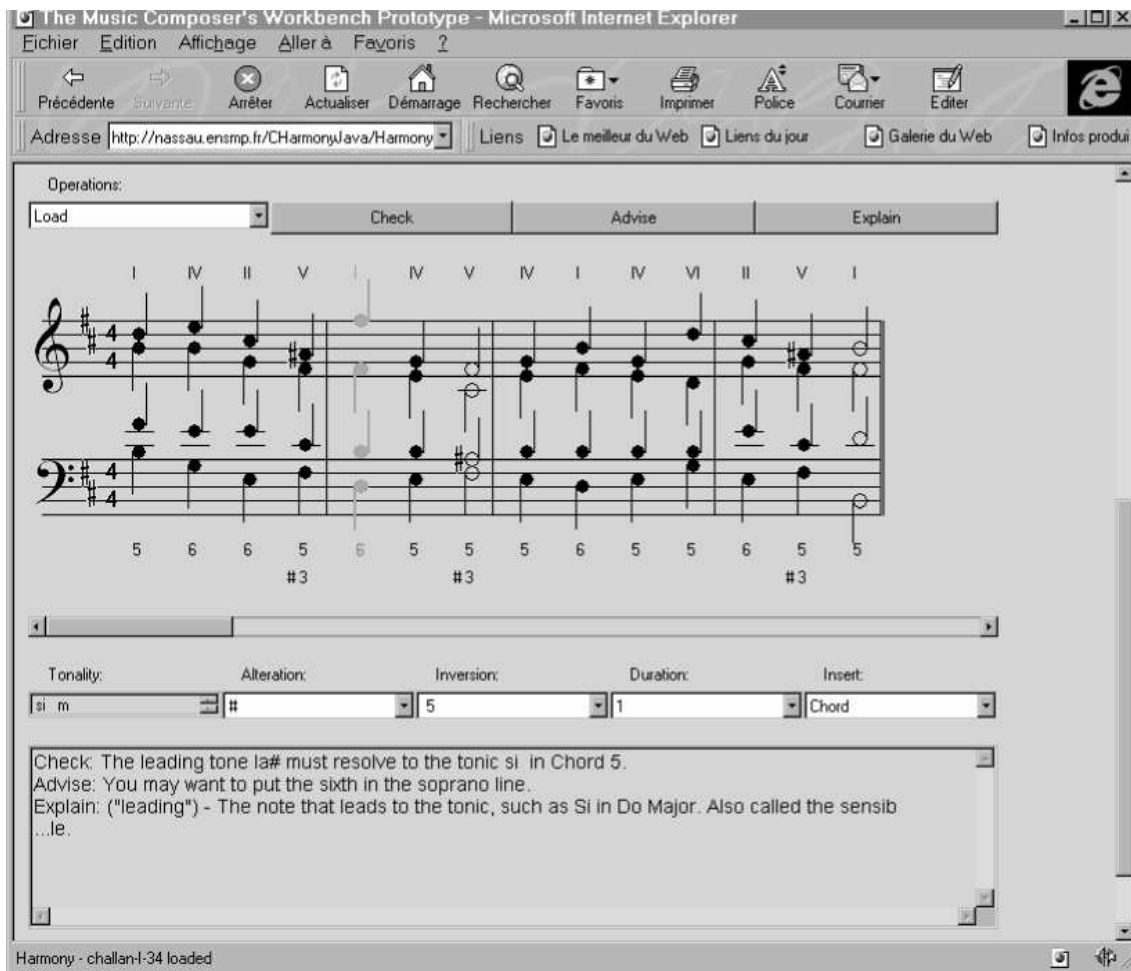


FIG. 1 - Interface utilisateur MCW

Au dessus de la fenêtre de dialogue, on trouve l'ensemble des commandes permettant d'éditer l'exercice en cours. Il est possible de modifier la tonalité par le menu déroulant Tonalité ; celle-ci est également

1. La barre de défilement permet de se déplacer le long de la partition.

reflétée par les altérations indiquées à la clef. Ensuite, la note sélectionnée à la souris peut être altérée² par le menu déroulant Alteration, ou voir sa durée modifiée par le menu Duration. L'accord où elle apparaît peut être modifié en insérant, à la souris, ou en éliminant, avec la touche Delete, des notes selon les directives présentes dans le menu Insert. Enfin, le choix du renversement souhaité est affiché par le menu Inversion; celui-ci est modifiable par l'utilisateur (5, 6 ou 6/4 dans la version actuelle) et les altérations classiques peuvent être ajoutées, après sélection à la souris, sur chaque chiffre via le menu Alteration. Le système calcule automatiquement les degrés, qui sont affichés au-dessus de la partition.

Deux restrictions importantes : (1) les silences ne sont pas autorisés dans cette version, ceux-ci n'ayant pas, au niveau où se place ce prototype, de rôle harmonique à part entière et (2) s'il est possible de moduler dans toutes les tonalités par ajout d'altérations aux chiffres, le module de vérification sémantique du prototype n'utilise, pour l'instant, dans ses analyses que la tonalité spécifiée pour la partition entière.

Au dessus des deux fenêtres, un premier menu Operations permet d'effectuer des opérations classiques de gestion de la base des exercices comme le chargement ou la sauvegarde d'un exercice, ou des opérations globales comme le changement de tempo ou de mesure. Par ailleurs, l'option Play permet, à partir d'une réalisation éventuellement incomplète d'un exercice, de générer un fichier MIDI qui pourra être écouté sur la station de l'élève, offrant ainsi la possibilité de comparer différentes réalisations d'exercices et de justifier certains partis pris dans l'élaboration des règles d'harmonie.

Les trois autres boutons restants ouvrent l'accès à la base de connaissance qui apporte le "plus" sémantique qui manque aux autres environnements de composition, et qui en fait tout l'attrait dans une optique d'enseignement de la composition musicale :

- Le bouton Check permet de vérifier si une réalisation est conforme aux règles d'harmonie intégrées dans le moteur. Les messages sont imprimés dans la fenêtre de dialogue avec l'utilisateur.
- Le bouton Advise permet d'orienter un élève en suggérant certaines modifications à une réalisation imparfaite. A titre d'exemple, on peut obtenir des conseils comme "sur un premier renversement du degré IV, doubler de préférence la tonique plutôt que la sixte".
- Le bouton Explain fournit à l'utilisateur une définition explicative sur tout mot qui aura été surligné dans la fenêtre de dialogue.

Les messages présents dans la fenêtre de dialogue de la figure 1 ont été obtenus en cliquant successivement sur Check, Advise et Explain, après avoir surligné le mot "leading tone" (sensible, en anglais).

Dans cette version, les règles de la base de connaissance utilisées pour Check, Advise et Explain sont fixes pour l'ensemble des utilisateurs. La section 4 suggère d'autres alternatives ; en particulier, on pourrait préférer laisser le professeur paramétrer celles-ci en fonction de la progression de l'élève.

3.2 Implémentation

Nous abordons ici les aspects essentiels de l'implémentation du prototype MCW. Notons que certaines des limitations qui pourront apparaître lors de cette description seront évoquées dans la section 4 où certaines extensions seront proposées.

Puisque l'un des objectifs principaux du projet MCW est d'accroître la diffusion de la connaissance musicale à travers le réseau Internet, l'utilisation de Java s'imposait comme moyen quasi-universel de distribution du système. Le seul outil qui doit être installé sur le système client de l'élève est un simple browser supportant Java, ainsi que, éventuellement, une carte son si l'on souhaite être à même d'effectuer des écoutes des réalisations. Les mises-à-jour éventuelles du MCW sont automatiquement intégrées lors du chargement de l'applet³ ce qui limite énormément le coût de gestion administrative du projet. Le browser permet aussi de gérer, de manière découplée pour l'instant, les communications entre l'élève et le professeur par simple courrier électronique.

Une applet Java a été développée pour faciliter l'apprentissage pratique de l'harmonie : elle est disponible à l'adresse <http://www.cri.ensmp.fr/~pj>. L'orientation objet de Java permet de facilement définir des objets "métiers" appartenant à des classes comme Note, Tonalite ou Chiffrage. La bibliothèque AWT standard facilite la gestion de tout ce qui touche à l'interface homme-machine : souris,

2. Dans la version actuelle, les indications d'altérations présentes dans une mesure ne sont pas automatiquement transférées aux notes suivantes de la mesure.

3. Sa taille actuelle est d'environ 2500 lignes de Java, soit environ 80 koctets de bytecode.

menus déroulants, boutons, etc. Toutes les interactions sont traitées localement au système client ; elles modifient une représentation mémoire de la partition créée à l'aide des classes ci-dessus et qui reste locale jusqu'à sauvegarde par l'intermédiaire du menu Operations. Cette gestion très distribuée permet de garantir de bons temps de réponse interactifs.

Dans la version actuelle du système, les règles d'harmonie sont codées par des procédures chargées lors du transfert de l'applet. Elles peuvent être appliquées à la réalisation d'exercices de basses et chants donnés jusqu'à l'introduction de la septième de dominante. A titre illustratif de la simplicité de codage qu'offre l'orientation objet de Java, voici un exemple de méthode de la classe `Accord` permettant de vérifier qu'une sensible n'est pas doublée dans une tonalité donnée `t` :

```
public void pasDoubleSensible( Tonalite t )
    throws CheckException {
    Note sensible = t.noteSensible() ;
    int occur = 0 ;

    for( int i = 0 ; i < notes_par_accord ; i++ ) {
        occur += ( notes[i].octaveAvec( sensible ) ) ? 1 : 0 ;
    }
    if( occur > 1 ) {
        throw new CheckException( "The leading tone "+
            sensible.nom()+
            " shouldn't be doubled" ) ;
    }
}
```

Toute instance objet de la classe `Accord` possède une variable tableau `notes [notes_par_accord]`, chaque note de la classe `Note` codant pour une des quatre voix d'une réalisation SATB (soprano, alto, ténor, basse). Une exception de la classe `CheckException` définie dans l'applet est levée dans le cas où le nombre d'occurrences de la sensible ("leading tone") dans l'accord représenté par `notes` est supérieur à un ; le message associé sera affiché par la suite dans la fenêtre de dialogue.

Du fait des limitations inhérentes au mode de fonctionnement de Java et, en particulier, de l'impossibilité de stocker sur le site client des informations pérennes, l'applet `MCW` lancée après chargement automatique lors de l'accès du lien `MCW` doit communiquer avec un serveur situé sur le site du Centre de Recherche des mines pour accéder aux divers exercices et permettre de sauvegarder les réalisations propres à chaque élève. Ce serveur est également implémenté en Java. L'identification des élèves est faite lors de l'initiation de l'applet par l'intermédiaire d'une fenêtre de login lancée automatiquement ; cette identification permet de gérer pour chaque utilisateur un espace fichier propre. Cette centralisation des données offre également la possibilité au professeur d'accéder aux résultats de ses élèves sans avoir d'accès privilégié aux sites clients d'où les élèves se connectent.

Une autre fonction du serveur est de stocker les fichiers MIDI dont la création est déclenchée par l'option `Play` du menu `Operations`. L'applet Java, dans la version 1.0.3 compatible avec les deux browsers Netscape et Internet Explorer, n'a pas, pour des raisons de sécurité, la permission d'accéder directement au matériel des stations clients⁴. Cette sécurité pourrait être désactivée, mais cela n'est pas une solution satisfaisante ; la technique utilisée par `MCW` consiste à générer sur le serveur un fichier d'extension `.mid` qui sera rechargé tout de suite après création à l'aide d'une instruction Java `showDocument`. Le fichier ainsi chargé lancera le plug-in du browser traitant les fichiers MIDI présents sur le Web. Cette solution, si elle n'est que partiellement satisfaisante, devrait être, à terme, remplacée par l'utilisation de l'Application Programming Interface (API) Java consacrée au support des cartes MIDI sur les stations clientes.

Le dernier rôle du serveur actuel est de gérer la base des définitions des termes, qui est accessible par l'intermédiaire du bouton `Explain`. Il n'est pas envisageable d'importer la base complète dans l'applet, qui doit rester aussi petite que possible pour limiter au maximum les temps de chargement.

4. Une autre restriction de la version Java 1.0.3 est l'impossibilité d'imprimer une copie d'écran. Ce n'est qu'un problème technique résolu dès la version 1.1 qui n'est, malheureusement, pas encore bien prise en compte par les différents browsers. `MCW` devrait, à l'avenir, inclure un service d'impression de partitions, selon un schéma similaire à celui utilisé pour la génération de fichiers MIDI.

4 Perspectives

Le prototype du Music Composer's Workbench n'a pour seule ambition que de suggérer le type d'environnement qui, à terme, devrait voir le jour pour faciliter à la fois le travail du compositeur et l'apprentissage des bases de la composition musicale. Cet environnement ne couvre, pour l'instant, que l'harmonie dans ses aspects les plus rudimentaires, mais, augmenté de l'ensemble des connaissances que l'on trouve regroupées dans les différents traités d'harmonie, de contrepoint, d'instrumentation ou, même, de composition [14], il ne fait pas de doute que son domaine d'application est vaste, ne serait-ce que dans le domaine de l'enseignement.

L'aspect réseau du MCW, tel qu'il est présenté, n'est qu'embryonnaire. Seule une interaction minimale par courrier électronique est supportée, via l'interface de courrier électronique des browsers. A terme, il conviendrait d'intégrer dans cet environnement des forums de discussion (type newsgroups, déjà disponible avec les browsers actuels) ou des tableaux noirs interactifs permettant aux élèves de discuter entre eux, à distance, et avec leurs professeurs. L'observation directe étant extrêmement formatrice, observer les réalisations faites par d'autres apprentis, ainsi que les erreurs qu'ils ont pu commettre, ne peut qu'accroître l'intérêt pédagogique de cet environnement, une fois généralisé à un nombre plus conséquent d'utilisateurs. Le prototype du MCW, malgré ses limitations, a déjà été utilisé à travers le monde depuis sa création il y a six mois, comme le montrent les commentaires envoyés aux auteurs par courrier électronique.

La gestion des règles d'harmonie et, plus généralement, des bases de connaissance devrait être envisagée de manière plus déclarative. Une première approche serait de pouvoir paramétrer l'utilisation de telle ou telle règle. Par exemple, le professeur pourrait commencer par limiter les chiffres au simple état fondamental pour un cours introductif, ajoutant par la suite les notions plus complexes. On pourrait aussi, comme le propose [12], envisager de définir un métalangage de déclaration des règles d'harmonie ou autres, de manière à ce qu'une modification en temps quasi-réel de l'environnement soit possible ; l'environnement MCW serait alors structuré autour d'un interprète de ces règles plutôt que de les voir codées de manière procédurale comme c'est le cas actuellement.

Dans le système actuel, les routines d'analyse (Check) et de conseil (Advise) travaillent sur un contexte extrêmement limité : l'accord sélectionné ou une paire d'accords successifs. Il est évident qu'une prise en compte d'un contexte plus large, dans un mode procédural aussi bien que déclaratif, permettrait d'affiner la qualité de l'analyse et la pertinence des conseils fournis à l'utilisateur.

Si l'ensemble des règles, sous forme procédurale ou déclarative, se trouvait être de taille trop importante pour être chargé à chaque lancement de l'applet, on pourrait envisager de transférer la représentation de la partition sur le serveur pour opérer un Check. Cela aurait l'avantage, dans le cas d'un système déclaratif de règles, de pouvoir coupler le système avec un solveur de contraintes (i.e., de règles) interfacé avec Java, du type BinProlog [15] ou autre, réglant du même coup le problème évoqué dans le paragraphe précédent. Ce développement va, bien sûr, au delà de l'objectif du prototype évoqué ici qui n'avait qu'un rôle de validation des concepts.

Il pourrait, par ailleurs, être intéressant, par exemple pour pallier l'absence du professeur, de coupler un tel système d'enseignement avec des systèmes experts, comme un harmoniseur automatique tel qu'il en existe pour les chorals de Bach [11]. Leur apport pédagogique est certes faible, mais ils peuvent servir de pis aller. Il serait également utile de pouvoir proposer automatiquement des solutions proches de celles fournies par l'utilisateur, de pouvoir terminer un exercice laissé pendant ou, encore plus difficile, de suggérer des endroits où d'autres solutions que celle donnée par l'utilisateur auraient permis de terminer l'exercice.

Les professeurs de composition ont un rôle majeur à jouer comme cela a été évoqué dans la section 2. En particulier, il convient d'accroître la base de connaissances de toute la physique nécessaire pour justifier certaines des règles. Par exemple, insister, dans le cas de l'harmonie, sur l'importance des résonances pour justifier d'éviter les quintes ou octaves parallèles . . . ou les imposer, mais cette fois pour des raisons instrumentales (doublage des violoncelles et contrebasses à l'octave). Cet ensemble d'informations, de type multimédia - on peut penser à des extraits sonores ou audiovisuels - profiterait de l'adjonction d'un CD-ROM, dans le cadre d'un système hybride réseau/CD-ROM. Il faut insister sur le fait que l'objectif du MCW n'est pas d'entamer l'influence des professeurs dans leur enseignement, mais, au contraire, de revaloriser celui-ci en comptant sur un système "intelligent" et évolué pour éliminer tout ce qui est mécanisable dans le processus d'apprentissage et laisser aux enseignants tout le loisir d'apporter aux élèves leur vraie valeur ajoutée.

5 Conclusion

Le projet Music Composer's Workbench a pour ambition d'intégrer au sein d'un même système éducatif distribué l'ensemble des connaissances harmoniques, contrapunctiques, instrumentales. . . que requiert l'exercice de la composition musicale, avec ses règles et ses exceptions. Connaître les règles est un point de passage obligé, même pour tout compositeur contemporain, ne serait-ce que pour pouvoir les transgresser sciemment !

Par l'utilisation des technologies Internet, cet environnement améliorera l'enseignement de la composition, en permettant d'accroître la population à même de profiter de cet enseignement, d'augmenter l'interactivité entre élèves et professeurs et d'accélérer le processus d'apprentissage grâce à un outil intelligent prenant en charge certains des aspects les plus simples et mécaniques de l'apprentissage.

Un premier prototype de validation de ce projet, axé sur l'apprentissage de l'harmonie, a été réalisé en Java et est disponible depuis plusieurs mois. Les premières expérimentations ont montré l'intérêt de ce système, bien que qu'il ne fasse pas de doute que beaucoup reste encore à faire, aussi bien dans le domaine de l'élaboration des connaissances, de la représentation de celles-ci ou de l'environnement interactif entre élèves et professeurs.

Remerciements

L'auteur tient à remercier Fabien Coelho et les membres du comité de programme pour leurs commentaires, et Laurent Daverio pour son aide.

Références

- [1] Assayag, G., Agon, C., Fineberg, F. et Hanappe, P. An Object Oriented Visual Environment for Musical Composition. ICMC'97, Grèce.
- [2] Berlioz, H. *Traité d'instrumentation et d'orchestration*. Editions Lemoine.
- [3] Challan, H. *380 basses et chants donnés*. A. Leduc, 1960.
- [4] Desportes, Y. *Traité d'harmonie*. G. Billaudot, 1977.
- [5] Dupré. *Cours de contrepoint*. A. Leduc, 1938.
- [6] Gosling, J., Joy, B., et Steele, G. *The Java Language Specification*. Addison Wesley, 1996.
- [7] Fober, D., Letz, S. et Orlarey, Y. L'environnement de composition musicale Elody. JIM'97, Lyon, 1997.
- [8] Hanappe, P. et Assayag, G. A framework to integrate environments for computer aided composition and sound synthesis. JIM'97, Lyon, 1997.
- [9] Kunze, T. et Taube, H. SEE—A Structured Event Editor: Visualizing Compositional Data in Common Music. ICMC'96, Hong Kong.
- [10] Kunze, T. et Taube, H. An HTTP Interface to Common Music. ICMC'97, Grèce.
- [11] Pachet, F. et Roy, P. Mixing Constraints and Objects: a Case Study in Automatic Harmonization. TOOLS Europe'95, Prentice-Hall.
- [12] Pachet, P. The MusES system: an environment for experimenting with knowledge representation techniques in tonal harmony. First Brazilian Symposium on Computer Music, SBC&M '94, Aout 3-4, Caxambu, Minas Gerais, Brésil.
- [13] Roads, C. *Computer Music Tutorial*. MIT Press, 1995.
- [14] Schoenberg, A. *Fundamentals of musical composition*. Faber & Faber, 1967.
- [15] Tarau, P. BinProlog User Guide. Disponible à l'adresse Web <http://clement.info.-umoncton.ca/BinProlog>.
- [16] Vercoe, B. *Csound Manual*. MIT Media Lab, 1986.
- [17] Waters, R. The Programmer's Apprentice: A session with KBEmacs. IEEE Transactions on Software Engineering, Novembre 1985.