DOSSIER

DE

L'AFIA

OCTOBRE 1995

N°23

Association Française pour l'Intelligence Artificielle

DOSSIER I.A. ET MUSIQUE

DOSSIER I.A. ET MUSIQUE

réalisé par François PACHET (*) et Gérard ASSAYAG (**)
*** Laforia - IBP e-mail : fdp@laforia.ibp.fr**
**** IRCAM**

Equipes présentées

L'Institut Blaise Pascal
GAIV (Paris 8)
Laforia (Paris 6)
Ircam (Paris)
LABRI (Bordeaux-I)
Son & Vision (Paris 4)
Studio Grame (Lyon)
GREYC (Université de Caen)
IRISA (Rennes)

Introduction : IA et musique ?

Ce dossier présente un panorama non exhaustif mais qui se veut représentatif de la recherche en Musique et Intelligence Artificielle en France. Comme on pourra le voir à la lecture de ces présentations, ces recherches sont le fruit de travaux essentiellement individuels, la musique occupant une place souvent accessoire dans les centres de recherche en informatique, sauf exceptions notables (Ircam, Studio Grame). Par ailleurs, si les frontières de l'Intelligence Artificielle sont floues, celles de la musique le sont tout autant. On ne s'étonnera donc pas de l'extrême diversité des travaux combinant l'une avec l'autre. En réalité, il est bien difficile de décider quand l'informatique devient IA, lorsqu'elle s'applique à un domaine comme la musique. Plutôt que de se perdre en débats sur la nature de l'IA, nous adoptons ici le point de vue simple selon lequel le simple fait de modéliser des tâches relevant des domaines de compétence des musiciens ou des musicologues, fait rentrer ce type d'activité dans le domaine de l'intelligence artificielle. Le lecteur appréciera de lui-même cette équation en lisant les comptes-rendus d'activités présentés ici.

Le dossier est structuré comme suit :
Un encart retrace les travaux en musique et informatique à l'Institut Blaise Pascal, qui sert d'introduction historique aux trois présentations du GAIV (Paris 8), du LAFORIA (Paris 6) et de l'IRCAM . Suivent deux présentations d'équipes universitaires: le LABRI (Université de Bordeaux) : Modèles formels du temps en composition, et le laboratoire SON & VISION (Paris 4), spécialisé entre autres dans la conception d'automates musicaux, dans la lignée de Pierre Barbaud. Suit la présentation du studio GRAME (Lyon), une des rares institutions françaises dédiées à la recherche en informatique musicale, et enfin des présentations de recherche isolées, au sein de départements universitaires d'informatique : le GREYC (Université de Caen) avec des travaux en Ethno-musicologie et IA, et l'IRISA (Rennes) avec deux travaux isolés sur la reconnaissance de partitions et le «doigtage» intelligent pour la guitare. Par ailleurs, un encart recense les références et sources d'information de base en IA et musique.

Musique et Informatique à l'Institut Blaise Pascal

La musique a toujours été un terrain d'expérimentation privilégié de la recherche en Informatique. A l'Institut Blaise Pascal (fédération des laboratoires d'informatique de l'Université Pierre & Marie Curie) en particulier, la musique a traditionnellement joué un grand rôle. D'abord avec les travaux initiés par Patrick Greussay (Cf. la présentation du laboratoire GAIV), Xavier Rodet et Pierre Cointe, qui ont donné lieu, entre autres, aux systèmes *Chant* [Rodet et al. 1984], puis *Formes* [Rodet and Cointe 1991], tous deux développés à l'Ircam. Une des conséquences de ces travaux fut le développement du langage *ObjVlisp*, à la base de l'école française de programmation par objets [Cointe 1984]. Ce mariage de la recherche en informatique et de la musique a continué au LITP, par des recherches aussi bien théoriques qu'appliquées : voir, en particulier, la thèse de Philippe Gautron sur la simulation d'improvisations de Jazz, et ses rapports avec la programmation de processus concurrents [Gautron 1986]. Suivant une approche purement formelle, les travaux de Marc Chemillier [Chemillier 1990b], [Chemillier and Timis 1988], qui traitent des rapports entre automates finis et structures musicales, avec comme résultat particulièrement frappant la «rationalité de la musique sérielle» [Chemillier 1990a]. Plus récemment, Francis Rousseaux [Rousseaux 1990], [Rousseaux 1992] a été un des premiers à utiliser les techniques de l'apprentissage symbolique automatique pour l'enseignement intelligent du solfège (avec le système *LeMusicologue*).

Cette tradition particulière, née du croisement entre informatique plus ou moins théorique, et recherche musicale, se manifeste aujourd'hui plus que jamais, dans plusieurs institutions: le GAIV (fondé par Patrick Greussay entre autres), le Laforia (un des laboratoires de l'IBP), l'Ircam (l'équipe représentations musicales), et, de manière plus isolée, par les travaux de Marc Chemillier au GREYC.

Publications IBP
(à l'exclusion de celles émanant du Laforia, que l'on trouvera dans la présentation des activités de ce labo)

[Chemillier 1990a] **Chemillier M**. *Langages musicaux et automates: la rationalité du langage sériel*. Colloque International «Musique et Assistance Informatique», Marseille, p. 211-227,1990a.
[Chemillier 1990b] **Chemillier M.** Structure et méthode algébriques en informatique musicale. Paris 6, Ph.D., 1990b.
[Chemillier and Timis 1988] **Chemillier M and Timis D**. *Towards a formal theory of musical languages*. 14th International Computer Music Conference, p. 175-183,1988.
[Cointe 1984] **Cointe P.** Implémentation et interprétation des langages orientés objets : application aux langages Smalltalk, ObjVlisp et Formes. Paris 6, Thèse d'état, 1984.
[Gautron 1986] **Gautron P.** Une approche logicielle de la simulation de l'improvisation dans le Jazz. Paris 6, Thèse de 3ème cycle 1986.
[Rodet and Cointe 1991] **Rodet X and Cointe P**. Formes: Composition and Scheduling of Process. The Well-tempered Object: Musical Applications of Object-oriented Programming. S. Pope, MIT Press, 1991.
[Rodet et al. 1984] **Rodet X, Potard Y, Barrière J-B**. The CHANT project: From the Synthesis of the Singing Voice to the Synthesis in General. Computer Music Journal 8 (3): 15-31., 1984.
[Rousseaux 1990] **Rousseaux F.** Une contribution de l'intelligence artificielle et de l'apprentissage symbolique automatique à l'élaboration d'un modèle d'enseignement de l'écoute musicale. Paris 6, Ph.D., 1990.
[Rousseaux 1992] **Rousseaux F**. *Le Musicologue, a Learning Apprentice System for Music Education*. IJCAI-89 Workshop on AI and Music, Detroit, Michigan, p. 105-116,1992.

GAIV - Université PARIS 8

2, rue de la Liberté- 93526 Saint-Denis
GAIV - Groupe Art & Informatique de Vincennes à Saint-Denis

Le **GAIV** (Groupe Art et Informatique de Vincennes à Saint Denis) à été fondé en 1969 par Patrick GREUSSAY et Hervé HUITRIC en tant qu'association libre d'artistes informaticiens. Actuellement composé de Marc BATTIER, Guillaume BILLAUX, Allaoua BOUABDALLAH, Giuseppe ENGLERT, Max HEDIGUER, Jean HOLLEVILLE, Vincent LESBROS et Lirio MARTINEZ, l'atelier musique du GAIV est un lieu de recherches musicales intégré au sein du **Laboratoire d'Intelligence Artificielle** de l'**Université de Paris 8**. L'acquis du GAIV (dans le domaine de la composition automatique, du temps-réel, des systèmes transportables...) ne se laisse pas formuler en quelques phrases, car il consiste en la somme de travaux réalisés (le plus souvent à titre individuel) depuis maintenant 25 ans. Nous ne revendiquons aucune orientation méthodologique ni esthétique, le GAIV reste ouvert à tout apport de recherche réalisable avec le matériel dont nous disposons. L'ambition du GAIV depuis 1974 à été surtout de présenter ses œuvres en concert live, sur du matériel transportable (ce qui explique que la documentation écrite à été quelque peu négligée). Due à la configuration du matériel dont le groupe disposait, l'orientation de la recherche visait en premier lieu les problèmes de composition automatique ou d'interaction homme-machine (improvisations). Aujourd'hui, avec de nouvelles techniques, le groupe se trouve face à une problématique nouvelle : des recherches qui étaient impossibles hier le sont maintenant devenues. A titre d'exemple, suit une présentation d'un travail récent de cette équipe : la sonorisation d'événements.

Sonorisation d'événements

Allaoua BOUABDALLAH, GAIV
alawa@ai.univ-paris8.fr

Introduction

Le logiciel que nous avons conçu, SonoScope, a pour but d'associer une réalité perceptible à des phénomènes ou processus qui en sont dépourvus. C'est le cas, par exemple, de l'ensemble des opérations logiques qui ont lieu au sein de nos systèmes informatiques. Nous avons de ces opérations logiques une connaissance principalement théorique, théorie dont nous vérifions l'exactitude le plus souvent à l'aide de visualisations graphiques ou typographiques. Nous proposons quant à nous une *sonorisation*, c'est à dire une représentation auditive non verbale, en lieu et place ou en complément de toute autre forme de représentation (BOUABDALLAH 1994).

La vue, l'ouïe, l'odorat sont autant de systèmes de perception qui, branchés sur notre environnement immédiat, concourent, chacun dans leur spécialité, à nous informer sur notre milieu ambiant et notre position dans celui-ci. Ils sont tous uniques, capitaux, et il n'est pas question pour nous ici d'établir une quelconque hiérarchie entre nos sens. Au contraire, nous recherchons une utilisation optimale de l'ensemble de nos moyens de perception afin d'appréhender au mieux les phénomènes complexes.

Principe mis en œuvre

Le principe consiste dans un premier temps à associer un timbre à un événement pour lui donner une signature, une identité acoustique. Dans un second temps, nous allons répercuter les variations de l'objet observé sur les différentes dimensions de cette signature.

DOSSIER I.A. ET MUSIQUE

Livres de référence

Il existe aujourd'hui plusieurs ouvrages indispensables sur l'informatique musicale, la plupart en anglais:

[Balaban et al. 1992] est le seul ouvrage disponible traitant spécifiquement des rapports entre musique et IA. Plus proche des sciences cognitives, citons les deux ouvrages chez Academic Press: [Marsden and Pople 1992], [Howell et al. 1991], qui traitent principalement de problèmes de représentation de connaissances musicales. Plus généralement, les ouvrages [Schwanauer and Levitt 1993] [Roads 1989] et [Roads and Strawn 1988] contiennent des papiers de références dans le domaine de l'informatique musicale, dont quelques papiers relevant directement de problèmes d'IA. Signalons par ailleurs [Cope 1991] qui traite de la représentation informatique du style musical, et [Pope 1991] qui traite des rapports entre programmation par objets et musique.
En français, il n'existe à notre connaissance aucun ouvrage spécifique à l'IA/musique. Citons toutefois l'ouvrage volumineux sur la musique et les sciences cognitives rassemblant les présentations d'une conférence de 1988 [McAdams and Deliège 1988].

Bibliographie :

[Balaban et al. 1992] **Balaban M, Ebcioglu K, Laske O**. Understanding Music with AI, AAAI Press/MIT Press 1992.
[Cope 1991] **Cope D**. Computers and Musical Style, Oxford University Press 1991.
[Howell et al. 1991] **Howell P, West R, Cross I**. Representing Musical Structures. London, Academic Press 1991.
[Marsden and Pople 1992] **Marsden A and Pople A**. Computer Representations and Models in Music, Academic Press 1992.
[McAdams and Deliège 1988] **McAdams S and Deliège I**. La musique et les sciences cognitives. Liège, Mardaga 1988.
[Pope 1991] **Pope S**. The Well-Tempered Object, MIT Press 1991.
[Roads 1989] **Roads C**. The Music Machine, MIT Press 1989.
[Roads and Strawn 1988] **Roads C and Strawn J**. Foundations of Computer Music, MIT Press 1988.
[Schwanauer and Levitt 1993] **Schwanauer S M and Levitt D A**. Machine Models of Music, MIT Press 1993.

Les dimensions d'un son sont nombreuses : la fréquence, l'amplitude, le tempo, la spacialisation etc. ... Diverses études (BLY 1982) montrent que l'on peut aisément en distinguer sept, d'autres encore (YEUNG 1980) avancent le nombre de vingt. En tout état de cause cette richesse nous fournit un système de codage de l'information potentiellement très étendu.
Nous pouvons par exemple signifier l'utilisation plus ou moins intensive du processeur d'une machine en associant à cette activité un timbre de cliquetis métallique et une stratégie de sonorisation basée sur le tempo. Une telle sonorisation produit un effet sonore comparable à celui produit par les changements de régime d'un moteur. Ainsi nous pouvons avoir connaissance de l'état de plusieurs indicateurs simultanément et en temps réel, de plus nous n'avons pas les contraintes d'une observation visuelle qui nous impose de diriger et maintenir notre regard.

Architecture Logicielle/Matérielle

L'architecture logicielle de SonoScope, basée sur le modèle Client/Serveur, est constituée de trois modules distincts afin d'assurer les trois fonctions suivantes :

- Collecte de l'information(Serveur).
- Paramétrage et traitement de l'information (Client).
- Gestion de l'interface audio (serveur)

Les communications entre modules sont basées sur les RPC (remote procedure call). Nous pouvons donc à partir d'une machine **(A)**, sonoriser des données en provenance d'une machine **(B)** tout en confiant éventuellement l'émission effective des notes à une machine **(C)** équipée d'une interface MIDI. Concernant la partie matérielle nous noterons que nous utilisons une interface MIDI et que nous avons délaissé momentanément les interfaces de type DAC/ADC (Digital to Analog Converter) en raison de la puissance de calcul nécessaire pour le traitement des échantillons audio.

Conclusion

Nous avons conçu ce logiciel comme un outil d'*observation auditive* générique, puis nous l'avons adapté à l'observation spécifique du fonctionnement intime du système d'exploitation informatique UNIX.
Les résultats obtenus jusque là ne nous permettent pas de conclure de façon tranchée. Nous n'avons pas en effet mené d'expérimentation afin de mesurer l'utilité réelle d'un tel système. Enfin, SonoScope fait toujours l'objet de développements, qui tendent notamment à l'enrichir de stratégies de sonorisations et à améliorer la gestion simultanée de multiples interfaces audio.

Bibliographie GAIV

Sara BLY. Presenting information in sound. *Proceeding of Human Factors in Computer Systems,* 8(1):371-375, 1982.
Allaoua BOUABDALLAH. Sonorisation d'événements appliquée à un système informatique. Maitrise d'informatique, Université Paris 8 - Vincennes à Saint-Denis, Septembre1994. filière M.I.M.E.
E.S YEUNG. Pattern recognition by audio representation of multivariate analytical data. *Analytical Chemistery,* 52(7):1120-1123, 1980.
Gregory KRAMER, editor. *Auditory Display : Sonification, Audification and Auditory Interfaces,* volumeXVIII of *A proceeding volume in the Santa Fe Institute studies in the science of complexity.* Addison-Wesley, 1994.

DOSSIER I.A. ET MUSIQUE

LAFORIA - IBP

MUSIQUE
ET INTELLIGENCE ARTIFICIELLE

Groupe d'intérêt inter-équipes OMC
(Objets, modèles et connaissances, J.-F. Perrot)
et ACASA (Acquisition de connaissances et apprentissage symbolique, J.-G. Ganascia)
Chercheurs: François Pachet (Maître de conférences),
Geber Ramalho (thésard), Pierre-Yves Rolland (thésard),
Pierre Roy (thésard), Jean Carrive (DEA).

Introduction

Depuis quelques années plusieurs travaux sont menés au LAFORIA visant à concevoir des systèmes informatiques modélisant certaines tâches musicales. La musique représentant pour nous un cadre idéal pour expérimenter et valider différentes techniques de représentation de connaissances. Les travaux effectués s'articulent autour de deux pôles principaux : la représentation de connaissances en musique tonale, et la simulation d'improvisation de Jazz. Nous décrivons brièvement les résultats principaux et donnons un aperçu des travaux futurs de l'équipe.

Représentation de connaissances en musique tonale

La musique tonale est une mine d'or pour les chercheurs en représentation de connaissances. Nous nous intéressons depuis plusieurs années à la résolution de différents problèmes de musique tonale, ces travaux ayant donné lieu à la réalisation d'un système «de base», appelé MusES, qui contient une représentation des principaux concepts de la musique tonale: notes, pitch-classes, intervalles, accords, mélodies [Pachet 1994a; Pachet 1994b] L'ambition de MusES et de circonscrire l'ensemble des «micro» problèmes liés aux notions de base de l'harmonie comme : calculer un intervalle en tenant compte de l'enharmonie (la quinte diminuée de do est différente de sa quarte augmentée, tout en ayant le même pitch), calculer les accords engendrés par les gammes, reconnaître les noms d'accord à partir de la liste de leurs notes, et inversement, etc. Ce système sert aujourd'hui de base à toute une série d'expérimentations, ayant chacune des objectifs différents. Ecrit en Smalltalk, il comprend une centaine de classes et environ 1500 méthodes. Des éditeurs spécalisés permettent de rentrer et éditer des mélodies et des grilles d'accords connectés sur les modèles objets MusES (Cf. figure ci-contre); une interface rudimentaire permet de relier MusES au monde MIDI (midiFile et utilisation en mode séquencer).

Un système d'analyse automatique de grilles d'accords a été réalisé, en ajoutant une expertise sous forme de règles de production et de méta-règles [Pachet 1994c; Mouton and Pachet 1995]. Un de buts de ce système est d'étudier les rapports étroits entre programmation par objets et programmation par règles, et notamment d'extraire des régularités (des *patterns*) permettant la construction de *frameworks* pour la réalisation de bases de connaissances incluant un raisonnement temporel. A ce titre, un rapprochement inattendu été effectué entre ce système d'analyse musicale et un système de raisonnement temporel médical [Pachet and Dojat 1995], montrant que dans les deux cas, le même type de raisonnement est en œuvre, et que l'on peut le représenter par des librairies de classes et de bases de règles communes. Par ailleurs, ces travaux ont débouché sur d'autres plus formels sur la représentation des relations temporelles dans le cadre de temps circulaire [Pachet et al. 1995].

D'autre part, un système d'harmonisation automatique nous a conduit à nous intéresser au problème de l'intégration de techniques de satisfaction de contraintes dans les langages à objets [Pachet and Roy 1995b]. Le problème musical est ici l'harmonisation de chorales à quatre voix, suivant les règles de l'harmonie et du contrepoint de première espèce. Ce travail, outre son intérêt musical, a donné lieu à des travaux plus spécifiquement liés à l'exploitation de la relation de composition dans les systèmes intégrant objets et CSP [Pachet and Roy 1995a].

Simulation de l'improvisation et de l'accompagnement en jazz

Ce projet a pour but de modéliser les connaissances et processus mis en jeu lors d'improvisation et d'accompagnement en direct en jazz. Les motivations ayant conduit à choisir cette activité musicale comme domaine d'étude procèdent de différentes considérations. D'abord, il s'agit d'une activité difficile à modéliser comme en témoignent plusieurs tentatives effectuées. Ensuite, ce genre musical culturellement bien enraciné est caractérisé par la coexistence de connaissances théoriques et pratiques, peu formalisées, par contraste avec le genre classique du XVIIIe siècle par exemple.

Un modèle simulant le jeu d'un bassiste accompagnant un musicien jouant en solo à partir d'une grille d'accords prédeterminée (un standard) a été conçu. Parmi ses entrées, le système utilise la grille d'accords enrichie par le résultat de l'analyse harmonique présenté plus haut. Pour prendre en compte plus finement l'environnement où se trouve un bassiste de jazz, on a ajouté aux entrées un *Scénario* contenant des événements symboliques qui représentent les actions des autres musiciens et les réactions de l'audience (applaudissements, sifflets).

Ce modèle propose de prendre en compte une *mémoire musicale* contenant des fragments joués et entendus lors de l'improvisation et de l'écoute de morceaux de jazz [Ramalho and Ganascia 1994]. Bien que cette mémoire musicale soit inspirée des travaux sur le raisonnement à partir de cas, le modèle intègre également des mécanismes d'inférence à partir de règles de production. Cette intégration est possible, grâce à une représentation explicite des *actions musicales* à

travers la notion introduite par François Pachet de *PACTs* («Potential Actions in a Context») [Pachet 1991]. Quelques exemples de ces PACTs sont : “jouer la gamme diatonique dans la direction ascendante pendant cette mesure”, “jouer ces notes un ton plus haut”, “jouer de plus en plus fort”, etc. La principale propriété des PACTs est que deux PACTs peuvent être, selon leur compatibilité, combinées pour devenir plus «jouables», c’est à dire capables de produire effectivement des notes. Cette propriété est au cœur de la méthode de résolution de problèmes proposée. D’abord, les PACTs sont *activées* selon les derniers événements du scénario et les accords du segment courant de la grille d’accords. Ensuite, ces PACTs ainsi que d’autres éventuelles PACTs activées précédemment sont *assemblées* par l’application successive d’opérateurs de résolution de conflit de combinaison [Ramalho and Pachet 1994]. Le résultat est une ligne de basse, fabriquée en temps réel au format MusES ou MidiFile (cf exemple de la figure p33).
Notons enfin qu’en explorant les notions de mémoire musicale et de contexte de l’improvisation ce modèle propose une voie alternative aux approches plus courantes pour la simulation des tâches créatives [Ramalho and Ganascia 1993].

Autres travaux

Divers autres travaux de recherche en musique et IA sont effectués au Laforia, dont les deux suivants :

Aide à la programmation de sons

Un travail de recherche dans le domaine de l’*Aide à la synthèse sonore* [Rolland and Pachet 1996] a été réalisé. L’objectif est d’améliorer les interfaces homme-machine des synthétiseurs du commerce, en permettant notamment aux musiciens non-experts de créer («programmer») des sons de manière plus naturelle qu’avec les interfaces existantes. Le moyen que nous proposons est de capturer les connaissances pratiques (savoir-faire) des experts en programmation de synthétiseurs. Le cadre conceptuel et fonctionnel que nous avons défini à cet effet est basé sur une classification des sons en fonction des transformations que l’expert sait leur appliquer. Nous mettons en œuvre une représentation des sons à deux niveaux, combinant programmation par objets (Smalltalk) et logiques terminologiques (BACK), et proposons un schéma d’intégration. Le projet se poursuit autour du système prototype que nous avons développé pour le synthétiseur Korg 05R/W.

Induction de patterns

En musique comme en biologie moléculaire, l’existence de motifs ou fragments récurrents est très fréquente dans les séquences que sont les mélodies et grilles harmoniques d’une part, les protéines et acides nucléiques de l’autre. Dans le domaine musical l’importance des motifs vient du fait que toute régularité structurale récurrente est porteuse d’une sémantique. La détection de motifs peut par exemple permettre d’établir que tel enchaînement d’accords est caractéristique d’un compositeur ou d’un genre musical donnés. De même l’analyse structurale peut montrer que dans les improvisations d’un soliste donné, une phrase A est annonciatrice d’une phrase B, c’est à dire qu’après A on retrouve systématiquement B un nombre déterminé de mesures plus loin. Trois applications sont visées : la reconnaissance stylistique, l’anticipation de séquences musicales dans le cadre de l’accompagnement en direct, et l’aide à l’analyse musicologique de gros corpus musicaux.

Bibliographie

[Mouton and Pachet 1995] **Mouton R and Pachet F**. *Numeric vs symbolic controversy in automatic analysis of tonal music*. IJCAI’95 Workshop on Artificial Intelligence and Music, Montréal, p. 32-40,1995.
[Pachet 1991] **Pachet F**. *Representing Knowledge Used by Jazz Musicians*. International Conference on Computer Music 1991, Montréal, p. 285-288,1991.
[Pachet 1994a] **Pachet F**. *The MusES system : an environment for experimenting with knowledge representation techniques in tonal harmony*. First brazilian symposium on computer music - SBC&M ’94, Caxambu, Minas Gerais, Brazil, p. 195-201,1994a.
[Pachet 1994b] **Pachet F**. *An object-oriented representation of pitch-classes, intervals, scales and chords*. Premières Journées d’Informatique Musicale, LaBRi, Université de Bordeaux, p. 19-34,1994b.
[Pachet 1994c] **Pachet F**. *Vers un modèle du raisonnement dans les langages à objets*. Colloque «Langages et Modèles à Objets», Grenoble, p. 111-123,1994c.
[Pachet and Dojat 1995] **Pachet F and Dojat M.** Un framework pour la représentation de connaissances temporelles en NéOpus, rapport technique Laforia-IBP, 95/19, 1995.
[Pachet et al. 1995] **Pachet F, Ramalho G, Carrive J, Cornic G**. *Representing temporal musical objects and reasoning in the MusES system*. International Congress in Music and Artificial Intelligence, Edinburgh, p. 33-48,1995.
[Pachet and Roy 1995a] **Pachet F and Roy P**. *Integrating constraint satisfaction techniques with complex object structures*. 15th Annual Conference of the British Computer Society Specialist Group on Expert Systems, ES’95, Cambridge, Awarded best technical paper, 1995a.
[Pachet and Roy 1995b] **Pachet F and Roy P**. *Mixing constraints and objects: a case study in automatic harmonization*. TOOLS Europe ’95, Versailles, Prentice-Hall, p. 119-126,1995b.
[Ramalho and Ganascia 1993] **Ramalho G and Ganascia J-G**. *The role of Musical Memory in Creativity and Learning: a study of Jazz Performance*. AI-ED Conference, Edinburgh, GB, p. 349-350,1993.
[Ramalho and Ganascia 1994] **Ramalho G and Ganascia J-G**. *Simulating Creativity in Jazz Performance*. 12th AAAI Conference, Seattle, AAAI Press, p. 108-113,1994.
[Ramalho and Pachet 1994] **Ramalho G and Pachet F**. *From Real Book to Real Jazz Performance*. International Conference on Music Perception and Cognition (ICMPC), Liège (Belgium), p. 349-350,1994.
[Rolland and Pachet 1995] **Rolland P-Y and Pachet F**. *A framework for representing knowledge about synthesizer programming* Computer Music Journal, 20:3, 1996, à paraître.

DOSSIER I.A. ET MUSIQUE

Représentations Musicales
IRCAM

Le groupe de recherche sur les Représentations Musicales de l'IRCAM a été fondé en Juin 1986. Autour d'un noyau permanent de chercheurs — Gérard Assayag, Camilo Rueda, Carlos Agon — des compositeurs invités — Tristan Murail, Antoine Bonnet, Claudy Malherbe etc. — apportent leur expertise musicale et leurs directions de recherche; l'objectif général est la définition de modèles informatisés utilisables dans des situations où le compositeur désire préparer des matériaux musicaux complexes et structurés, relativement à une certaine formalisation ou un certain ensemble de contraintes qui lui sont propres et qu'il est en mesure d'exprimer d'une façon cohérente.

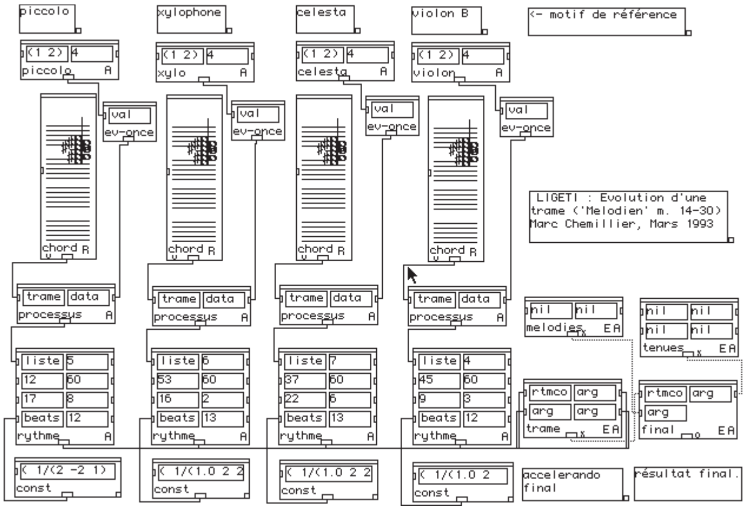
De tels modèles seront utilisables aussi bien pour la production que pour l'analyse musicale. Cette conception de l'assistance informatique comme un aller retour permanent entre analyse et production est solidement établie dans le domaine de la synthèse sonore, mais beaucoup moins développée en ce qui concerne l'aide à la composition; nous espérons développer ce secteur d'une façon significative dans le futur.

Nous sommes concernés principalement — mais pas limités — par l'écriture instrumentale. Ceci explique que nous accordons une grande importance aux paradigmes de la formalisation discrète et du calcul symbolique.

Nous développons cependant de plus en plus nos activités dans le domaine de la musique mixte (instruments / synthèse) et , à cet égard, nous sommes particulièrement intéressés par la relation dialectique entre d'une part la structuration discrète, symbolique, des paramètres musicaux, notamment le temps, et les effets continus inhérents à la synthèse et au traitement des sons. Cette dichotomie implique des systèmes de spécification et de contrôle inédits, et représente un enjeu majeur de la recherche musicale telle que nous la concevons à l'IRCAM.

Il est par conséquent critique pour nous de nous situer à un niveau où des catégories musicales abstraites peuvent être définies et manipulées indépendamment de leurs actualisation sonore. Précisons de surcroît que ces catégories peuvent être a-temporelles — c'est là une chose commune dans la démarche compositionnelle : pensons par exemple à un graphe exprimant des relations harmoniques potentielles. Ce coté a-temporel est des plus difficiles à accéder dans les systèmes de synthèse sonore ou encore dans les programmes dédiés au pilotage de dispositifs MIDI[1] tels que les séquenceurs.

L'ensemble de ces expertises a été concentré dans un logiciel sur lequel l'IRCAM travaille depuis 1990. **PatchWork** est un environnement de programmation visuel et ouvert, aisément accessible au non-informaticien, qui permet de programmer un modèle susceptible d'engendrer des structures musicales puis de visualiser ces dernières sous diverses formes graphiques dont la notation musicale traditionnelle. La composante visuelle permet d'avoir une appréhension schématique de la structure considérée et constitue un excellent apport pédagogique.

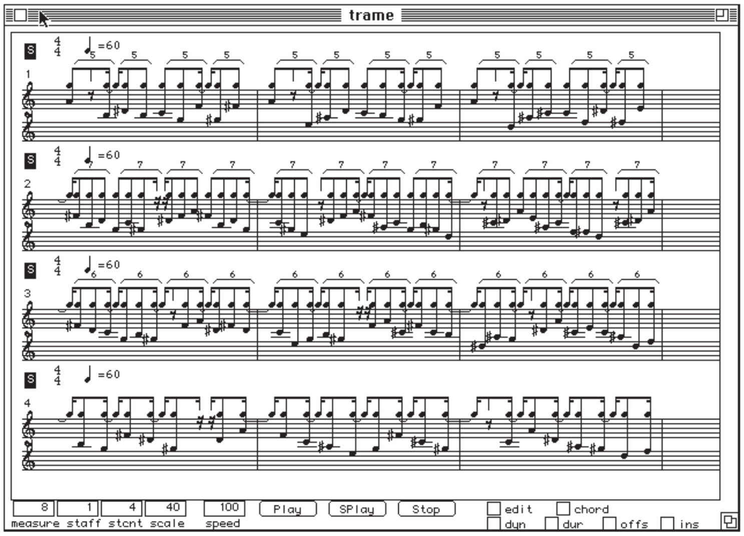


LIGETI : Evolution d'une
trame ('Melodien' m. 14-30)
Marc Chemillier, Mars 1993

Figure 1

Ainsi dans la figure 1, un modèle pour une section de Melodien pour orchestre de G. Ligeti a été réalisé en PatchWork par Marc Chemillier, chercheur à l'université de Caen. Chaque module vertical représente un instrument (piccolo, xylophone, celesta, violon). Dans la partie supérieure se trouve l'entrée des paramètres du modèle, ici un réservoir de notes figuré en notation musicale. Dans la partie inférieure, la boite processus représente l'algorithme qui engendre le flux mélodique, et la boîte rythmes détermine la structure rythmique de la pièce. Ces deux boîtes sont des abstractions au sens informatique du terme : elles masquent des structures plus complexes, accessibles à travers l'interface utilisateur de PatchWork, et qui se présentent de la même manière que le schéma montré ici. Les divers niveaux d'abstraction peuvent être ainsi articulés et limités visuellement au strict nécessaire pour la compréhension d'un niveau d'analyse[2]. La boîte en bas à droite collecte le résultat de la simulation. Elle peut être ouverte elle aussi, livrant le résultat final en notation musicale (figure 2).

DOSSIER I.A. ET MUSIQUE



measure 8 staff 1 stcnt 4 scale 40 speed 100

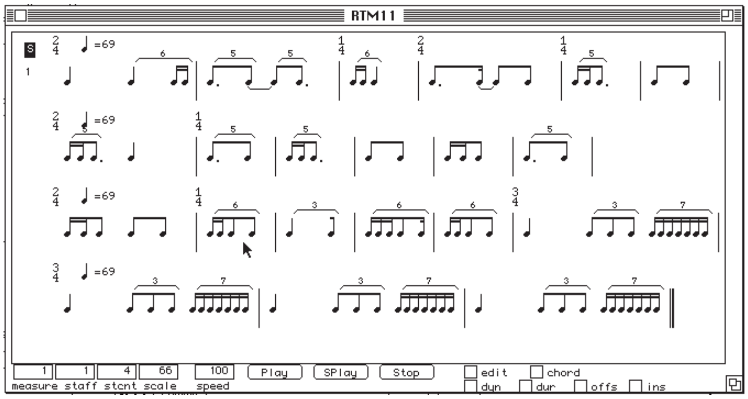
Figure 2

Pour illustrer le type d'opérations qu'un compositeur peut être amené à effectuer dans un environnement d'aide à la composition, nous avons choisi un exemple de *quantification rythmique*. Ce problème, bien connu des compositeurs contemporains, consiste en la transcription selon la notation traditionnelle de la musique d'un flux de durées disponibles sous la forme de valeurs numériques réelles. L'intérêt de la question provient du fait que beaucoup de compositeurs utilisent des formalismes à sortie numérique pour engendrer algorithmiquement des configurations complexes de durées. Ainsi, dans la séquence illustrée par la figure **3** , due à Tristan Murail, des courbes mathématiques (sinus et exponentielle combinées) ont permis de construire un processus d'accélération et de décélérations rythmiques locales enveloppées dans une accélération progressive et globale. Les durées sont figurées de manière proportionnelle. Dans la figure **4**, un traitement automatique des données d'entrée a permis de déceler les éléments de structure qui permettent d'en donner une représentation en notation musicale. Une technique de segmentation automatique a mis à jour un découpage logique de la séquence en fonction des tassements locaux des événements sonores. La métrique (le découpage en mesure) en découle directement. La recherche d'un tempo (une pulsation de base) découle elle aussi de cette segmentation. Pour cela, les mesures sont considérées comme des super-événements ayant leur durée propre et un calcul de plus grand diviseur commun avec approximation est appliqué à ces durées, fixant l'unité de temps qui détermine la cohérence de l'ensemble (ici la noire à 69 sur le métronome).

La validité de la solution proposée peut être mesurée au fait que des éléments de structure saillants (redondance des deux premiers groupes et des quatre derniers groupes par exemple) sont conservés et mis en valeur par le découpage métrique et que la notation finale reste relativement simple. L'esquisse peut ensuite être retravaillée par le compositeur qui dispose là d'un matériau plus proche du langage auquel il est accoutumé.

Patchwork est écrit en Common Lisp / CLOS (Common Lisp Object System) et utilise intensivement les concepts orientés objet dans l'implémentation du modèle visuel. D'autre part, les objets musicaux en sortie des modules, de plus en plus complexes, sont le plus souvent des objets voire des réseaux d'objets. Cependant, cette puissance n'est disponible que pour le programmeur LISP car la syntaxe et la sémantique des schémas visuels (appelés patches) sont purement fonctionnelles (elles sont équivalentes au langage LISP lui même). Il en résulte des difficultés dans la manipulation des objets musicaux qui ressortent d'un modèle pour ce qui concerne leur structure propre et d'un autre modèle pour ce qui concerne le système de contrôle dans lequel ils circulent.

L'expérience accumulée au sein du projet PatchWork et du projet Situation (C. Rueda, A. Bonnet) a mis en évidence les points suivants :
- succès non démenti du concept de *programmation visuelle* auprès des utilisateurs musiciens.
- puissance et flexibilité d'un langage tel que *Lisp / CLOS* combinant *programmation fonctionnelle*, *programmation objet,* et manipulation d'entités symboliques (listes, structures, arbres, etc.)
- confirmation de l'intérêt des musiciens pour un modèle prospectif intégrant le concept de *satisfaction optimale de contraintes* spécifiées par l'utilisateur.



measure 1 staff 1 stcnt 4 scale 66 speed 100

Figure 4

Petit à petit a donc émergé l'idée de réfléchir sur le moyen et long terme à un environnement qui combinerait dès la base de son architecture même ces trois paradigmes. Un tel projet, qui implique une conception informatique au niveau de la structure du langage, dépasse probablement les moyens de l'équipe représentation musicale. Camilo Rueda a donc proposé un projet plus large, financé en partie par COLCIENCIAS (organisme Colombien homologue de notre CNRS) et instituant une collaboration entre chercheurs de l'IRCAM et de trois organismes universitaires (Université de Cali, de Valle et de Manizales) totalisant quatre chercheurs et deux ingénieurs.

Dans un premier temps, des spécifications formelles régissant les aspects syntaxiques et sémantiques entre les trois paradigmes devront être élaborées de manière à constituer une fondation rigoureuse pour l'architecture du système. En particulier, les aspects visuels (programmation et édition) devront être conçus de telle sorte que la sémantique associée aux opérateurs visuels soit spécifiable et éditable par l'utilisateur. Ainsi, il serait tout à fait imaginable que les langages d'implémentation pour la partie visuelle et la partie implémentant la sémantique diffèrent (par exemple C++ et Lisp dans cet ordre).

Dans un deuxième temps, une implémentation du modèle sera proposée et des applications indépendantes développées à titre de validation. Pour ce qui nous concerne, une de ces applications sera PatchWork lui même : outre le fait qu'une telle implémentation de PatchWork sera beaucoup plus maîtrisable qu'elle ne l'est aujourd'hui, nous gageons aussi une plus grande compacité, une plus grande efficacité, une extensibilité accrue, une adhérence à l'évolution technologique des matériels et logiciels système sur micro-ordinateurs (e.g. technologie Open-Doc, Apple-events etc.), l'accès, aujourd'hui exclu, à la programmation et l'édition d'objets, l'accès naturel à la programmation par contraintes, et enfin une indépendance des éditeurs graphiques qui pourront être connectés à d'autres logiciels (e.g. max).

Un projet de collaboration en cours de discussion avec Keith Hamel (University of British Columbia à Vancouver), auteur des logiciels d'édition musicale professionnelle NoteWriter et NoteAbility, pour la mise au point d'éditeurs musicaux modulaires, application-indépendants, pourrait être rattaché à ce projet.

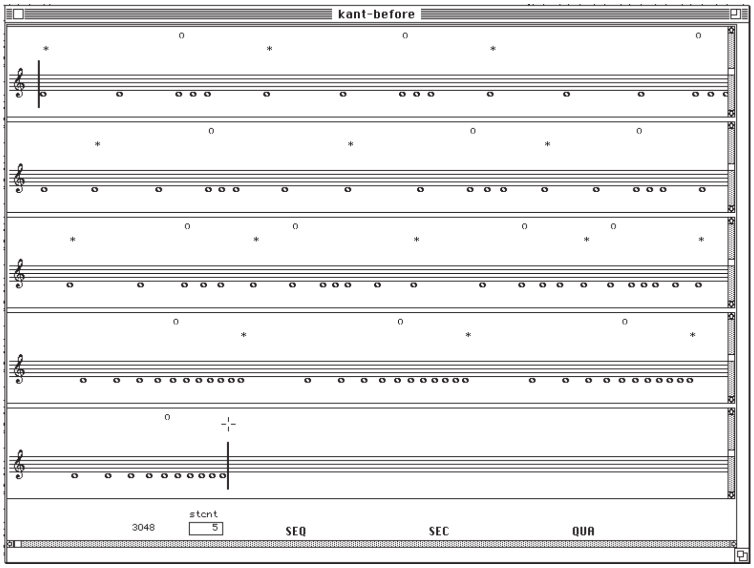


Figure 3

Références

Assayag, G. Cholleton, J.P. Musique, Nombres et Ordinateur. La Recherche, Juillet-Août 1995.
Agon, C. Assayag, G. Fineberg, J. Rueda, C. «Kant: a Critique of Pure Quantification.» Proceedings of the ICMC 94, Aarhus, 1994.
Assayag, G. Cholleton, J.P. L'appareil musical. Résonance. Octobre 1994.
Chemillier, M. Analysis and Computer Reconstruction of a Musical Fragment of LIGETI's Melodien. International Symposium on Music and Mathematics. Bucharest, 1994.
Assayag, G. Rueda, C. «The Music Representation Project at IRCAM.» Proceedings of the ICMC 93, Tokyo, 1993.
Assayag, G. Rueda, C. Représentations Musicales. Intermedia. 93. Barcelone. 1993.
Assayag, G. Composition assistée par ordinateur. Vers la partition potentielle. Les Cahiers de l'IRCAM. Mai 93.
Laurson, M. Duthen, J. «A compositional environment based on Preform II, PatchWork and Esquisse» Proceedings of the ICMC 1990. Glasgow, 1990.
Assayag, G. Timis, D. Computer Printing, Storage, and Transfer of Musical Score, ICMC 87. 1987.
Assayag, G. Timis, D. A ToolBox for Music Notation, ICMC 86. 1986.
Amiot, E. Assayag, G. Malherbe, C. Riotte, A. Duration structure generation and recognition in music writing. ICMC 1986. 1986.
Assayag, G. Castellengo, M. Malherbe, C. Functional integration of complex instrumental sounds in music writing, ICMC 85. 1985.
Assayag, G. Malherbe, C. Cinq questions sur l'audibilité en musique. Conséquence, n. 7-8. 1985.
Assayag, G. Malherbe, C. Manipulation et représentation d'objets musicaux. ICMC 84. 1984.

---

[1] *Musical Instrument Digital Interface*, un protocole de communication ordinateurs/instruments électroniques universellement utilisé.

[2] La formalisation complète de cet exemple pourra être trouvée dans *Les Cahiers de l'IRCAM , La Composition Assistée par Ordinateur*, IRCAM-Centre Georges Pompidou, 1993.

DOSSIER I.A. ET MUSIQUE

LaBRI

Composition musicale assistée par ordinateur

Kablan Barbar, Anthony Beurivé,
Myriam Desainte-Catherine,
Robert Strandh
LaBRI, Université Bordeaux I
email : myriam@labri.u-bordeaux.fr

Équipe

Le groupe de recherche en informatique musicale fait partie de l'équipe «Programmation Symbolique» du LaBRI et est composé de: Robert Strandh, Professeur en informatique à l'Université Bordeaux-1; Myriam Desainte-Catherine, Maître de Conférences à l'ENSERB; Kablan BARBAR, Maître de Conférences à l'Université Bordeaux-1; Antony Beurivé, Thésard en informatique à l'Université Bordeaux-1.
Les deux projets principaux de notre groupe de recherche sont la réalisation d'un éditeur de partitions et le développement (théorique et pratique) d'un modèle hiérarchique codant la problématique de la composition musicale.

I- Éditeur de partitions

Le but de ce projet est de construire un éditeur de partitions musicales comportant un langage d'extension accessible par l'utilisateur. De plus, la qualité de la présentation doit être extrêmement élevée. Ce projet est destiné à être diffusé sous GPL (GNU General Public Licence). Le développement se fait essentiellement à l'aide de projets de programmation en maîtrise d'informatique, mais aussi par d'autres personnes (Thésard CNAM, DESS, DEA, ...). Actuellement, un langage pour la description graphique d'une partition existe et est appelé Scodel. Ce langage servira comme moteur de calcul pour représenter la partition à la fois sur écran et sur papier. Ce projet s'inspire de l'éditeur «Lime» en ce qui concerne les fonctionnalités vues par l'utilisateur. De plus, notre éditeur sera livré avec codes et sources, et contiendra un langage d'extensions de haut niveau.

II- Modèle hiérarchique musical

Notre but est de concevoir un modèle de calcul de données au sein d'une œuvre musicale. Ce modèle doit assister le compositeur dans la mise à jour des données et dans la vérification des règles musicales qu'il s'est lui-même imposé, constituant ainsi en partie «son style musical». Les calculs s'effectuent sur une représentation structurelle et temporelle de l'œuvre, construite par le compositeur, et susceptible de changer au cours du processus de composition. Notre modèle doit réaliser les calculs rapidement et dynamiquement afin de répondre le plus vite possible aux modifications souhaitées par le compositeur. Nous travaillons essentiellement sur ces problèmes de rapidité et d'efficacité des calculs, en tenant compte des besoins d'expression du compositeur. Contrairement aux systèmes de résolution de contraintes (CHIP, CLPR, ...) qui gèrent les problèmes dans leur totalité et permettent, pour certains, l'ajout incrémental de contraintes, nous proposons une meilleure assistance au compositeur dans le conception de son problème (i.e. de sa pièce de musique). Cette assistance permet en particulier la correction du problème en cas d'inconsistance, par le retrait d'une ou de plusieurs contraintes en cause. En contrepartie, nous perdons une part de la puissance des autres systèmes de résolution de contraintes.

Structure hiérarchique

Nous développons un modèle hiérarchique basé sur deux opérateurs, la concaténation et la superposition. Ainsi, la structure temporelle d'un morceau de musique est représentée par un arbre et les caractéristiques musicales (durée, ton, ...) sont exprimées par des attributs. Plus précisément, à un morceau de musique on associe un arbre de dérivation engendré par une grammaire algébrique et les caractéristiques sont codées par des attributs attachés aux noeuds de l'arbre de dérivation. Les équations qui lient les attributs de la grammaire sont associées aux productions et traduisent les règles musicales relatives à la concaténation et à la superposition. La propagation de caractéristiques et le maintien de la cohérence sont basés sur un parcours optimal des nœuds de l'arbre de dérivation associé à la pièce de musique. Dans ce cadre, nous avons travaillé sur les thèmes suivants :
- la généralisation de la définition de la sémantique des opérateurs,
- la détection des incompatibilités et des conflits entre caractéristiques,
- la puissance d'expression de notre modèle hiérarchique,
- le partage de morceaux musicaux (dû aux répétitions musicales).

Sémantique des opérateurs

A chaque opérateur de concaténation et de superposition, on peut associer plusieurs ensembles de systèmes d'équations. Chaque système d'équations définit une sémantique possible de l'opérateur correspondant. Nous avons introduit un formalisme complet pour définir plusieurs sémantiques à un opérateur et défini les caractéristiques d'une pièce par la solution des ensembles de systèmes d'équations associées à l'arbre de dérivation de la pièce (voir [4,5]).

Cohérence d'une pièce et conflit

Le processus de composition musicale est interactif, le compositeur fournissant les valeurs des caractéristiques une à une. Donc, il est impératif de vérifier qu'une nouvelle valeur d'une caractéristique est compatible avec les valeurs déjà calculées sur la pièce de musique. Ce problème, nous l'avons résolu par des algorithmes de parcours d'arbre avec un étiquetage des noeuds de l'arbre. Les algorithmes développés sont incrémentaux et peuvent défaire des calculs déjà effectués sur l'arbre de la pièce de musique (voir [6]).

Puissance d'expression du modèle hiérarchique

La concaténation et la superposition correspondent à 2 des 13 opérateurs d'Allen [1]. Notre papier [7] dresse une comparaison entre ces opérateurs. Nous montrons comment couvrir d'autres opérateurs par des parcours d'arbres et des attributs supplémentaires. La limite d'expression de notre modèle hiérarchique est ainsi clairement définie.

DOSSIER I.A. ET MUSIQUE

Partage de morceaux musicaux

Le partage de morceaux musicaux correspond à la répétition de morceaux dans une pièce de musique; dans notre modèle, cela revient à partager des nœuds et des sous arbres ainsi que leurs caractéristiques; nous travaillons actuellement sur ce thème.

Références

[1]- J.F. Allen. Maintaining Knowledge about Temporal Intervals. Communications of the ACM, 26 (11), pp 832-843 (1983).

[2]- Barbar Kablan et Desainte-Catherine Myriam. A representation of musical pieces based on attributed grammars, ECAI'92 Workshop on AI and Music, Vienna, Autriche.

[3]- Barbar Kablan, Desainte-Catherine Myriam et Miniussi Alain, The semantics of musical hierarchies, Computer musical Journal, volume 17 (4), pages 30-37, 1993.

[4]- Barbar Kablan et Desainte-Catherine Myriam , Using attribute grammars to find solutions for musical equational programs, ACM SIGPLAN Notices, volume 29, number 9, page 56, Septembre 1994.

[5]- Barbar Kablan, Beurivé Anthony et Desainte-Catherine Myriam, Incremental evaluation in music hierarchy, First Brasilian Symposium on Computer Music, pages 99-105, August 1994, Caxumbu, Brésil.

[6]- Barbar Kablan, Beurivé Anthony et Desainte-Catherine Myriam, On the resolution of musical conflit, International Colloquium on new music research, Octobre 1994, Pages 50-57, Ghent, Belgique.

[7]- Barbar Kablan, Beurivé Anthony et Desainte-Catherine Myriam, Structure hiérarchique et relations temporelles d'une pièce de musique, JIM'95 (Journées d'Informatique Musicales), Paris, Avril 1995.

Laboratoire Son et Vision

Institut d'Esthétique et des Sciences de l'Art, Université de Paris l pièce 309, 162 rue St Charles, 75016 Paris.
Directeur : Frank Brown (Maître de conférences, IUT du Mans)
Chercheurs : Philippe Ballesta (Sophia Antipolis) et Rémy Mouton (Paris)
email : brown@uni-lemans.fr

Résumé : *Nos recherches en informatique et musique tonale se poursuivent sur trois axes principaux : lecture automatique de partitions, analyse automatique, composition automatique. Nous avions commencé en 1994 par l'élaboration d'une notation analytique permettant à nos trois domaines de recherches de communiquer et à la constitution des organes de traduction de cette notation vers les notations traditionnelles (portées et MIDI), et réciproquement. Les règles des traités d'écriture ont ensuite été implémentées sous deux formes différentes (automates et programmation par contraintes), aboutissant à la construction de plusieurs automates compositeurs. Les résultats les plus récents et les travaux en cours concernent l'analyse automatique, avec la réalisation d'un analyseur tonal capable de lever les ambiguïtés d'un* MIDIfile, *d'un détecteur de répétitions, d'un analyseur d'»orchestration» et d'un chiffreur harmonique. Ces travaux constituent la matière de deux thèses (soutenances fin 1994 et fin 1995) et de plusieurs articles dans les diverses disciplines abordées (Musicologie, IA, ...).*

Introduction

Le but de nos recherches est d'explorer la partie du travail d'un compositeur susceptible d'être automatisée, et d'étendre au plus cette partie. Nous nous intéressons en priorité à la musique tonale, parce que sa microforme est déjà théorisée, et parce que cette musique «du passé» est relativement dégagée des controverses esthétiques contemporaines. Concrètement, nous essayons de simuler l'activité d'un élève de conservatoire qui lit une oeuvre, l'analyse et écrit une pièce dans le même style.

Notation analytique

La première difficulté que l'on rencontre en voulant traiter de la musique tonale par ordinateur est le choix d'une notation. Une notation à base d'objets graphiques, équivalente à la notation sur des portées, est encombrée d'une quantité d'informations purement graphiques, sans intérêt pour l'analyse (taille et position des hampes, par exemple). La notation MIDI, au contraire, a perdu beaucoup d'informations orthographiques qu'il n'est pas facile de reconstituer. Nous avons donc fait le choix d'une notation analytique qui permette à nos trois domaines, lecture, analyse et composition automatiques, de communiquer aisément. Les diverses traductions nécessaires, de cette notation vers MIDI ou portées et réciproquement, ont été réalisées. Les plus délicates seront évoquées dans la suite de ce rapport, aux chapitres «lecture automatique» et «analyse automatique».

Lecture automatique de partitions

Notre lecteur automatique de partitions de musique fonctionne depuis 1994 et a permis de saisir un certain nombre d'oeuvres pour petites formations (clavier seul, un instrument et clavier, quatuor, ...).
Nous obtenons de bonnes performances, grâce à une utilisation rationnelle mais sans rigidité de connaissances sur la structure du langage graphique considéré. L'analyse d'images se fait traditionnellement en deux étapes indépendantes : la segmentation, utilisant des traitements «lourds» appliqués à toute la surface de l'image (filtres, extraction de contours, de squelettes, ...), et la reconnaissance proprement dite, envisagée comme la répartition de vecteurs-mesures sur l'image entre autant de classes qu'il y a de types d'objets à reconnaître. Les Sciences Cognitives tendent actuellement à montrer que cette dichotomie n'est pas naturelle ; dans notre cerveau, les fonctions «de haut niveau», associées à la connaissance du langage graphique à lire, ne seraient pas l'objet d'une couche spéciale de traitement de l'information, mais se traduiraient par une organisation interdépendante des fonctions «purement perceptives» de bas niveau.

DOSSIER I.A. ET MUSIQUE

Nous avons donc choisi de ne pas séparer les procédures (numériques) de traitement d'image et les procédures (logiques) de mesure-et-test, et d'organiser la lecture en utilisant le plus possible la structure du langage musical écrit.
Le résultat de la lecture est accessible sous trois formes : dans la notation graphique complète de l'éditeur de partitions que nous utilisons, sous forme de MIDIfile et dans notre notation analytique.

L'étape suivante de ce travail est le portage du logiciel de notre ancien Atari sur un PC ou sur une station moderne et plus rapide.

Analyse automatique

Un des buts de la partie d'analyse de notre travail est de faire «mesurer le style» d'une œuvre fournie à l'ordinateur, de manière à ce que le résultat de cette mesure puisse servir à un automate-compositeur à reconstituer une pièce différente, mais de même style.
Nous avons d'abord cherché à nous munir d'outils permettant la saisie d'une quantité significative de musique sous une forme utilisable, la première nécessité était donc de traduire un MIDIfile dans notre notation analytique. La reconstitution correcte de l'orthographe se fait par un suivi tonal partiel du morceau, avec criblage à chaque instant des tonalités possibles (d'après les sons présents à cet instant) et/ou probables (d'après l'environnement et une distance entre tonalités), jusqu'à lever l'ambiguïté enharmonique pour chaque note. Une «solution par défaut» est appliquée dans les cas d'ambiguïté tonale intrinsèque.

Nous avons abordé ensuite l'analyse musicale proprement dite, avec un algorithme qui détecte les répétitions dans un texte constitué des valeurs d'un paramètre dans une mélodie, un analyseur d'»orchestration» et un chiffreur harmonique.

Diverses informations peuvent être obtenues par la détection des répétitions, suivant le paramètre choisi en entrée : le répétitions exactes de la hauteur MIDI, les répétitions à transposition près, en fournissant les intervalles successifs, etc. Une analyse détaillée des résultats permet de repérer facilement thèmes, marches, formules caractéristiques d'un auteur, formules cadencielles, ... La difficulté consistait ici à obtenir une complexité moindre que O(n2), sans bénéficier de la présence du ‘'délimiteur de mots» que constitue le blanc pour un texte littéraire.

Notre analyseur d'»orchestration» prend en compte la répartition des notes dans les diverses voix (*tracks* d'un MIDIfile, portées du système) et effectue un filtrage des hauteurs à petite échelle (un temps ou une mesure) de manière à repérer la présence d'accords plaqués, d'arpèges, de notes étrangères à l'harmonie, etc. ; il cherche à déterminer pour chaque voix s'il s'agit plutôt d'une mélodie ou d'un accompagnement. Notre chiffreur harmonique reprend ces informations ainsi que le suivi tonal approximatif réalisé lors de la reconstitution de l'orthographe, corrige celui-ci et propose pour chaque position la tonalité et le degré les plus vraisemblables pour l'accord en cours (s'il y en a un). La méthode ici est analogue à celle utilisée pour la lecture automatique de partitions, à savoir, nous avons fait collaborer sans les séparer des informations structurelles «de haut niveau» (structure d'un accord, règles d'écriture tonale, ...) et des informations numériques «de bas niveau» obtenues par filtrage (fréquences locales des hauteurs, etc.).

Composition automatique

Nous maîtrisons maintenant relativement efficacement la microforme de la musique tonale, c'est-à-dire ce qui correspond aux règles des traités d'écriture, que nous avons implémentées de deux manières différentes : programmation par contraintes (P. Ballesta) et automate (R. Mouton et F. Brown), y compris l'automate KANTOR qui compose dans divers styles en temps réel. Nous sommes actuellement confrontés à deux problèmes interdépendants : d'une part, le contrôle de la macroforme sans contradiction avec la microforme, et d'autre part, la paramétrisation d'un automate compositeur en vue de lui faire suivre les indications délivrées par la phase d'analyse. Une première idée à ce sujet a été de faire guider la mélodie, créée par un compositeur automatique, par une courbe fractale, ce qui permet de contrôler à la fois son «agitation» (la dimension de la fractale) et son taux de répétitions. Nous avons ainsi réalisé un automate capable de composer des thèmes harmonisés de forme antécédent/conséquent, ayant une ligne mélodique cohérente. Les recherches se poursuivent, entre autres à propos du moyen de mesurer une telle courbe fractale sur de la musique réelle, sur une formalisation plus précise des divers styles tonals, et vers une notion de variation d'un thème ou d'un motif donné.

Présence de la machine BINIOU

P. Barbaud avait toujours souhaité disposer d'une machine capable de jouer une polyphonie en micro-intervalles. La machine BINIOU fut construite par F. Brown en 1986 dans ce but. Le procédé de synthèse sonore utilisé fut qualifié d'algébrique-mixte : numérique en ce qui concernait les générateurs de fréquences et analogique quant aux commandes d'amplitudes ; les contenus spectraux étaient obtenus par synthèse soustractive (filtrage dynamique), additive (cumul de vibrations) ou multiplicative (pseudo-modulation en anneau). Une sortie MIDI (tempérée) permettait d'accroître ses possibilités (10 voix en micro-intervalles et 2 canaux MIDI). Cette machine faisait suite à d'importants travaux logiciels assez ingrats, (synthèse des sons par le programme «Auditu» dont de nombreux résultats sonores sont conservés sur bandes magnétiques) et apportait une révolution dans nos méthodes de travail. P. Barbaud l'utilisa immédiatement de façon intensive et exclusive, jusqu'à son décès en 1990.

Nous avons récemment connecté la machine BINIOU à un PC plus standard (par une liaison série bilatérale), afin de sauvegarder des œuvres ne figurant qu'une seule fois sur des disquettes fragiles et de format non standard. Au passage, la musique tempérée (importante en volume) écrite pour BINIOU peut être récupérée et jouée par une version modifiée de notre automate temps réel KANTOR.

DOSSIER I.A. ET MUSIQUE

Conclusion

Comme on l'a constaté, le laboratoire «Son et Vision» a une approche très pluridisciplinaire de l'Informatique Musicale. Il est susceptible d'accueillir des chercheurs en informatique désirant appliquer leur art à l'une des nombreuses facettes composant la musique par ordinateur.

Références

Barbaud, P. Vademecuum de l'ingénieur en musique (ouvrage posthume) Springer-Verlag, 1993.
Ballesta, P. Contraintes et objets : clefs de voûte d'un outil d'aide à la composition ? Thèse de l'Université du Maine, décembre 1994.
Mouton, R. Outils intelligents pour les musicologues. Thèse de l'Université Paris-I, octobre 1995.
Mouton, R. Analyse musicale assistée par ordinateur. 4e congrès international sur la signification musicale (ICMS'4), Paris, octobre 1994.
Mouton, R. Pachet, F. Numeric vs symbolic controversy in automatic analysis of tonal music. IJCAI'95 Workshop on Music and Artificial Intelligence.

Le laboratoire de recherche en Informatique Musicale de Grame

1. Introduction

GRAME est un centre musical ayant pour vocation le développement et la promotion de la création musicale contemporaine. Le centre réunit au sein d'une même structure quatre secteurs d'activités complémentaires :
* création musicale.
* diffusion.
* formation.
* recherche en informatique musicale.

Ces activités s'appuient sur une équipe de 20 personnes, un collège de compositeurs, un conseil scientifique et un ensemble de moyens matériels dont deux studios de production musicale et un laboratoire de recherche.

Grame est soutenu financièrement par la Direction de la Musique et de la Danse du Ministère de la Culture (département Création et département Recherche), la région Rhône-Alpes, la ville de Lyon et différentes sociétés civiles.

2. Les axes de recherche

Le laboratoire de recherche de Grame développe depuis 1982 une activité de recherche fondamentale et appliquée en Informatique Musicale.

Rappelons que cette discipline dont les premiers travaux remontent au milieu des années 50, se situe au carrefour de plusieurs champs de la connaissance : informatique, acoustique, musicologie, sciences cognitives.

Ses domaines d'application concernent différents aspects de la vie musicale, composition et interprétation des oeuvres, lutherie, enseignement musical, acoustique des salles etc.

Parmi ces problématiques, le laboratoire développe principalement deux axes de recherche ayant pour thèmes l'architecture des systèmes musicaux et les langages formels pour l'écriture musicale.

suite p 42

Conferences :

•ICMC (International Computer Music Conference): la conférence principale, annuelle, en informatique musicale. Gérée par l'ICMA (International Computer Music Association, http: coos.dartmouth.edu/~rsn/icma).

•ICMPC (International Conference on Music Perception and Cognition) conférence plus axée vers la perception et les sciences cognitives, organisée par l'ESCOM (European Society for the Cognitive Sciences of Music).

•ICMAI (International Congress on Music and Artificial Intelligence). Officiellement la seule conférence spécifiquement dédiée à IA et musique. La première édition de cette conférence a été organisée par l'université d'Edinburgh, en septembre 1995. Noter que des séminaires sur IA et musique ont lieu régulièrement à l'Université d'Edinburgh. Contacter emilios@music.ed.ac.uk qui tient aussi à jour une mailing-list «IA and music».

Signalons enfin l'apparition régulière de workshops sur IA et musique dans les conférences d'IA depuis quelques années: ECAI'90, ECAI'92, IJCAI'89, IJCAI'95, AI-Ed 93. Les actes de ce dernier workshop ont été publiés chez Springer-Verlag, *workshops in computing series*, Smith, Smaill & Wiggins Eds. La plupart des présentations des premiers workshops IJCAI sur IA et musique a été rassemblée dans l'ouvrage de Balaban et al.

Revues et journaux principaux :

• *Le Computer Music Journal*: revue principale pour la musique et informatique. Noter en particulier deux numéros spéciaux récents traitant de l'IA et la musique (vol 16, n. 1 et 2, 1992). Deux numéros plus anciens ont fait le point sur l'IA et musique en 1980, et certains articles sont toujours d'actualité (vol. 4, n. 2 et n. 3, 1980). On peut trouver d'autres informations sur le CMJ sur le web: http://www-mitpress.mit.edu/Computer-Music-Journal/CMJ.html.

• *Journal of new music research* (autrefois appelé *Interface*). journal inter-disciplinaire, avec une forte composante IA et modélisation cognitive. Informations sur le web: http://www.swet.nl/jnmr/jnmr.html.

• *Music Perception*. revue inter-disciplinaire, propose régulièrement des articles de psychologie cognitive musicale, d'IA, de musicologie, etc. University of California Press, Diana Deutsch Editor.

2.1. Architecture des Systèmes Musicaux :

2.1.1. Objectifs

Cet axe de recherche se propose d'étudier des méthodes et des outils au service des concepteurs d'applications musicales dans le but de :
* Faciliter la conception, le développement et la mise au point des applications musicales.
* Permettre une prise en compte efficace et simplifiée de la dimension temporelle inhérente aux applications musicales.
* Proposer un cadre de coopération et de collaboration homogène, général et durable aux applications musicales, favorisant une attitude «créative» des utilisateurs.

Les questions abordées concernent bien entendu les systèmes temps-réel, mais également les modèles de communication (la communication étant entendue ici comme le moyen par lequel les parties interagissent pour former un tout), ainsi que l'échange de données multimédia et la normalisation de ces mêmes données. Cette dernière question devient aujourd'hui très pressante car une partie du patrimoine musical contemporain est menacé de disparition à cause, faute de normalisation, d'une trop grande dépendance vis à vis d'un matériel en constante évolution.

2.1.2. Applications

La principale réalisation du laboratoire dans ce domaine est le système d'exploitation musical MidiShare.
Lauréat des trophées Apple 89 et du prix Paris-Cité 90, MidiShare est aujourd'hui adopté par des industriels du logiciel musical et des développeurs indépendants. Tout en offrant des performances élevées, MidiShare simplifie considérablement la conception d'applications musicales temps-réel complexes et facilite le portage entre plusieurs plateformes hardware. MidiShare permet en outre à ces applications de fonctionner simultanément sur une même machine et de collaborer entre elles par le biais d'un système sophistiqué de communications inter-applications.

Plus récemment nous avons développé un protocole de communication temps-réel sous Ethernet qui permet d'étendre le modèle de communication et de coopération de MidiShare à un réseau de machines. Ce système préfigure la version multi-plate-forme distribuée de MidiShare. Nous avons également mis au point OpenShare, un émulateur rendant «compatible MidiShare» les applications développées pour les systèmes OMS et MIDI-Manager.

2.2. Langages Formels pour l'Ecriture Musicale :

2.2.1. Objectifs

Cet axe de recherche vise à mettre au service des compositeurs les possibilités de formalisation et de manipulation symbolique offertes par l'informatique pour :

* faciliter l'élaboration et la mise en œuvre de nouvelles techniques d'écriture musicale ;
* favoriser l'articulation entre formalisation et expérimentation dans la démarche de composition ;
* enrichir les modalités de représentation, d'abstraction et de manipulation des matériaux musicaux, tout en s'inscrivant en continuité avec l'écriture musicale traditionnelle.

Rappelons qu'un langage de programmation, et plus généralement un système de notation, qu'il soit musical, chorégraphique, ou mathématique, n'est évidemment pas un outil neutre, permettant de consigner de manière objective une idée, un phénomène ou un objet. C'est à la fois un moyen de création et de conception, un moyen de connaissance, et un moyen de conservation et de communication. C'est également le plus souvent le résultat d'un processus historique et dialectique de maturation, influencé et influençant le développement de la discipline à laquelle il se rattache.

Dans le domaine musical, l'évolution de la notation musicale du IXe siècle à nos jours est indissociable des transformations et des mutations des pratiques musicales. Elle en est à la fois le reflet et l'un des moteurs essentiels. La possibilité de noter, par des signes conventionnels et abstraits, les phénomènes musicaux et leurs relations temporelles a favorisé l'émergence de nouvelles formes musicales.

Elle a entraîné une séparation progressive et une professionnalisation des métiers de compositeur et d'interprète. Elle a bouleversé l'enseignement et l'apprentissage de la musique. Elle a permis la constitution et la transmission d'un immense patrimoine culturel, toujours vivant, car toujours étudié, interprété, enrichi et re-créé.

La notation musicale s'est avérée être un outil cognitif essentiel. En prolongeant et en amplifiant les capacités mentales de mémorisation et de représentation du compositeur, la notation musicale a contribué à l'élaboration d'une pensée dont la sophistication et la complexité n'ont cessé de croître.

Il est important de considérer l'avènement de langages informatiques pour la composition musicale et la synthèse sonore dans cette perspective, comme un prolongement et une amplification du mouvement d'évolution historique de la notation musicale.

Cette amplification porte sur les rôles traditionnels de la notation musicale : conservation, communication, connaissance et création auxquelles s'ajoute désormais la «computation» comme possibilité de décrire des processus compositionnels et de les "faire faire", de les déléguer à la machine.

2.2.2. Applications

Ces travaux sur les langages de composition musicale ont donné lieu à plusieurs réalisations en particulier MidiLogo (84), MidiLisp (86) en collaboration avec l'IRCAM et ACT informatique, et plus récemment CLCE (90).

La possibilité offerte par ces systèmes d'analyser et de répondre en temps-réel au jeu instrumental, a permis le développement de pièces musicales dites «interactives» dont l'importance artistique ne cesse de croître.

DOSSIER I.A. ET MUSIQUE

Nos travaux portent également sur l'élaboration de nouveaux modèles de programmation tenant compte à la fois de la nature musicale et temporelle des informations manipulées ainsi que du contexte très spécifique, celui de la création artistique, dans lequel ces langages sont employés.

Nous nous intéressons en particulier à des langages dits *homogènes*, dérivés du Lambda-Calcul non-typé, dont les termes sont tous munis d'une dimension temporelle et où les notions usuelles de structuration des objets musicaux (séquence, accord, etc.) s'appliquent également aux programmes de composition que l'utilisateur va pouvoir décrire.

Nous espérons de cette approche un renversement de perspective. Il ne s'agit plus d'envisager l'activité de composition comme une extension de l'activité de programmation, ni le programme informatique comme une représentation de la partition musicale, mais au contraire d'inscrire l'activité de programmation et de description d'opérations compositionnelles dans le prolongement naturel de l'activité de composition.

Pour en savoir plus
Le serveur web du Grame (http://www.grame.fr) permet d'accéder à quelques unes des publications ainsi qu'a différents logiciels tels que MidiShare et OpenShare.

Sélection de publications

[Orlarey 1984] - M-LOGO : un langage de programmation orienté composition musicale - *Proceedings of the International Computer Music Conference 1984*, Computer Music Association, San Francisco.
[Boynton, Lavoie, Orlarey, Rueda, Wessel 1986] - MIDI-LISP, a LISP-based music programming environment for the Macintosh - *Proceedings of the International Computer Music Conference 1986*, Computer Music Association, San Francisco, CA.
[Orlarey 1986] - MidiLogo : a Midi Composing Environment for the Apple IIe - *Proceedings of the International Computer Music Conference 1986*, Computer Music Association, San Francisco.
[Orlarey, Lequay 1989] MidiShare : a Real Time multi-tasks software module for Midi applications - *Proceedings of the International Computer Music Conference 1989*, Computer Music Association, San Francisco.
[Orlarey 1991] - Hierarchical Real Time Interapplication Communications - *Proceedings of the International Computer Music Conference 1991*, Computer Music Association, San Francisco.
[Fober, Letz, Orlarey 1992] - «Notation et représentation formelle des processus de composition» - Rapport de recherche interne- Grame, Lyon.
[Orlarey, Fober, Lertz, Bilton 1994] - Lambda Calculus and Music Calculi - *Proceedings of the International Computer Music Conference 1994*, Computer Music Association, San Francisco.
[Fober 1994] - Real-Time Midi data flow on Ethernet and the software architecture of MidiShare - *Proceedings of the International Computer Music Conference 1994*, Computer Music Association, San Francisco.
[Fober, Letz, Orlarey 1995] - Architectures logicielles pour la musique - *Proceedings of 01Design'95*, Autrans.
[Fober, Letz, Orlarey 1995] - MidiShare, un système d'exploitation musical pour la communication et la collaboration - *Proceedings of the Journées d'Informatique Musicale JIM95*, Paris.
[Letz, Orlarey, Fober 1995] - Real Time Functional Languages - *Proceedings of the International Computer Music Conference 1995,* International Computer Music Association, San Francisco. (à paraitre)

Revue et conférences en France

- *Musurgia*, revue qui fait suite à la *revue d'analyse musicale*. Editions ESKA, 7 rue Dunois, 75013 Paris. Principalement axée sur les techniques d'analyse musicale, mais ouverte aux travaux en informatique et intelligence artificielle.
- Les conférences JIM. Ces conférences rassemblent les chercheurs en informatique musicale francophones. La première conférence JIM a eu lieu à Bordeaux (Cf. compte-rendu dans le CMJ, vol. 18, n. 4, pp. 97-98), la deuxième à Paris (cf. compte rendu dans le bulletin de l'AFIA, n. 22, Musurgia, Vol. 2, n. 3 et CMJ à venir). Sa troisième édition sera tenue à Caen en Mars 96, et sera organisée par Marc Chemillier et Gérard Assayag (mail jim96@canardo.info.unicaen.fr). Les JIMs sont gérées par l'association ADERIM (Association pour le Développement de l'Enseignement et de la Recherche en Informatique Musicale).
- Les actes du colloque *Musique et Assistance Informatique* (Marseille, 1990) contiennent de nombreux articles très intéressants. Laboratoire Musique et Informatique de Marseille (MIM), cité de la musique, BP 8, 13234 Marseille.

Résumé des adresses web

- Studio Grame: http://www.grame.fr
- IRCAM: http://www.ircam.fr, contient des liens vers de nombreuses sources d'information en informatique musicale sur le web.
- Musique au Laforia: http://www-laforia.ibp.fr/~fdp/MusES
- Computer Music Journal: http://www-mitpress.mit.edu/Computer-Music-Journal/CMJ.html.
- Journal of New Music Research, http://www.swet.nl/jnmr/jnmr.html.
- ICMA (association gérant ICMC), http://coos.dartmouth.edu/~rsn/icma.

DOSSIER I.A. ET MUSIQUE

GREYC

Ethnomusicologie et informatique

Marc Chemillier - GREYC
Université de Caen 14032 Caen
marc@info.unicaen.fr

L'ethnomusicologie étudie les formes et les pratiques musicales autres que celles de la musique occidentale savante. Les travaux menés dans cette discipline depuis deux siècles ont montré qu'il n'existe pas de société humaine dans laquelle la musique n'ait pas une place essentielle parmi les activités de la vie sociale. Au même titre peut-être que le langage, la musique doit être considérée, semble-t-il, comme un trait caractéristique de l'« espèce humaine ».

L'informatique permet d'intéressantes applications en ethnomusicologie : analyse acoustique (timbres, échelles), analyse syntaxique (formes, variations, styles), classification de corpus, transcription automatique. Un inventaire de 397 titres d'articles ou d'ouvrages traitant de ces différents sujets est présenté dans [Pacetti 1990].

La présente recherche a été menée dans le cadre du laboratoire CNRS-URA 1526 (Groupe de Recherche en Informatique, Image et Instrumentation de Caen), en collaboration avec l'Université de Paris X (Laboratoire d'Ethnologie et de Sociologie Comparative) et l'UMR 9957 du Musée de l'Homme (Ethnomusicologie). Elle prolonge d'autres travaux similaires effectués au sein de l'UMR 9957 [Estival 1994], et porte sur l'analyse syntaxique d'un répertoire de harpe africaine.

Figure 1

La musique des harpistes nzakara

Les Nzakara vivent à l'est de la République Centrafricaine. Dans le courant des années soixante, Eric de Dampierre (Université de Paris X) a recueilli une collection d'enregistrements de harpistes de cette région. En 1992-93, trois missions sur le terrain ont permis d'enrichir la collection, et de compléter auprès des musiciens nzakara les informations nécessaires à l'analyse du corpus.

La harpe nzakara est une petite harpe à cinq cordes, accompagnant une poésie chantée ayant le caractère d'un récitatif. La partie instrumentale de chaque pièce est composée d'une courte formule de harpe qui se répète avec des variations. L'une des mains du harpiste pince les trois cordes aiguës (*si*b, *sol, mi* dans l'exemple de la figure 1, avec des hampes supérieures), l'autre main pince les trois cordes graves (*mi, ré, do* avec des hampes inférieures). Le répertoire des poètes-harpistes nzakara comporte un ensemble de formules traditionnelles de ce type que les musiciens adaptent à leur style de jeu et qui leur servent de « grilles ».

Formules de harpe « en canon »

Les musiques de tradition orale ont une qualité essentielle : elles ne sont jamais banales. Parfois, certaines combinaisons remarquables s'imposent parmi les autres combinaisons musicales possibles, comme si elles avaient été « filtrées » au cours de la transmission orale. En quoi la formule de la figure 1, par exemple, est-elle remarquable ?

En numérotant 1, 2, 3 les trois notes jouées par chaque main (*si*b, *sol, mi* d'une part, *mi, ré, do* d'autre part), chaque voix de la formule de la figure 1 se réécrit sous forme d'une succession de chiffres. Comparons les deux lignes obtenues, mais en commençant celle du bas à partir du cinquième chiffre :

2 1 3 2 1 1 2 2 1 3
2 1 3 2 1 (3) 2 2 (3) 3

Elles sont identiques à deux exceptions près. Le profil mélodique de la voix supérieure est donc reproduit par la voix inférieure, mais décalé de quatre notes : cette formule de harpe est un canon.

Nous avons isolé dans le répertoire six formules instrumentales « en canon » qui sont regroupées figure 2. Les combinaisons de notes jouées par chaque main sont *a* = (*si*b, *mi*), *b* = (*si*b, *ré*), *c* = (*sol, ré*), *d* = (*sol, do*), *e* = (*mi, do*). Chaque formule est identifiée par un numéro, *Ngb5a* correspond à celle de la figure 1.

Ngb5a = *c b e d b a d c a e*

Ngb10a = *b a b c a e a b e d e a d c d e c b c d*

Ngb10b = *e c d c a d e d b e a e c a b a d b c b*

Ngb10c = *c b c b e d e d b a b a d c d c a e a e*

L10a = *a e a e c b e d e d b a d c d c a e c b c b e d b a b a d c*

L10b = *e a b e d b a b c a e c b c d b a d c d e c b e d e a d c a*

Figure 2

Enumération et singularités

Les recherches en analyse musicale utilisant l'ordinateur consistent souvent à induire du corpus étudié des règles générales de construction, puis à engendrer des variantes à partir de ces règles. Il est rare toutefois que les imitations ainsi produites donnent pleinement satisfaction : en un sens, elles expriment des conditions « nécessaires mais non suffisantes ». Chercher des conditions nécessaires et suffisantes, cela conduit à se demander : en quoi une pièce de musique est-elle singulière ?

Les formules instrumentales de la figure 2 sont toutes construites selon une même règle : différentes permutations circulaires d'une succession donnée des lettres *a, b, c, d, e* sont entrelacées les unes avec les autres. Par exemple, la formule *Ngb5a* est construite en entrelaçant deux cycles obtenus en permutant circulairement *a c e b d*, comme on le voit après séparation des deux cycles :

c - e - b - d -[a -
- b - d -[a - c - e

Les formules *Ngb10a, Ngb10b, Ngb10c* sont ainsi constituées de quatre cycles, et les formules *L10a, L10b* de six cycles.
Un algorithme utilisant les mots de Lyndon permet d'énumérer les formules de harpe comportant un nombre donné de cycles, et vérifiant les conditions suivantes :
(i) pas de formules identiques à une permutation circulaire près,
(ii) pas de répétition de lettre dans une formule,
(iii) pas de factorisation d'une formule en sous-séquence répétée.

Il a été implémenté en Common Lisp, dans l'environnement PatchWork (Ircam) qui permet d'afficher en notation musicale et de jouer sur un synthétiseur les séquences calculées. On obtient 12 solutions dans le cas de quatre cycles, et 134 solutions dans le cas de six cycles.

Le programme a fourni un résultat surprenant dans le cas de deux cycles : il n'y a pas d'autre solution que la formule *Ngb5a*. Ce résultat peut être vérifié « à la main » : on fixe l'un des cycles (*c e b d a*), et on fait varier le second par permutation circulaire. La solution unique vérifiant les conditions est marquée d'une croix :

c c e e b b d d a a
impossible (répétition de lettre)
c e e b b d d a a c
impossible (répétition de lettre)
* *c b e d b a d c a e*
c'est la solution *Ngb5a*
c d e a b|c d e a b
impossible (répétition d'une sous-séquence)
c a e[c b e d b a d
impossible (identique à *Ngb5a*)

La formule *Ngb5a* est donc remarquable en ce sens : c'est un point singulier dans l'espace combinatoire défini par la règle d'entrelacement de cycles, et les conditions (i), (ii) et (iii).

Aspects cognitifs

Comment le processus de transmission orale a-t-il conduit à l'élaboration des structures décrites précédemment ? Y a-t-il dans le mode de penser nzakara un trait particulier expliquant l'apparition de formules de harpe « en canon » ? A ces questions, l'ethnologie apporte un début de réponse. Chez les Nzakara, la naissance de jumeaux nécessite un rituel particulier, qui permet d'apaiser l'inquiétude causée par l'apparition de deux êtres presque identiques. Ce rituel utilise une plante (*bisibili*), à cause de sa forme particulière : celle-ci est constituée de deux séries de feuilles qui ne sont pas symétriques de part et d'autre de la tige, mais sont décalées l'une par rapport à l'autre, dans deux plans différents. La « géométrie » de la plante-des-jumeaux est peut-être à l'origine des formules de harpe en canon, le décalage des deux séries de feuilles donnant un équivalent visuel du décalage des deux lignes mélodiques dans une formule de harpe canonique.

P.S. Merci à Gérard Assayag pour sa relecture et ses conseils.

IRISA

Segmentation et reconnaissance de documents guidées par la connaissance a priori : application aux partitions musicales

Bertrand COÜASNON, Jean CAMILLERAPP, Ivan LEPLUMEY
IRISA-INSA, Équipe IMADOC
INSA Département Informatique
20, avenue des Buttes de Coësmes
35043 Rennes Cedex - France
couasnon@irisa.fr

Lorsque l'on s'intéresse à la reconnaissance de documents, un objectif important est de réaliser un système fiable, afin que l'utilisateur n'ait pas à relire entièrement le document produit par le système, pour détecter et corriger les erreurs. Cette fiabilité peut s'obtenir, d'une part en améliorant la qualité de la reconnaissance et d'autre part en profitant de la redondance pour détecter par le système lui-même ses éventuelles erreurs de reconnaissance.
Dans le cadre de la reconnaissance optique de partitions musicales, l'amélioration de la qualité de la reconnaissance passe par la résolution de problèmes de segmentation :
- les symboles musicaux sont connectés par les lignes de portée ;
- la densité de l'information musicale génère des connexions parasites entre les objets : altération touchant une tête ou une autre altération.

Ces problèmes de segmentation rendent la reconnaissance de ces objets très difficile car pour reconnaître correctement deux objets connexes il faut les segmenter (les séparer) correctement ; mais pour les segmenter correctement il

Références GREYC

CHEMILLIER M., Structure et méthode algébriques en informatique musicale, Thèse, Univ. Paris 7, LITP 90-4, 1990.
CHEMILLIER M., La musique de la harpe, de DAMPIERRE E. (ed.), Une esthétique perdue, Paris, Presses de l'Ecole Normale Supérieure, 1995, pp. 99-205 (sous presse).
ESTIVAL J.-P., Musiques instrumentales du Moyen-Xingu et de l'Iriri, Thèse, Univ. Paris X, 1994.
PACETTI P., Etnomusicologia e computer : bibliografia ragionata e classificazione (1949-1989), Thèse, Univ. degli studi di Roma «La Sapienza», 1990.
République Centrafricaine. Musiques des anciennes cours Bandia, enregistrements, photographies et notice de Marc Chemillier & Eric de Dampierre, disque compact de la collection CNRS/Musée de l'Homme, Le Chant du Monde CNR 2741009 (à paraître, 1995).

faudrait les avoir reconnus avant. Bien que les systèmes existants ne puissent résoudre ce genre de problème [1] [2], il est possible de surmonter partiellement cette difficulté, en utilisant au maximum les connaissances a priori sur l'écriture musicale.

Cependant, dans quelques cas, tous les efforts pour améliorer les capacités de reconnaissance du système resteront insuffisants : le document d'origine peut être trop dégradé pour être reconnu. Ces difficultés résiduelles peuvent être atténuées en permettant au système de détecter les zones dans lesquelles il fait des erreurs afin de les signaler pour réaliser une correction " manuelle " ultérieure. Il reste alors uniquement à vérifier les zones erronées, détectées par le système. Grâce à la redondance de l'information sur une partition (par exemple, l'alignement vertical des notes synchronisées), cette détection est possible. Cependant, pour réaliser cette détection d'erreur il faut à nouveau utiliser au maximum le contexte (*i.e.* la connaissance a priori).

Nous proposons une méthode dans laquelle la connaissance contrôle l'ensemble du processus d'analyse, et plus particulièrement la segmentation et l'étiquetage des objets. Nous avons modélisé la notation musicale à l'aide d'un formalisme grammatical possédant des opérateurs pour définir la position relative des objets. La connaissance est représentée sous la forme d'une grammaire, bien séparée du programme afin de rendre le système beaucoup plus adaptable, ce qui permet également une mise au point plus facile (l'écriture musicale étant complexe à modéliser). Les grammaires hors-contexte pouvant être facilement traduites vers le formalisme des clauses de Horn [3], notre grammaire est implémentée en lProlog [8], avec une connexion à des modules écrits en C pour l'interface avec les opérateurs travaillant sur la matrice image. Grâce à une extension bidimensionnelle des analyseurs de *Definite Clause Grammar* (DCG)[4][3][9], la grammaire est compilée pour produire automatiquement l'analyseur en lProlog. La grammaire a été définie entièrement avec l'aide d'un musicien professionnel.

Le système [5][6][7] est actuellement capable de reconnaître des partitions d'orchestre à portées polyphoniques, en corrigeant certaines erreurs de segmentation (symboles touchant des notes) et en détectant une grande partie des erreurs de reconnaissance induites par une mauvaise segmentation.

Bibliographie

[1] D. Blostein and H. Baird. A critical survey of music image analysis. In Springer-Verlag, editor, *Structured Document Image Analysis*, pages 405–434. Eds. H.S. Baird, H. Bunke, K. Yamamoto, 1992.

[2] N. P. Carter and R. A. Bacon. Automatic recognition of printed music. In Springer-Verlag, editor, *Structured Document Image Analysis*, pages 456–465. Eds. H.S. Baird, H. Bunke, K. Yamamoto, 1992.

[3] W.F. Clocksin and C.S. Mellish. *Programming in Prolog*. Springer-Verlag, 1987.

[4] A. Colmerauer. Les grammaires de métamorphose. Technical report, Groupe d'Intelligence Artificielle, Marseille, France, 1975.

[5] B. Couasnon, P. Brisset, and I. Stephan. Using logic programming languages for optical music recognition. In *International Conference on the Practical Application of Prolog*, pages 115–134, Paris, France, April 1995.

[6] B. Couasnon and J. Camillerapp. A way to separate knowledge from program in structured document analysis: Application to optical music recognition. In *ICDAR, International Conference on Document Analysis and Recognition*, Montreal, Canada, 1995.

[7] B. Couasnon and B. Retif. Using a grammar for a reliable full score recognition system. In *International Computer Music Conference*, Banff, Canada, September 1995.

[8] D.A. Miller and G. Nadathur. A logic programming approach to manipulating formulas and programs. In S. Haridi, editor, *IEEE Symp. Logic Programming*, pages 379–388, San Francisco,, USA, 1987.

[9] F.C.N. Pereira and D.H.D. Warren. Definite clauses for language analysis. *Artificial Intelligence*, 13:231–278, 1980.

Doigtage "intelligent" d'une partition de guitare

Marie-Odile Cordier,
cordier@irisa.fr
IRISA, Campus de Beaulieu, 35042
Rennes Cedex

Equipe concernée :
Projet REPCO de l'IRISA (voir numéro 13 du bulletin pour une présentation de cette équipe).
Responsable de ce logiciel : Marie-Odile Cordier. Logiciel réalisé par des étudiants de l'IFSIC (Institut de Formation Supérieur en Informatique et Communication), de l'Université de Rennes 1, encadrés lors de stages ou projets de fin d'études.

1. Introduction.

Ce logiciel de doigtage automatique d'une partition de guitare s'adresse aux guitaristes débutants ou confirmés et leur permet d'obtenir, à partir d'une partition la meilleure façon de jouer celle-ci. Plus précisément, la partition est saisie à l'aide d'un éditeur de partition qui produit un fichier MIDI et le logiciel calcule le "meilleur" doigtage correspondant. Comme nous l'expliquons dans la section 3, le choix du meilleur doigtage se fait en fonction de préférences; celles-ci sont propres au guitariste et le logiciel lui permet de les modifier facilement. Il peut aussi "forcer" le doigté d'une ou plusieurs notes et laisser le logiciel compléter au mieux le doigtage de la partition.

2. Doigté et doigtage.

La guitare est un instrument à cordes se jouant avec les deux mains ; la main gauche appuie sur les cordes au niveau du manche alors que la droite pince les cordes au niveau de la caisse de résonnance.

On appelle doigté les éléments indiquant comment jouer une note de la main gauche et doigtage la suite de doigtés permettant de jouer une suite de notes.

Une même note peut être jouée de diverses manières : en faisant varier la corde jouée et la case du manche sur laquelle le doigt appuie, il y a en moyenne cinq manières possibles de jouer une même note. A cela s'ajoutent les différentes positions de la main sur le manche et le doigt utilisé pour appuyer sur la corde. Un *doigté* est représenté par un n-uplet (numéro de corde, position du doigt, numéro du doigt, position de la main). La position du doigt est donnée par le numéro de case où le doigt appuie sur la corde. La position de la main est une donnée importante pour le jeu à la guitare et celle-ci sera identifiée par le numéro de la case devant laquelle se trouve l'index. Cette information n'est pas redondante avec les précédentes car le fait de pouvoir écarter plus ou moins un doigt de sa position naturelle permet d'avoir plusieurs positions de main correspondant à un même numéro de case pour la position du doigt.

3. "Meilleur" doigtage.

Un *doigtage* est une suite de doigtés et nous voulons déterminer "le meilleur" doigtage, étant donné un ensemble de notes et un guitariste. Nous associons à chaque doigtage une évaluation numérique par l'intermédiaire d'une fonction ; celle-ci prend en compte la plus ou moins grande facilité pour le guitariste de jouer chacun des doigtés ainsi que la plus ou moins grande facilité de passer d'un doigté au suivant ; elle tient compte aussi de la qualité du son obtenu en fonction du doigté. En effet, une même note jouée selon deux doigtés différents n'a pas la même "couleur" et les doigtés sont donc évalués en tenant compte du type de morceau joué et des effets que le joueur considère comme souhaitables. Nous distinguons par la suite les paramètres, les contraintes et les préférences qui permettent de calculer respectivement les doigtages possibles, les doigtages jouables, et les doigtages préférés.

3.1. Les paramètres

Les paramètres permettent de décrire l'instrument. Ce sont principalement le nombre de cordes (classiquement 6), le nombre de cases (18 pour une guitare classique et 24 pour une guitare électrique) et l'accordage de la guitare.

A un accordage donné correspond une tablature. Une *tablature* donne pour chaque note l'ensemble des couples (numéro de corde, position du doigt) permettant d'obtenir cette note en appuyant sur la corde à l'endroit du manche correspondant à la position du doigt. Le logiciel permet de modifier l'accordage, (de réaccorder) et met à jour la tablature en conséquence.

Ces trois paramètres permettent de calculer pour une note donnée les doigtés possibles pour une guitare donnée ; on obtient des couples (numéro de corde, position du doigt).

3.2. Les contraintes

Les contraintes permettent de ne retenir, parmi les doigtés possibles, que ceux qui sont physiquement jouables. Ce sont :

- la limitation dans l'extension des doigts : un doigté se caractérise par la position de la main (donnée par la position de l'index) et la position du doigt joueur. La position dite "naturelle" (et priviégiée) est celle où, la main étant posée, chaque doigt joue dans la case qui lui fait face (le majeur dans la case immédiatement à droite de celle de l'index...).

Pourtant, il est possible au joueur d'étendre son doigt à droite ou à gauche de cette position naturelle. En général, on considère que cette extension n'est possible que d'une case ; on interdit donc tout doigté dans lequel l'extension est de plus de 1.

- l'impossibilité de croiser les doigts : si l'on joue avec le majeur sur la case 8, il est impossible de jouer une autre note en même temps avec l'annulaire en case 7 (la case la plus haut du manche est la case 1).
- l'interdiction de changer de position de main pour deux doigtés correspondant à deux notes qui doivent être jouées en même temps.

3.3. Les préférences

Les préférences permettent de classer les doigtages jouables et de dégager ainsi les "meilleurs". Les préférences prises en compte par le logiciel sont les suivantes :

- Le déplacement de la main entre deux notes successives : il est plus facile de jouer deux notes successives sans déplacer la main le long du manche. Un critère important est donc le nombre de cases caractérisant le déplacement de la main entre deux notes.
- L'agilité du doigt joueur : l'agilité d'un doigt, évaluée de 1 à 5, indique la capacité du joueur à utiliser ce doigt pour joueur. Par défaut, le petit doigt a une agilité plus faible (2) que les autres doigts (5 pour l'index).
- La région du manche, repérée par le numéro de la case jouée : la partie du manche sur laquelle préfère jouer un guitariste est une question d'expertise (un débutant préfèrera jouer en haut du manche), d'habitude et de sonorités.
- La position naturelle du doigt joueur : celle-ci (expliquée en 3.1) peut être privilégiée en pénalisant l'extension du doigt joueur.
- Le jeu des cordes à vide : ceci correspond à jouer une note en se contentant de pincer une corde à la main droite sans appuyer avec la main gauche. Ceci facilite le jeu du morceau mais est aussi utilisé pour obtenir des sonorités propres à certains types de musique. Ainsi, pour une musique baroque, il est courant de maximiser les cordes jouées à vide.
- Les hauteurs des cordes : plus les cordes sont aiguës, et plus elles sont faciles à jouer; un débutant préfèrera un doigtage sur celles-ci.
- Le jeu en arpège : celui-ci favorise le jeu sur une même corde et n'est souhaitable que pour certains morceaux (Jeux interdits par exemple!).

Ces préférences permettent d'évaluer la qualité des doigtages grâce à une fonction d'évaluation qui effectue une somme pondérée des valuations correspondant à chacun des critères. Les pondérations, qui peuvent aller jusqu'à anihiler le critère, sont fixées par le guitariste et peuvent être modifiées facilement. Cette possibilité est évidemment très importante puisque c'est dans la prise en compte de ces préférences que réside l'"intelligence" du logiciel. Des pondérations par défaut existent qui tiennent compte du niveau du joueur.

4. Algorithme de recherche du meilleur doigtage.

La recherche du meilleur doigtage revient à la recherche d'une solution optimale dans un espace de solutions. Une recherche exhaustive est impossible en raison de la combinatoire du problème. En supposant un nombre de doigtés jouables de m par note, et un nombre de notes de n, on obtient une complexité de m à la puissance n. Heureusement, la structure en couches de l'arbre de recherche nous permet d'utiliser l'*algorithme de Viterbi*. On obtient alors un algorithme de recherche garantissant l'obtention de la solution optimale en temps *polynomial* (en mxn). L'algorithme de Viterbi repose sur les principes de la programmation dynamique, à savoir que toute sous-solution d'une solution optimale est elle aussi optimale. Soit m_i le nombre de doigtés jouables pour la ième note; il est suffisant de ne conserver que le meilleur doigtage aboutissant à chacun des mi doigtés. Ces mi doigtages sont alors prolongés et permettent de calculer les mi+1 meilleurs doigtages aboutissant aux mi+1 doigtés jouables pour la (i+1)ème note.

5. Résultats et Perspectives.

Le logiciel est développé en Turbo Pascal Objet et tourne sur PC. Il prend en entrée un fichier MIDI, permet au joueur d'entrer les paramètres correspondant à sa guitare et de pondérer les préférences en fonction de son niveau et du type de morceau à doigter. Le meilleur doigtage est alors affiché. Le guitariste peut, s'il le souhaite, modifier les pondérations ou forcer le doigté d'une note et obtient alors la nouvelle meilleure solution.

Le logiciel a été testé sur plusieurs partitions telles que "Jeux interdits", "Stairway to heaven", "Owner of a lonely heart", "Musique pour les silences futurs" de Dan Ar Braz ou "1er prélude en Do Majeur pour clavecin" de Bach transcrit par M. Dadi. Les doigtages obtenus sont de bonne qualité ; il est possible d'obtenir des doigtages très proches de ceux de référence et, bien sûr, en faisant varier les pondérations des doigtages différents mais souvent intéressants.

La principale limite actuelle du système est son impossibilité à traiter le cas où plus de deux notes doivent être jouées en même temps ; il ne traite pas vraiment correctement les barrés et demi-barrés. D'autre part, le logiciel ne tire pas parti de l'existence d'accords, alors qu'un guitariste expérimenté aura tendance à s'appuyer sur eux pour trouver son doigtage. La prise en compte des accords s'est avéré assez incompatible avec l'approche retenue s'appuyant sur l'algorithme de Viterbi ; de plus les résultats obtenus montrent que le logiciel, sans les utiliser, "retrouve" souvent les accords lorsqu'ils sous-tendent naturellement une partition.

Références

1- **Viterbi, J.** Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm» IEEE Transactions on Information Theory, vol IT-13(2):260-269.

2- **Sayegh S. I.**, Fingering for String Instruments with the Optimum Path Paradigm, Computer Music Journal, vol 13, 3, 1989.

Les dossiers du *Bulletin de l'AFIA*

Apprentissage	Bulletin n°22
Les explications dans les SBC	Bulletin n°20
Pétrole-Chimie	Bulletin n°19
Le raisonnement à partir de cas	Bulletin n°18
I.A. et temps-réel	Bulletin n°17
Planification et action	Bulletin n°16
Traitement automatique des langues	Bulletin n°15
I.A. et médecine	Bulletin n°14
Diagnostic à base de modèles	Bulletin n°13
Validation des SBC	Bulletin n°12
Le connexionnisme	Bulletin n°11
I.A. et jeux	Bulletin n°10
E.I.A.O.	Bulletin n°9
I.A. et gestion	Bulletin n°8
Conception et I.A.	Bulletin n°7
Intelligence artificielle distribuée	Bulletin n°6
Acquisition des Connaissances	Bulletin n°5
IA et ordonnancement	Bulletin n°4

Présentation du bulletin

Le **Bulletin de l'AFIA** est le bulletin de l'**Association Française pour l'Intelligence Artificielle**. Il vise à fournir un cadre de discussion et d'échanges au sein de la communauté universitaire et industrielle. Ainsi, toutes les contributions, pour peu qu'elles aient un intérêt général pour l'ensemble des lecteurs, sont les bienvenues. En particulier, les annonces, les compte-rendus de conférences, les notes de lecture, les articles de débat sont particulièrement recherchés. Le Bulletin de l'AFIA publie également des dossiers plus substantiels sur différents thèmes liés à l'IA.

Pour contacter l'AFIA

Jean-Marc DAVID (Président)
Renault DSCIT Service 0497
860, Quai Stalingrad
92109 Boulogne- Billancourt

Liens avec les adhérents
René QUINIOU
INRIA / IRISA
Campus de Beaulieu
35042 Rennes Cedex
e-mail : quiniou@irisa.fr

Parrainage de manifestations
Claudette SAYETTAT
Ecole des Mines
Centre SIMADE
158 cours Fauriel
42023 Saint Etienne Cedex 2

Philippe LAUBLET
Rédacteur en chef du bulletin
CAMS - Université Paris 4
96 boulevard Raspail, 75006 PARIS
laublet@laforia.ibp.fr

afia
ASSOCIATION FRANCAISE POUR L'INTELLIGENCE ARTIFICIELLE

Imp. Augustin Paris - 40.36.10.15