



HAL
open science

L'algorithmie cognitive et ses applications musicales

Joséphine Calandra

► **To cite this version:**

Joséphine Calandra. L'algorithmie cognitive et ses applications musicales. Musique, musicologie et arts de la scène. Sorbonne Université, 2023. Français. NNT : 2023SORUL148 . tel-04581505

HAL Id: tel-04581505

<https://theses.hal.science/tel-04581505>

Submitted on 21 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



SORBONNE UNIVERSITÉ

ÉCOLE DOCTORALE V

IReMus - LaBRI

T H È S E

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ SORBONNE UNIVERSITÉ

Discipline : Musique et musicologie

Présentée et soutenue par :

Joséphine CALANDRA

le : 22 décembre 2023

**L'Algorithmie Cognitive
et ses applications musicales**

Sous la direction de :

M. Jean-Marc CHOUVEL – Professeur, Sorbonne Université

Mme Myriam DESAINTE-CATHERINE – Professeure émérite, Bordeaux INP

Membres du jury :

M. Gérard ASSAYAG – Directeur de Recherche, IRCAM - Président du Jury

M. Frédéric BIMBOT – Directeur de Recherche, CNRS, IRISA - Rapporteur

M. Francis COURTOT – Maître de Conférences, Université de Lille - Examinateur

Mme Florence LEVÉ – Professeure, Université de Picardie - Rapporteur

Mme Marie TAHON – Professeure, Le Mans Université - Examinatrice

« La catastrophe des concepts, pour la science, n'est jamais qu'imminente. La science la repousse toujours plus loin, vers les "deux infinis" du temps et de l'espace.

Pour l'art, la catastrophe est actuelle et interne. Elle est constitutive de sa pensée : l'art est la pensée de l'énigme. L'art pense le réel, ici même et maintenant, en ce que sa présence échappe aux concepts de l'espace et du temps.»

- Marc le Bot, *Les conditions de l'espace et du temps*

Remerciements

Je souhaite tout d'abord remercier mes deux directeurs de thèse Myriam Desainte-Catherine et Jean-Marc Chouvel, d'avoir cru en moi en premier lieu, en m'intégrant dans les équipes du SCRIME pour mon stage de fin d'études tout d'abord puis en me permettant de travailler sur cette thèse, dont la multi-disciplinarité et l'étendue des champs de réflexion m'anime. Je remercie mes directeurs pour leurs enseignements riches et leur présence durant ces quatre années mais je les remercie également chaudement pour leur compréhension et leur bienveillance face aux difficultés de santé dont j'ai pu faire face au cours de ces années de thèse.

Merci ensuite aux membres du jury pour avoir accepté d'assister celle-ci : merci à Frédéric Bimbot pour son intérêt et les nombreux échanges que nous avons pu avoir, ainsi que Florence Levé pour ses conseils, et à tous deux d'avoir accepté de prendre de leur temps précieux pour être rapporteurs de cette thèse. Merci à Francis Courtot et Marie Tahon d'avoir dans un premier temps guidé les évolutions de ma thèse avec bienveillance en tant que membres de mon comité de suivi et d'avoir accepté de clore joliment ce suivi en tant que membre du jury. Je remercie également Gérard Assayag pour son intérêt pour le travail apporté dans le cadre de cette thèse et dont les travaux apportent une dimension passionnante à ce sujet, ainsi que pour avoir également accepté de faire partie du jury de cette thèse.

Je remercie par ailleurs le SCRIME en général pour m'avoir, dès mon arrivée à l'ENSEIRB-Matmeca, fait prendre conscience de la possibilité réelle de travailler dans l'informatique musicale et m'avoir permis de réaliser concrètement ce projet. Je remercie ses gestionnaires, Annick Mersier dès le début, rejointe ensuite par Remy Piazza pour leur travail important et leur disponibilité au cours des différentes tâches administratives, mais également pour leur grande gentillesse et soutien. Je remercie également Pierre Hanna pour son aide conséquente d'abord dans le cadre de ses enseignements au côté de Matthias Robine, puis pour son encadrement au cours de mon stage et ensuite sa disponibilité et son aide précieuse aux débuts de cette thèse.

Je remercie chaleureusement les étudiants acteurs de cette thèse, qui ont pu apporter une dimension concrète et interdisciplinaire à celle-ci dans le cadre de leurs stages : Robin Montfermé de l'Université de Bordeaux, Erwan Michel de l'Université Toulouse Jean-Jaurès, Asia Auville de l'ENSEIRB-Matmeca, Anastazja Adamczyk de l'Université de Bordeaux et Elia Bichoy de L'Université Sorbonne Université. Je remercie également les 6 étudiants de l'ENSEIRB-Matmeca et les 3 étudiants de Sorbonne-Université qui ont choisi de travailler sur ma thèse en tant que projet universitaire et dont le travail est fondamental pour cette thèse.

Je remercie l'ensemble des doctorants, professeurs et acteurs de l'informatique musicale et de la musicologie computationnelle avec qui j'ai pu avoir des échanges riches sur le sujet de ma thèse.

Sur une touche plus personnelle, je souhaite remercier les doctorants aux côtés de qui j'ai pu participer dans le cadre du concours *ma thèse en 180 secondes* : le travail effectué avec une si jolie cohésion a été un véritable plaisir qui s'est ressenti au-delà du concours. Je remercie également chaleureusement les collègues du LaBRI, de l'AFoDIB mais pas que, pour les moments quotidiens de bonne humeur, les événements divers, les soirées jeux et les sorties escalade qui ont rendu le travail conséquent plus joyeux chaque jour. Je vous souhaite à tous beaucoup d'épanouissement dans vos projets personnels et professionnels.

Ce travail n'aurait par ailleurs pas été possible sans le soutien de mes proches que je remercie du fond du cœur. Tout d'abord, je souhaite remercier ma famille, mon père Jean-Philippe Calandra et ma mère Valérie Henrion-Calandra pour m'avoir insufflé la passion pour la musique et m'avoir toujours encouragée pour mon attrait vers les mathématiques et la logique. Je remercie mon frère Théotime, ma sœur Marine et mes cousins Arthur et Bianca qui sont passionnés et inspirants, ainsi que mes grands-parents et mes oncles et tantes pour leur soutien depuis toujours et leur aide précieuse tout au long de cette thèse.

Je remercie mes amis pour leur bienveillance et leur soutien : merci à Sophie Juppet pour sa présence depuis le début de mon aventure dans l'informatique, Pierre Kamlay pour avoir partagé mon quotidien durant la majorité de cette thèse et m'avoir épaulée lors de mes soucis de santé, Léa Darwich et Dexter Quincy-Jones pour leur soutien durant cette période de rédaction ainsi que Léa Joséfowich pour avoir embelli ces temps de rédaction également en m'offrant l'occasion de rédiger au milieu des montagnes et de mes amis. Un merci particulier à N'gie Aimé, Antoine Rolin, Charles Lorentz et Sophie pour leur intérêt envers ma thèse et leur relecture de ce manuscrit.

Merci finalement à mes proches, avec qui je partage mon quotidien et mes voyages et qui me font voyager au quotidien, de m'aider à aboutir à des projets tels que celui-ci.

Table des matières

Résumé étendu en français	1
Introduction	3
Contexte	3
Objectif	6
Périmètre de l'étude	8
Contributions	9
Plan	10
1 Etat de l'Art	13
1.1 Analyse musicologique de la forme	13
1.1.1 Paysage de l'analyse musicale	14
1.1.2 Analyse de la forme	16
1.1.3 Analyse sémiologique	17
1.2 Vers une modélisation de la cognition et de la psychologie de l'écoute	18
1.2.1 Psychologie de l'écoute	19
1.2.2 Prise en considération de la musicologie cognitive	19
1.3 Musicologie computationnelle	20
1.3.1 Traitement du signal	21
1.3.2 Logiciels et algorithmes d'analyses	24
2 L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS	29
2.1 Le Diagramme Formel	30
2.1.1 Présentation du Diagramme Formel de J.-M. Chouvel	30
2.1.2 Définitions de notre modèle	34
2.1.3 Algorithme	34
2.1.4 Formalisation	37
2.1.5 Exemples du <i>Geisslerlied</i> et du <i>Rondo</i> extrait du <i>Menuet K545</i> de W.A. Mozart	39

2.1.6	Équivalence entre les représentations symboliques et le Diagramme Formel	41
2.1.7	Formalisation temps réel	41
2.1.8	Pré-traitement	44
2.1.9	Prise en compte des vecteurs en entrée de MORFOS	46
2.1.10	Polyphonie et multiplicité des échelles de temps	48
2.2	L'Algorithme MORFOS : Hiérarchisation	49
2.2.1	Hiérarchisation du modèle	50
2.2.2	Formalisation	50
2.2.3	Exemple du <i>Rondo</i> extrait du <i>Menuet K545</i> de W.A. Mozart	51
2.2.4	Formalisation temps réel	52
2.2.5	Échelles musicales	56
2.3	Les trois plans	58
2.3.1	Présentation des plans	58
2.3.2	Formalisation	60
2.3.3	Applications	61
2.4	Agenda	63
2.4.1	Écoutes	63
2.4.2	Similarité	69
2.4.3	Segmentation	71
2.4.4	Multi-dimentionnalité	73
3	Architecture et Implémentation de MORFOS	77
3.1	Architecture du logiciel	78
3.1.1	Pré-traitement des données	79
3.1.2	Module de segmentation	79
3.1.3	Module de classification	79
3.1.4	Implémentation des structures de données	80
3.1.5	Visualisation du diagramme formel	81
3.1.6	Interface des paramètres	82
3.1.7	Algorithme MORFOS	83
3.2	Implémentation	84
3.2.1	Pré-traitement	85
3.2.2	L'oracle Multi-Echelle	92
3.2.3	L'algorithme MORFOS	101
3.2.4	Lien signal/symbolique	108
3.3	Règles	110
3.3.1	Règles de Classification	110
3.3.2	Règles de Segmentation	116
3.4	Paramétrisation	126
3.4.1	Données à analyser	127

3.4.2	Critères de pré-traitement	127
3.4.3	Critères de similarité	128
3.4.4	Critères de segmentation	128
4	Un Modèle d'Écoute : coûts et cognition	131
4.1	Bibliographie sur la modélisation des phénomènes cognitifs	133
4.1.1	Structure et Complexité	133
4.1.2	Les phénomènes d'attention	134
4.1.3	La charge cognitive	135
4.2	Complexité	136
4.2.1	Définition de la complexité	137
4.2.2	Complexité par calcul des opérations	142
4.2.3	Complexité par calcul de similarité	147
4.2.4	Analyses	149
4.3	Analyse de second ordre et phénomènes d'attention	157
4.3.1	Phases cognitives	157
4.3.2	Analyses	163
4.4	Comparaison aux modèles cognitifs	168
4.4.1	Modèles d'écoute par EEG	168
4.4.2	Calcul de la charge cognitive par étude double tâche	174
5	Visualisations des diagrammes formels multi-échelle	189
5.1	État de l'art pour la visualisation musicale	190
5.1.1	Représentations manuscrites	191
5.1.2	Vers une représentation de la musique audio	191
5.2	Les trois visualisations	193
5.2.1	Visualisation 2D	193
5.2.2	Visualisation 3D en pavés	199
5.2.3	Visualisation 3D sphère	200
5.3	Visualisation des phénomènes d'attention	206
5.3.1	La mémoire	206
5.3.2	Les hypothèses	206
5.4	Interface	207
5.4.1	Agenda	208
5.4.2	Interface graphique : visualisation de l'analyse	209
5.4.3	Gestion de l'audio	213
5.4.4	Lien entre l'interface et MORFOS	214
5.4.5	Gestion des projets musicologiques	216
5.4.6	Améliorations	217

6 Applications et Analyses Musicologiques	221
6.1 Analyses pour l'étude du logiciel	222
6.1.1 Geisslerlied	222
6.1.2 <i>sonate pour piano n°16 en Do Majeur</i> de W.A. Mozart . . .	232
6.2 Analyses musicologiques sur un corpus spécifique	237
6.2.1 <i>Passion selon Saint Matthieu BWV.244</i> , J.S. Bach	238
6.2.2 <i>Concerto pour violon en la mineur RV356</i> , A. Vivaldi	238
6.3 Étude de la base RWC Pop	241
6.3.1 Création de nouvelles règles et uniformisation des niveaux .	241
6.3.2 Analyse MIREX	242
6.3.3 Études spécifiques des pièces RWC Pop 11 et RWC Pop 21 .	244
6.4 Analyses dans le cadre du cours <i>Outils numériques pour l'analyse musicale</i>	247
6.4.1 Présentation de l'interface prototypique	248
6.4.2 Solo de cor anglais de l'acte III de <i>Tristan et Isolde</i> , Richard Wagner	250
6.4.3 <i>Musik to heare</i> , Igor Stravinsky	253
6.4.4 <i>L'Esperanza</i> , Sven Våth	257
6.4.5 Bilan	258
Conclusion et Perspectives	263
Résumé	263
Perspectives	265
A Formalisation temps réel : exemple du <i>Rondo</i> de W.A. Mozart	267
B Algorithme de compression du diagramme formel multi-échelle	271
C Complexité : calcul du nombre de liens de l'oracle	277
D Corpus musical de l'expérience cognitive	279
Lexique	281
Références	283
Publications	295

Table des figures

1	Représentation schématique de la constitution structurée des objets d'un univers, en parallèle avec un schéma de la tripartition conceptuelle de la forme en espace, modèle et objet, extrait de [33].	6
1.1	Forme d'onde du <i>la 440 Hz</i> (sinusoïde pure, haut) et Forme d'onde du <i>fa 349,23 Hz</i> (somme de sinusoïdes, bas).	21
1.2	Forme d'onde du <i>Rondo</i> de la <i>Sonate pour piano n° 16</i> de Wolfgang Amadeus Mozart, obtenus avec le logiciel <i>Audacity</i>	22
1.3	Spectre d'une sinusoïde pure en la 440 Hz (haut) et d'un fa 349,23Hz joué au piano (bas), obtenus avec le logiciel <i>Audacity</i> (fenêtre de Hann, 1024 échantillons).	23
1.4	Sonagramme du la 440 Hz joué au piano (haut), du fa 349,23 Hz joué au piano (milieu) et du <i>Rondo</i> de Mozart (bas), obtenus avec le logiciel <i>Audacity</i> (fenêtre de Hann, 1024 échantillons).	24
2.1	Partition (haut) et diagramme formel (bas) du <i>Geisslerlied Maria muoter reinû maît</i> au niveau structurel de la note.	31
2.2	Représentation du resserrement sur les premières mesures du <i>Geisslerlied</i> à l'échelle de la note (haut). L'extrait est regroupé en trois parties a, b et c, qui sont reprises tout au long de la pièce telles que représentées sur la figure du bas. On retrouve la figure de resserrement à l'échelle de la mesure.	32
2.3	Partition (haut) et diagramme formel par Jean-Marc Chauvel (bas) du <i>Rondo K545</i> de W.A. Mozart. Les premiers matériaux mélodiques élémentaires considérés sont encadrés en pointillés sur la partition.	33
2.4	Création en temps-réel du diagramme formel à l'échelle de la note du <i>Geisslerlied</i>	36
2.5	Diagramme de l'algorithme MORFOS.	36
2.6	Formalisation des différentes représentations initiale, segmentée et finale dans l'Algorithme MORFOS.	39

2.7	Diagramme formel final automatisé du <i>Geisslerlied</i> . Le premier matériau correspond au silence qui débute le fichier audio.	42
2.8	Diagramme formel final automatisé du Rondo K545 de W.A. Mozart.	43
2.9	Automate des opérations de l'Algorithme MORFOS. Chaque état correspond à une opération effectuée <i>atr</i> , <i>sgm</i> ou <i>cls</i>	44
2.10	Représentation du logiciel MORFOS constitué d'une phase de pré-traitement et de l'Algorithme MORFOS.	45
2.11	Diagramme formel initial automatisé du <i>Geisslerlied</i> obtenu à partir d'un fichier MIDI synthétisé avec un son de piano à 120bpm. Chaque objet est une frame de 0,023s : chaque bloc d'une seconde est donc en réalité divisé en une quarantaine de sous-blocs. Les nuances de gris correspondent ici à l'information de la dynamique (plus le bloc est foncé, plus la dynamique est élevée). Comme pour le diagramme formel final présenté précédemment, un premier matériau correspond au silence qui débute le fichier audio : la dynamique est nulle et les objets sont donc blancs.	46
2.12	Diagramme formel initial automatisé du Rondo K545 de W.A. Mozart.	47
2.13	Représentation de MORFOS : une phase de pré-traitement interne permet de fournir un ensemble de descripteurs audio et éventuellement une représentation symbolique à partir d'un fichier audio (diagramme du haut), et d'une phase de pré-traitement externe fournissant une représentation symbolique à partir d'une représentation audio ou symbolique (diagramme du bas). Ces représentations initiales sont ensuite fournies à l'Algorithme MORFOS dont les contraintes de segmentation et de classification divergent en fonction du format vectoriel ou symbolique.	48
2.14	Organisation de MORFOS pour les représentations vectorielles. Celles-ci sont introduites en tant que représentation segmentée et sont soumises directement à l'opération de classification.	49
2.15	Organisation de MORFOS : le logiciel est constitué de la phase de pré-traitement ainsi que de l'Algorithme MORFOS itéré à plusieurs niveaux.	51
2.16	Diagramme formel multi-échelle du <i>Rondo K545</i> de W.A. Mozart.	53
2.17	Automate des opérations de l'Algorithme MORFOS. Chaque état correspond à une opération et les arcs correspondent au niveau de l'état affecté par l'opération vers laquelle la flèche pointe.	55
2.18	Superposition des diagrammes formels dans le diagramme formel multi-échelle.	59
2.19	Plans de la forme (de niveau 0), de la structure (tous matériaux considérés) et de l'organisation (à la fin de l'analyse).	60

2.20	Plan de la forme, de l'organisation (et de la structure) fixes et changement de matériau.	62
2.21	Plan de l'organisation fixe et modification du plan de la forme et de la structure.	63
2.22	Plan de la structure fixe et modification du plan de l'organisation et de la forme.	64
2.23	Diagramme formel du <i>Geisslerlied</i> transposé à la tierce.	68
2.24	Diagramme formel du <i>Geisslerlied</i> avec les durées multipliées par 2.	68
2.25	Diagramme formel du <i>Geisslerlied</i> avec pour matériaux les motifs rythmiques.	73
2.26	Diagrammes formels ayant pour matériau la note (haut) et le rythme (bas) associés à la partition représentée au dessus de chaque diagramme.	75
3.1	Module de pré-traitement.	79
3.2	Module de segmentation.	80
3.3	Module de classification.	81
3.4	Module de structure des données.	81
3.5	Module de visualisation.	82
3.6	Module de paramètres.	83
3.7	Architecture de MORFOS.	84
3.8	FFT Classique.	87
3.9	FFT Bandes.	87
3.10	FFT fenêtres.	88
3.11	Oracle Multi-échelle.	93
3.12	Oracle de Markov Variable induisant la chaîne de caractère <i>abacabacdeab</i>	94
3.13	Oracles et Tables des Liens sur l'exemple du <i>Rondo</i> K545 de Mozart à l'acquisition du 5eme objet de niveau 0.	96
3.14	Oracles et Tables des Liens sur l'exemple du <i>Rondo</i> K545 de Mozart à l'acquisition du 7eme objet de niveau 0.	97
3.15	Diagrammes formels sur l'exemple du <i>Rondo</i> K545 de Mozart à l'acquisition du 5eme objet de niveau 0 avant et après segmentation et classification au niveau 1.	100
3.16	Distance euclidienne.	112
3.17	Similarité cosinus.	113
4.1	équivalences entre les phénomènes cognitifs et MORFOS.	132
4.2	Phénomènes cognitifs liés à l'écoute musicale : la classification/segmentation, les phénomènes d'attention et la charge cognitive.	133

4.3	Diagramme Formel Multi-échelle du <i>Rondo K545</i> de W.A. Mozart dans le cas sans segmentation.	150
4.4	Complexité lors de l'acquisition de chaque objet dans le cas sans segmentation.	151
4.5	Diagramme Formel Multi-échelle et complexité du <i>Rondo K.545</i> de W.A. Mozart dans le cas où les facteurs de taille 2 sont segmentés.	152
4.6	Exemples de Diagramme Formel Multi-échelle du <i>Rondo K.545</i> de W.A. Mozart dans le cas de deux segmentations aléatoires (haut, 1 à gauche, 2 à droite) et complexités correspondantes (bas).	153
4.7	Diagramme Formel Multi-échelle et complexité du <i>Rondo K.545</i> de W.A. Mozart avec l'ensemble A des règles sélectionnées dans l'Agenda.	154
4.8	Diagramme Formel Multi-échelle du <i>Rondo K.545</i> de W.A. Mozart avec l'ensemble B des règles sélectionnées dans l'Agenda.	155
4.9	Complexité lors de l'acquisition de chaque objet avec l'ensemble B des règles sélectionnées dans l'Agenda.	156
4.10	Modélisation des phases cognitives.	160
4.11	Phase 1.	161
4.12	Phase 2.	161
4.13	Phase 3.	162
4.14	Phase 4.	163
4.15	Phases cognitives du <i>Rondo K545</i> de W.A. Mozart selon Chouvel dans [32].	166
4.16	Modèle de S. Koelsch et W.A. Siebel [79] et comparaison avec les étapes de MORFOS.	172
4.17	Tâche de reproduction de séquences.	176
4.18	Définitions introduites dans l'expérience cognitive.	177
4.19	Questionnaire NASA-TLX.	180
4.20	Extrait de la phase de pondération du questionnaire NASA-TLX.	180
4.21	Questionnaire d'évaluation du profil.	181
4.22	Interface de l'expérience cognitive : étape double tâche.	182
5.1	Visualisation 2D du diagramme formel multi-échelle obtenu automatiquement à partir du <i>Rondo K545</i> de W.A. Mozart.	194
5.2	Visualisation 3D en pavés du diagramme formel multi-échelle représenté via Blender d'un extrait du <i>Rondo K545</i> de W.A. Mozart.	200
5.3	Visualisation 3D en sphères du diagramme formel multi-échelle représenté via Blender d'un extrait du <i>Rondo K545</i> de W.A. Mozart.	201
5.4	Visualisation 3D en sphères du diagramme formel multi-échelle représenté via Blender d'un extrait du <i>Rondo K545</i> de W.A. Mozart. De haut en bas : le plan de la forme, le plan de la structure, le plan de l'organisation.	203

5.5	Visualisation 3D en sphères du plan de la structure du diagramme formel multi-échelle représenté via Blender d'un extrait du <i>Rondo K545</i> de W.A. Mozart (gauche) et de <i>l'Hommage à Rameau</i> de Debussy (droite).	204
5.6	Représentation de la mémoire et des hypothèses.	207
5.7	Interface de paramétrage.	208
5.8	Widgets. De haut en bas et gauche à droite : glisseur, bouton, entrée libre, sélecteur.	209
5.9	Fenêtre d'affichage de la vue 3D.	210
5.10	Diagrammes de classe de <i>sound_material</i> et <i>family</i> .	211
5.11	Extrait du fichier JSON <i>materials.json</i> envoyé au logiciel après traitement de MORFOS.	211
5.12	Affichage des plans 2D.	212
5.13	Options d'affichage.	213
5.14	Bouton de lecture.	213
5.15	Système d'interactions entre l'Interface et le MORFOS.	215
5.16	Bouton de sauvegarde et d'impression.	216
6.1	Diagramme formel de niveau 0 du <i>Geisslerlied</i> avec une fenêtre de taille 1024 et un seuil de 0.97.	223
6.2	Diagramme formel de niveau 1 du <i>Geisslerlied</i> avec une fenêtre de taille 1024, un seuil de similarité de 0.97 et un seuil de segmentation de 150.	224
6.3	Diagramme formel de niveau 0 du <i>Geisslerlied</i> avec une fenêtre de taille 1024 et un seuil de similarité de 0.97 avec algorithme de correction.	225
6.4	Diagramme formel de niveau 0 du <i>Geisslerlied</i> avec une fenêtre de taille 512 et un seuil de similarité de 0.97 sans algorithme de correction.	226
6.5	Diagramme formel de niveau 0 du <i>Geisslerlied</i> avec une fenêtre de taille 2048 et un seuil de similarité de 0.97 sans algorithme de correction.	226
6.6	Diagramme formel de niveau 0 du <i>Geisslerlied</i> avec une fenêtre de taille 1024 et un seuil de similarité de 0,94 sans algorithme de correction.	227
6.7	Diagramme formel de niveau 0 du <i>Geisslerlied</i> avec une fenêtre de taille 1024 et un seuil de similarité de 0,98 sans algorithme de correction.	227
6.8	Diagramme formel de niveau 0 du <i>Geisslerlied</i> joué deux fois : une première fois à volume élevé et une deuxième fois à volume faible avec une fenêtre de taille 1024 et un seuil de similarité de 0,97 et algorithme de correction.	228

6.9	Diagramme formel de niveau 0 du <i>Geisslerlied</i> synthétisé à la guitare avec une fenêtre de taille 1024 et un seuil de similarité de 0.97, sans algorithme de correction.	228
6.10	Diagramme formel de niveau 0 du <i>Geisslerlied</i> synthétisé à la guitare avec une fenêtre de taille 1024 et un seuil de similarité de 0.97, avec algorithme de correction.	229
6.11	Diagramme formel de niveau 0 du <i>Geisslerlied</i> synthétisé au piano puis à la guitare avec une fenêtre de taille 1024 et un seuil de similarité de 0.97, sans (haut) et avec (bas) algorithme de correction.	230
6.12	Diagramme formel de niveau 0 du <i>Geisslerlied</i> synthétisé au piano puis à la guitare avec calcul de 50 coefficients cepstraux.	231
6.13	Partition (haut) et Diagramme formel de niveau 0 du <i>Geisslerlied</i> joué en canon obtenue manuellement (gauche) et automatisée avec calcul de la cqt (droite).	232
6.14	Diagramme formel de niveau 0 du <i>Geisslerlied</i> joué en canon obtenue manuellement (gauche) et automatisée avec calcul de la cqt (milieu) et de la fft (droite) avec affichage de la polyphonie.	233
6.15	Spectre de la cqt (gauche) et Diagramme formel de niveau 0 (droite) du <i>Geisslerlied</i> associé.	234
6.16	Diagramme formel de niveau 0 du <i>Rondo K.545</i> de W.A. Mozart obtenu à partir d'une synthèse MIDI.	234
6.17	Diagramme formel de niveau 0 du <i>Rondo K.545</i> de W.A. Mozart obtenu à partir d'une performance musicale.	235
6.18	Diagramme Formel Multi-échelle du <i>Rondo K.545</i> de W.A. Mozart à partir d'une représentation symbolique.	236
6.19	Diagrammes formels de niveau 0, 1 et 2 du <i>Choral O Haupt voll Blunt Wunder</i> de la <i>Passion selon Saint Matthieu BWV.244</i> , J.S. Bach.	239
6.20	Diagrammes formels de niveau 0, 1 et 2 du <i>Concerto pour violon en la mineur RV356</i> de A. Vivaldi.	240
6.21	Diagramme formel multi-échelle du morceau RWC Pop 11.	245
6.22	Diagramme formel multi-échelle du morceau RWC Pop 21.	247
6.23	Page de présentation de l'interface prototypique.	249
6.24	Page de paramétrisation de l'interface prototypique.	250
6.25	Analyse paradigmatique du solo par F. Lerdhal, présentée par J.J. Nattiez dans [113]).	251
6.26	Comparaison des diagrammes de niveau 0 du solo de cor obtenus pour la version enregistrée (haut) et la version midi (bas).	252

6.27	Comparaison entre le diagramme formel de niveau 4 du solo de cor obtenu pour la version midi (seuil de similarité de 0,97) et le découpage formel proposé par Lerdahl.	253
6.28	Comparaison entre le diagramme formel de niveau 3 du solo de cor obtenu pour la version midi (seuil de similarité de 0,97) et le découpage formel proposé par Lerdahl. Les deux principales récurrences paradigmatiques identifiées par l’algorithme ont été encadrées en rouge et en bleu.	254
6.29	diagramme de niveau 3 obtenu pour la version MIDI (seuil de similarité de 0,965) et fragments musicaux correspondant aux différentes occurrences des trois principaux paradigmes.	255
6.30	Diagramme formel de <i>Musik to heare</i> de niveau 0 obtenu avec le logiciel.	255
6.31	Diagramme formel de <i>Musik to heare</i> de niveau 0 obtenu manuellement par E. Constable.	256
6.32	Diagramme formel de <i>Musik to heare</i> de niveau 1 obtenu avec le logiciel.	257
6.33	Diagramme formel de <i>Musik to heare</i> de niveau 1 obtenu manuellement par E. Constable.	257
6.34	Diagramme formel de <i>Musik to heare</i> de niveau 2 obtenu avec le logiciel.	258
6.35	Diagramme formel de <i>Musik to heare</i> de niveau 3 obtenu avec le logiciel.	258
6.36	Résumé de la structure obtenue avec le logiciel.	258
6.37	Diagrammes formels de l’ <i>Esperanza</i> obtenu avec le logiciel. De haut en bas : niveau 0, 3, 5 et 7.	259
A.1	Succession des états 1 à 11 des diagrammes formels équivalents aux représentations de niveaux 1 à 3.	270
C.1	Arbre des coûts théoriques de l’oracle.	278

Résumé étendu en français

Cette thèse présente la formalisation et le développement d'un logiciel d'analyse musicale appelé "Multiscale Oracle Representations For Organized Sounds" (MORFOS). Ce logiciel vise à mettre en œuvre un modèle multi-échelle de la forme musicale basé sur l'Algorithme Cognitif de Jean-Marc Chouvel. Les travaux de cette thèse s'inscrivent dans la continuité de l'analyse cognitive en musicologie, visant à comprendre les processus cognitifs qui interviennent lors de l'écoute musicale. Nous étudions une représentation hiérarchisée de la musique et explorons l'influence de cette hiérarchisation dans l'organisation des événements musicaux dans le temps et la compréhension de la musique. Nous formalisons ainsi les concepts de matériau, d'objet et de diagramme formel, et nous introduisons le Diagramme Formel Multi-échelle qui décrit la structure musicale à différentes échelles temporelles et niveaux d'analyse. Celui-ci est composé de trois plans que nous introduisons : la forme, la structure et l'organisation.

L'implémentation informatique de MORFOS a été réalisée en Python et accepte des représentations audio, symboliques ou vectorielles. Ce logiciel présente une architecture modulaire intégrant différents modules de traitement audio, de classification et de segmentation : nous présentons ainsi différentes mesures implémentées sous forme d'un ensemble de règles et discutons des contraintes associées à l'étude de la classification et de la segmentation à partir d'une représentation audio. Nous présentons ainsi la notion d'Agenda, qui correspond au choix par l'utilisateur d'un ensemble de règles permettant de représenter un modèle « d'écoute » pour l'analyse d'une œuvre musicale par le logiciel.

La thèse explore également la question de la complexité de la structure musicale : nous proposons l'expression d'un coût associé à la description de l'objet musical acquis en fonction de son contexte, selon la définition de Kolmogorov. Nous cherchons également à comparer le comportement du logiciel MORFOS avec les phénomènes d'attention et la charge cognitive lors de l'écoute musicale. Une expérience visant à mesurer la charge cognitive pendant la tâche de segmentation

musicale a ainsi été conçue.

Cette thèse présente par ailleurs des réflexions sur la visualisation des diagrammes formels multi-échelles. A cette occasion, nous avons développé une interface permettant de rendre le logiciel accessible à tous les utilisateurs. Enfin, des exemples d'analyses musicales réalisées avec MORFOS sont présentées, sur une base de données musicales pop ainsi qu'un corpus d'œuvres classiques.

Introduction

Ce manuscrit est le résultat d'une thèse interdisciplinaire en Musicologie à Sorbonne-Université au sein de l'Institut de Recherche en Musicologie (IReMus) en co-direction informatique avec le Laboratoire Bordelais de Recherche en Informatique (LaBRI).

Nous présentons un modèle multi-échelle de la forme musicale à travers l'implémentation d'un logiciel que nous appelons *Multiscale Oracle Representations For Organized Sounds* (MORFOS), un oracle multi-échelle permettant de décrire la représentation formelle des œuvres musicales selon un modèle cognitif. Il s'agit d'une automatisation des travaux de Jean-Marc Chouvel sur la théorie de l'algorithmie cognitive [33] qui permet de produire une représentation formelle des œuvres musicales analysées avec ce que l'on appelle des *diagrammes formels*. On espère que l'étude de ce modèle permette de mieux comprendre les phénomènes cognitifs qui ont lieu lors de l'écoute musicale, notamment par l'étude et la représentation des relations internes des éléments constitutifs de l'œuvre à travers le temps mais également à travers les niveaux d'analyse.

Contexte

Les travaux de cette thèse s'inscrivent dans la continuité de l'analyse cognitive en musicologie. La première idée à considérer est que l'analyse de la musique est essentielle pour la compréhension de celle-ci. La musique n'est pas une entité figée dans le temps, mais plutôt un processus cognitif en constante évolution. En d'autres termes, analyser la musique pourrait être défini - entre autres - comme revenir à comprendre les processus cognitifs qui sous-tendent celle-ci.

Une question fondamentale soulevée par Jean-Marc Chouvel dans son ouvrage intitulé *Analyse Musicale : Sémiologie et cognition des formes temporelles* [33] est la définition même de la musique. Une première définition courante considère

la musique comme *"l'art d'agencer les sons de manière agréable à l'oreille"*, se concentrant ainsi uniquement sur l'aspect esthétique. Cette définition néglige l'aspect subjectif de l'auditeur et de l'interprète. Une définition plus tardive proposée par Luciano Berio élargit la notion de musique en la définissant comme *"tout ce que l'on écoute avec l'intention d'écouter de la musique"* [9]. Cette définition introduit l'élément d'intention de l'auditeur et la dimension cognitive de l'écoute musicale. La musique n'est donc pas qu'un simple flux sonore, elle implique également la manière dont ce flux est perçu et interprété par le système cognitif de l'auditeur. John Cage, par exemple, étend la musique à tous les sons. Ainsi, il est difficile de définir la musique de manière intrinsèque, mais ce qui peut être défini, ce sont les effets qu'elle engendre. Selon Kandinsky, la musique, tout comme la peinture, vise à élever l'âme humaine. La musique peut susciter des émotions, mais ces émotions sont-elles pleinement ressenties sans avoir compris la pensée musicale ? A. Boucourechliev [21] définit quant à lui la musique comme *"un système de différences qui structure le temps sous la catégorie du sonore"* : ainsi, la musique devient l'organisation des différences entre les phénomènes sonores permettant par leur étude de déterminer la structure sous-jacente à ceux-ci.

L'analyse musicale est ainsi essentielle pour comprendre la musique. Comme le souligne N. Meeùs [101], Molino et Nattiez définissent l'analyse comme *"un métalangage [...] elle exprime le sens de la musique"*. Étant donné que la musique reflète le système cognitif, l'analyse pourrait également permettre de mieux comprendre ce système. Cependant, il existe une approche d'analyse dite "neutre", présentée par Nattiez, qui se limite à l'aspect "immanent" du musical sans prendre en compte d'autres éléments, et qui ne sont donc orientées ni du côté de la création, ni du côté de la perception de la pièce musicale. Cette approche vise à définir la spécificité de l'élément musical lui-même.

On trouve dans la linguistique de Ferdinand de Saussure la distinction entre la synchronie, où les éléments coexistent simultanément, et la diachronie, où les éléments sont confrontés à leur évolution dans le temps. Les défenseurs du structuralisme soutiennent que l'entité est définie par les relations entre ses éléments plutôt que par les éléments eux-mêmes. Cela conduit à la distinction entre les rapports syntagmatiques (relations entre les éléments dans une séquence) et les rapports paradigmatiques (relations basées sur la similarité). Otto Laske a introduit la notion de "musicologie cognitive" pour comprendre comment l'exécution musicale affecte le système cognitif. Il la définit ainsi : *"La musicologie cognitive est le nom de la discipline que j'ai créée dont l'objectif est de comprendre en profondeur les relations entre les processus mentaux musicaux tels que la composition, l'analyse musicale, la composition musicale et la performance musicale - avec le*

résultat que ces processus produisent"¹. Ainsi, la musicologie cognitive cherche à explorer les relations entre les processus mentaux liés à la musique (composition, analyse musicale, interprétation, etc.) et l'expérience musicale. Elle vise à comprendre comment les processus cognitifs induits par les structures influencent la musique.

Ce que soulève la musicologie cognitive est également la question de la hiérarchisation. En effet, E. Bigand dans [12] souligne que si l'on cherche à comprendre la relation entre les éléments traités dans la pièce musicale, l'hypothèse dominante est celle de H. Schenker, qui suggère que l'organisation musicale s'effectue en fonction d'une hiérarchie dépendante de l'importance des événements musicaux. Jean-Marc Chouvel quand à lui dans [33] présente sa théorie de la *tripartition espace-modèle-objet* présenté en figure 1. L'objet est ainsi défini comme une instance, "*l'énoncé d'un événement de début, qui comprend diverses précisions [...] et d'un événement de fin*". L'espace est assimilé au dictionnaire qui contient les objets et le modèle correspond à la description du groupement des objets dans des entités plus importantes. Ainsi, ce principe peut être appliqué à un espace "plus grand", de niveau supérieur, qui produira un nouvel objet de niveau encore supérieur par imbrication successive. Pour illustrer la construction d'une hiérarchisation lors de l'écoute musicale, Jean-Marc Chouvel émet ainsi un parallèle entre l'organisation spatiale des objets à travers leur modèles (par exemple l'objet de l'atome dans son espace, regroupé spatialement avec d'autres atomes donne le modèle molécule, qui est lui même un objet dans l'espace des molécules, et ainsi de suite pour obtenir l'objet *individu*), et sa transposition au phénomène temporel (l'objet de l'échantillon de signal dans son espace, regroupé temporellement à d'autres échantillons donne le modèle note, qui est lui même un objet dans l'espace des notes, ainsi de suite pour obtenir l'objet *œuvre musicale audio*).

L'algorithme cognitif s'inscrit ainsi dans la lignée du structuralisme et de la musicologie cognitive en proposant un algorithme d'analyse musicale qui permet la production en temps réel, c'est-à-dire au fur et à mesure de l'acquisition du signal lors de l'analyse, de diagrammes formels qui sont une représentation de la *pperception* du schéma formel musical. L'algorithme permet de construire une hiérarchie de diagrammes formels de différentes échelles musicales, les diagrammes d'échelles temporelles plus grandes étant construits à partir des informations contenues dans les diagrammes d'échelles inférieures.

1. "*Cognitive musicologie is the name I gave to the discipline of my invention whose purpose it was to understand in depth the relationship between mental processes called "musical" -such as "composing", "analysing music", "conducting music", "performing music" etc. - and the "musical" result they produce*", <http://www.ottolaske.com/musicology.html>

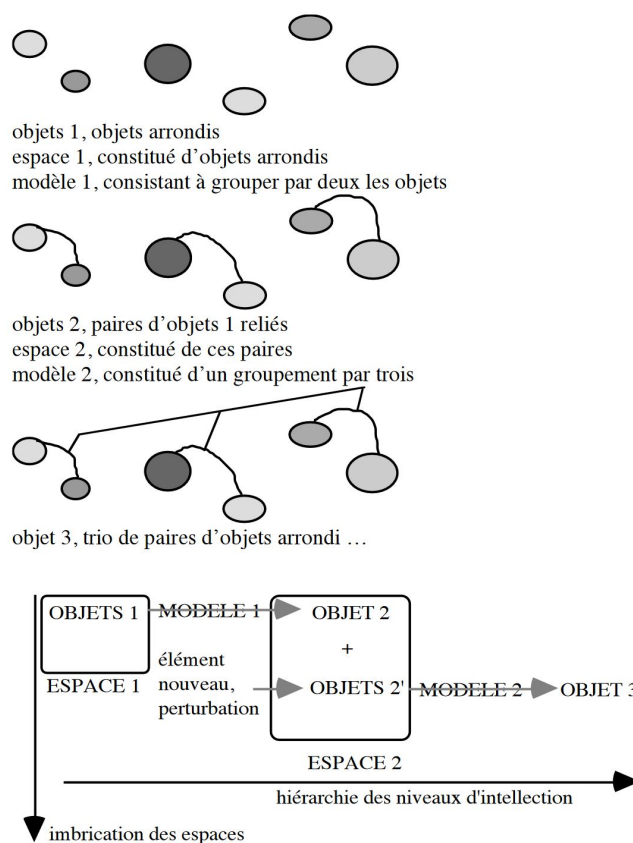


FIGURE 1 – Représentation schématique de la constitution structurée des objets d’un univers, en parallèle avec un schéma de la tripartition conceptuelle de la forme en espace, modèle et objet, extrait de [33].

Objectif

L’objectif de cette thèse est ainsi d’automatiser le modèle de l’algorithme cognitif avec la création du logiciel MORFOS. On cherche un modèle permettant de structurer et de proposer une analyse paradigmatique automatisée dans la continuité des théories linguistiques de Ferdinand de Saussure, c’est à dire selon l’étude des éléments constitutifs de l’œuvre et des relations entre ceux-ci. Une question fondamentale pour cette thèse est donc de proposer un modèle hiérarchisé à travers le développement de ce logiciel qui nous permette de mieux comprendre les processus qui mettent en lumière la hiérarchisation lors de l’écoute musicale, et d’analyser l’apport de cette hiérarchisation dans la compréhension de la musique et son écoute. Les exemples proposés par J.M. Chauvel concernent principalement

des partitions. L'analyse automatisée à partir de l'audio soulève des difficultés spécifiques liées à l'implémentation automatique de la notion de similarité.

Même si on souhaite automatiser cette modélisation à partir de fichiers audio, on proposera également que celle-ci puisse être applicable à partir de représentations symboliques. On souhaite par ailleurs que la représentation obtenue soit dépendante du déroulement temporel de l'œuvre, et donc que l'analyse à un instant t donné ne dépende que des éléments de l'œuvre qui précèdent.

Cette modélisation a lieu sur plusieurs niveaux d'analyse correspondant à différentes échelles temporelles pour la construction de la forme associée à l'œuvre écoutée, les niveaux étant déterminés par le résultat de la procédure elle-même et non pas à priori. L'objectif est donc de proposer une représentation hiérarchisée des diagrammes formels, et qu'ils soient tous construits en temps réel lors du processus d'analyse et d'acquisition de l'information.

Une hypothèse étant par ailleurs que le système soit suffisamment général pour décrire tous les types d'écoute sans être dépendant d'un profil spécifique d'auditeur. En effet, selon F. Delalande [42], chaque auditeur peut donner une analyse différente d'une même œuvre, on propose donc que le modèle soit paramétrable afin de laisser le choix à l'utilisateur des critères de discrimination pour les éléments apparents sur le diagramme formel.

Par ailleurs, on souhaite proposer un modèle qui ne dépende pas du genre musical analysé. On cherchera donc à déterminer un ensemble de règles de segmentation et de classification des éléments de l'œuvre qui se veulent les plus générales possibles.

Ainsi, l'automatisation d'un tel processus pourrait aider les musicologues de différentes manières : tout d'abord, si l'obtention d'un ensemble de diagrammes formels hiérarchisés peut prendre plusieurs heures à être produit à la main, cela peut être un gain de temps pour le musicologue de transmettre cette étape de la production à une machine. Si l'obtention du diagramme est automatisée, un expert musicologue est toujours nécessaire pour l'analyse du diagramme. Au-delà du temps pour une unique œuvre donnée, cela signifie qu'il devient possible d'étudier un corpus imposant, voir d'automatiser également les différents calculs et mesures permettant de comparer les résultats des œuvres d'un corpus donné. Une autre raison est que les calculs de similarité et de segmentation opérés par l'algorithme sont totalement indépendants des connaissances du musicologue qui va analyser l'œuvre. Cela signifie que la représentation obtenue, contrairement à une représentation manuelle, est exempte des biais du musicologue. En revanche, de nouveaux biais sur les choix d'implémentation sont certainement introduits par le dévelop-

pement.

Par ailleurs, on pourra chercher à analyser non pas seulement les diagrammes formels, mais le comportement de l'algorithme ayant permis de les produire. Cela pourrait mettre en avant les mécanismes de segmentation et de classification à travers notre modèle et ainsi espérer s'approcher d'une meilleure compréhension des mécanismes de l'écoute musicale.

Périmètre de l'étude

Nous avons émis des hypothèses fortes concernant la modélisation de la musique pour le logiciel dans un premier temps. Tout d'abord, nous ne souhaitons pas imposer de contraintes structurelles liées au genre musical. C'est pourquoi nous proposons une analyse à partir de fichier audio ainsi qu'un système de paramètres : c'est le traitement du fichier audio ainsi que l'ensemble des règles de similarité et de segmentation qui pourront être choisis en fonction du genre musical. Par exemple, une analyse timbrale pourra être plus adaptée pour de la musique spectrale, tandis qu'une analyse harmonique pourra l'être pour de la musique classique. Un certain nombre de traitements sont proposés et présentés dans cette thèse. Bien que l'objectif à long terme est de pouvoir proposer des traitements extérieurs à MORFOS en fonction des besoins de l'utilisateur, le logiciel est pour le moment limité aux mesures implémentées. Par ailleurs, le rythme, qui est une composante primordiale de l'analyse musicale, n'est pas encore pris en compte dans le logiciel. Les diagrammes formels obtenus sont alors actuellement dépendants d'un ensemble de critères définis par l'utilisateur qui appartiennent à la même dimension musicale. Ainsi, un diagramme formel sera obtenu en fonction du timbre, ou de l'harmonie, mais pas des deux à la fois. Il pourra alors être pertinent pour comprendre une œuvre dans son intégralité de mettre en vis-à-vis les différents diagrammes obtenus selon chacune des dimensions analysées.

Le modèle implémenté par ailleurs nous a rapidement permis de prendre en entrées des représentations symboliques de la musique, c'est pourquoi nous avons étendu le modèle à de telles représentations. Alors, la dimension de l'analyse prise en compte dépend de la représentation symbolique elle-même.

L'hypothèse sans doute la plus contraignante que nous avons émise dans notre recherche est la linéarité de l'œuvre musicale. En effet, nous supposons dans toute cette étude que la musique est construite comme une succession linéaire d'éléments qui ne se superposent pas temporellement. En effet, le problème de la séparation des voix est un travail à part entière que nous avons laissé de côté pour nous

concentrer sur le problème de la modélisation d'un modèle hiérarchisé. Cela nous contraint à laisser de côté le traitement polyphonique au sens contrapuntique.

L'implémentation par ailleurs de l'algorithme a été effectuée en Python pour la simplicité algorithmique et la flexibilité d'implémentation qu'offre ce langage vis-à-vis des perspectives d'évolution de l'implémentation tout au long de la thèse, au dépend d'une performance optimale. Ainsi, on cherche à obtenir des diagrammes formels en construisant en simultané des processus d'analyse qui sont effectués en fonction de l'ordre d'occurrence des éléments de l'œuvre musicale, mais l'analyse même ne sera pas effectuée totalement en temps réel au sens où les performances ne permettent pas nécessairement d'obtenir des résultats en un temps inférieur à la durée de l'œuvre analysée. Pour un affichage en simultané de l'écoute de l'œuvre, une solution proposée est donc de faire le traitement en amont et de ré-afficher dans un second temps en simultané de l'œuvre l'évolution des résultats obtenus.

Contributions

La contribution principale de cette thèse est d'une part la présentation d'un modèle informatique de la théorie de l'Algorithme Cognitif de Jean-Marc Chouvel, ainsi que l'implémentation en Python 3 de ce modèle. Ainsi, on propose un modèle avec une architecture modulaire divisée en un module algorithmique, un module pour la modélisation des structures de donnée, un module pour la visualisation de celle-ci ainsi que trois modules dont l'objectif est de pouvoir être augmentés par des traitements éventuels externes que sont un module de pré-traitement des données audio, ainsi que les deux modules de calcul de similarité et de segmentation. On propose ainsi un algorithme récursif qui prend en entrée un fichier audio ou une représentation symbolique vectorielle ou sous forme de séquence de caractères. Cet algorithme permet l'analyse successive de la classification et de la segmentation de chacun des éléments qui constituent l'œuvre et fournit une représentation hiérarchisée d'un ensemble de diagrammes formels, que l'on appelle des *diagrammes formels hiérarchisés*. Ces diagrammes sont calculés à partir d'une structure de données que l'on appelle l'*Oracle Multi-échelle* qui permet de représenter les liens entre les éléments qui constituent l'œuvre et leur relation dans le temps, mais également en fonction des niveaux. Cette structure de donnée est également un modèle de Markov, qui permet de modéliser les phénomènes d'attention, d'anticipation et de surprise *structurés*.

Par ailleurs, une interface, bien qu'incomplète, est implémentée en Electron. Elle permet ainsi de rendre l'algorithme cognitif et MORFOS accessibles à des

utilisateurs non familiers des programmes informatiques et se veut la plus facile à utiliser possible. L'implémentation de l'interface a été effectuée avec des étudiants en informatique de l'ENSEIRB-Matmeca², tandis que le travail sur l'interface a été effectuée avec un étudiant en ergonomie de l'Université de Toulouse Jean-Jaurès³. Nous avons ainsi pu présenter avec cet étudiant trois visualisations possibles pour les diagrammes formels, l'objectif étant de mettre en avant de la manière la plus adéquate possible les relations hiérarchiques entre les éléments des différents niveaux. En plus de la représentation des diagrammes formels habituels en deux dimensions, nous proposons donc également deux représentations en trois dimensions où les niveaux de hiérarchie sont superposés. Ces représentations en trois dimensions ont permis d'explicitier deux plans supplémentaires à celui des diagrammes formels (*plan de la forme*) qui sont le *plan de la structure* et le *plan de l'organisation*. Ces deux plans ouvrent ainsi le champ vers d'autres perspectives d'analyses et de représentations musicales et de traitement des données de l'œuvre.

Nous avons également ouvert des pistes concernant l'analyse du comportement de MORFOS lui-même. Une première piste est l'analyse de sa complexité. En effet, l'hypothèse est que la complexité permet de décrire la structure de l'œuvre considérée. On propose donc un ensemble de critères et un ensemble de coûts basés sur ces critères afin de déterminer une structure optimale, c'est-à-dire qui minimiserait la complexité obtenue. Une mise en perspective de l'implémentation de l'algorithme avec l'analyse de second ordre présentée par Jean-Marc Chouvel avec la notion de phase cognitive est également abordée. La deuxième piste correspond à la comparaison du comportement du logiciel avec les mécanismes cognitifs réels. Pour cela, nous avons mis en place une expérience double tâche avec une étudiante en science cognitive de l'université de Bordeaux pour l'élaboration du protocole et une étudiante en informatique de l'ENSEIRB-Matmeca⁴ pour le développement de l'interface, dans l'objectif de décrire la charge cognitive ayant lieu lors d'une tâche de segmentation pendant une écoute active de la musique. Les résultats de cette expérience restent à être traités et approfondis.

Plan

La thèse est constituée de 6 chapitres. Le chapitre 1 décrit un état de l'art musicologique et informatique. Nous présentons tout d'abord un paysage général de l'analyse musicale et des différentes méthodes d'analyse, qui dépendent des types de

2. Raphael Blard, Louis Dujardin, Léna Héreau, Louis Jouhault, Gael Perron et Nicolas Sanson

3. Erwan Michel

4. Anatazja Adamczyk et Asia Auville

média étudiés. Nous nous concentrons sur l'analyse de la forme musicale, que nous distinguons de la notion de structure, puis sur l'analyse sémiologique et son rapport à la forme. Nous abordons ensuite les problématiques associées à la modélisation de la cognition, et la psychologie de l'écoute, qui débouche sur l'émergence d'une musicologie cognitive. C'est dans ce contexte que nous présentons une première fois brièvement l'Algorithme Cognitif de J.M. Chouvel, qui est la théorie fondamentale à cette thèse. Enfin, nous abordons la question de la modélisation informatique des problèmes musicaux, et de la musicologie computationnelle. Nous décrivons dans un premier temps les problématiques du traitement du signal, avant de présenter un certain nombre de logiciels d'analyse musicale existants, afin d'explicitier la nécessité d'implémenter notre propre logiciel.

Le chapitre 2 présente la formalisation de MORFOS. Dans un premier temps, nous présentons de manière plus exhaustive le principe des diagrammes formels avant de présenter une formalisation statique et dynamique à un niveau d'analyse, puis sur l'ensemble des niveaux d'analyse en introduisant ce que nous appelons le diagramme formel multi-échelle. Deux exemples sont introduits dans ce chapitre et le rythmeront : il s'agit du *Geisslerlied Maria muoter reinû maît* et du *Rondo du Menuet K.545* de Wolfgang Amadeus Mozart. Cette représentation hiérarchisée du diagramme formel multi-échelle permet de décrire l'information en 3 dimensions sur les trois axes permettant de décrire l'information selon trois projections correspondant aux trois plans qui en découlent et qui sont détaillés dans une section suivante. Enfin, la notion d'*Agenda* est introduite et permet de soulever la question des écoutes multiples selon les individus, de la classification, de la segmentation et de la multi-dimensionalité des diagrammes formels pouvant être obtenus à partir d'une unique source audio.

Le chapitre 3 décrit l'implémentation informatique de MORFOS. Dans un premier temps, l'architecture du logiciel est décrite. Celle-ci est modulaire, de telle sorte que certains modules tels que ceux des règles de classification et de segmentation visent à pouvoir être complétés ou remplacés par des modules de calcul extérieurs. Ensuite, la structure de données de l'oracle multi-échelle est présentée et permet de modéliser les objets dans les différents niveaux et leurs relations selon les trois plans, ainsi que les phénomènes d'attention. Nous présentons également l'algorithme lui-même et plus particulièrement les fonctions de calcul de classification et de segmentation, ainsi que la gestion des règles associées qui sont implémentées dans MORFOS. L'ensemble des règles implémentées sont ensuite décrites, avant d'explicitier l'interface de paramétrisation du logiciel et ses contraintes, qui permettent de spécifier l'Agenda.

Le chapitre 4 réintroduit l'Algorithme Cognitif et MORFOS sous la perspective

de modèles d'écoutes. Après avoir abordé la relation entre complexité, phénomènes d'attention et charge cognitive, nous effectuons un bref état de l'art des ces trois notions. Une étude de l'explication de la structure sous le prisme de la complexité est proposée, avant d'aborder les phénomènes d'attention avec une modélisation des phases cognitives présentées par J.M. Chauvel dans [32]. Nous présentons ensuite le protocole d'une expérience pour la mesure de la charge cognitive lors de la tâche de la segmentation lors d'une écoute active sur un répertoire de musique classique.

Le chapitre 5 présente une réflexion sur la visualisation du diagramme formel multi-échelle, ainsi que l'implémentation d'une interface utilisateur qui intègre une des visualisations proposées. Tout d'abord, un état de l'art des représentations et visualisations de la musique et des structures musicales est effectué, puis trois visualisations sont proposées. Une première visualisation présentée est la visualisation en deux dimensions du diagramme formel multi-échelle avec les diagrammes formels des différents niveaux superposés tandis que les deux autres visualisations sont des visualisations en trois dimensions, permettant de mieux expliciter les trois plans détaillés dans le chapitre 2. Une proposition de la visualisation des phénomènes d'attention est également présentée, tandis que la section suivante présente l'implémentation en Electron de l'interface utilisateur.

Le chapitre 6, enfin, présente des cas concrets d'analyse musicale sur des diagrammes formels produits par MORFOS. Nous présentons tout d'abord une analyse sur une grande base de données qui est RWC Pop. Dans un second temps, des analyses plus spécifiques sont présentées, sur un corpus d'œuvres classiques.

Chapitre 1

Etat de l'Art

Le travail traité dans le cadre de cette thèse concerne le développement d'un logiciel automatisé d'analyse musicale s'inscrivant dans la continuité de l'analyse sémiologique et de la musicologie cognitive. Nous nous intéressons plus particulièrement à la représentation d'un modèle de la cognition lors de l'écoute musicale qui est multi-échelle : on cherche ainsi par l'automatisation d'un tel modèle à comprendre les phénomènes de l'écoute musicale et plus particulièrement les phénomènes de classification et de segmentation au cours du temps. On cherche ainsi à analyser l'apport de la hiérarchisation pour ces phénomènes et les rapports qui émergent entre les éléments étudiés aux différentes échelles de temps. Nous présenterons donc le contexte dans lequel nous nous situons d'un point de vue de l'analyse musicale et du point de vue de la modélisation de la cognition en musicologie, et nous effectuerons une revue des procédés de l'analyse automatisée et des logiciels existants. Nous présenterons un état de l'Art spécifique aux problématiques des modèles cognitifs et de la visualisation dans les chapitres concernés (respectivement chapitres 4 et 5).

1.1 Analyse musicologique de la forme

Nous cherchons à automatiser l'Algorithme Cognitif présenté par Jean-Marc Chauvel [33]. L'objectif d'une telle démarche est d'offrir aux musicologues une nouvelle grille de lecture des œuvres musicales. Ainsi cette algorithme permet de produire une représentation permettant au musicologue d'effectuer une analyse formelle de celle-ci. Nous présentons dans un premier temps les termes généraux de l'analyse musicale, avant de spécifier la définition et l'intérêt de l'analyse formelle

et sémiotique.

1.1.1 Paysage de l'analyse musicale

L'analyse musicale consiste à étudier une œuvre musicale en décortiquant les éléments intrinsèques et extrinsèques à celle-ci tels que les motifs, le rythme, mais également les éléments contextuels, afin de mieux comprendre entre autres la structure de l'œuvre, le contexte historique, les techniques compositionnelles et leurs effets sur l'auditeur. Ainsi, Ian Bent présente l'analyse musicale dans *L'analyse musicale, histoire et méthodes* [8] comme une discipline émergente dans les années 1950 qu'il définit comme étant "*la résolution d'une structure musicale en éléments constitutifs relativement plus simples, et la recherche des fonctions de ces éléments à l'intérieur de cette structure*". L'analyse musicale peut être définie selon deux courants : une analyse descriptive qui considère la musique comme un objet abstrait, par opposition à l'étude de l'aspect sémiologique et phénoménologique, où la musique est considérée en tant que phénomène cognitif à travers un processus temporel.

On peut se concentrer sur différents aspects de l'œuvre afin de procéder à l'analyse de celle-ci. Parmi ces divers aspects, l'analyse formelle permet de comprendre l'organisation des différentes sections d'une œuvre musicale et la relation entre ces sections. On peut par ailleurs effectuer une analyse harmonique qui permet d'étudier les relations entre les accords d'une œuvre musicale donnée ou encore une analyse motivique afin d'étudier les éléments mélodiques et rythmiques récurrents de l'œuvre musicale. Charles Rosen [133] présente à travers ses analyses comment l'harmonie et le rythme, entre autres, permettent d'organiser la structure, et comment celle-ci aide à mieux comprendre le contenu de l'œuvre dans le cadre de la musique classique. Heinrich Schenker [142] propose une théorie de l'analyse musicale permettant de hiérarchiser les notes de l'œuvre musicale et de réduire celle-ci à une structure tonale fondamentale appelée *Ursatz*. Une analyse sémiologique, telle que présentée par Jean-Jacques Nattiez [112], consiste à étudier la signification associée aux signes correspondants à des événements musicaux de l'œuvre tandis que Léonard Meyer [103] propose entre autres de se focaliser sur les aspects esthétiques de l'œuvre et les effets qu'ils produisent chez l'auditeur en terme d'émotions.

Ces différentes analyses sont complémentaires et apportent des informations sur différentes dimensions de l'œuvre musicale. Ainsi, notre approche se concentre sur l'analyse paradigmatique et sémiologique de l'œuvre à travers une représentation formelle de celle-ci. On s'inscrit ainsi dans la veine du structuralisme, où

l'on cherche à mettre en évidence la structure de l'œuvre musicale en relevant les éléments constitutifs de l'œuvre et la manière dont ceux-ci interagissent entre eux. L'analyse paradigmatique peut être effectuée à partir de différentes dimensions de l'œuvre telles que l'aspect harmonique ou timbral, et on propose une analyse à différentes échelles temporelles de celle-ci, permettant ainsi une réduction ou compression de l'information à la manière de la théorie proposée par Schenker. Cette analyse sémiologique est intimement liée aux effets sur l'auditeur et à la dimension esthétique, que nous traiterons par l'analyse du comportement de notre algorithme.

Pour parler d'analyse, il est nécessaire de parler de l'objet de l'analyse : la musique. Celle-ci se présente sur différents médias : écrits (partitions, visualisations diverses), ou audio (performances, enregistrements). Alors qu'une analyse sur une représentation écrite permet d'effectuer une analyse indépendante du déroulement temporel de l'œuvre, un fichier audio perçu en temps réel ne permet pas de connaître les événements constitutifs de l'œuvre à l'avance et d'effectuer une analyse hors-temps. Cependant, une fois l'écoute effectuée, et lors d'une écoute suivante, l'information connue de l'analyste est globale et peut être restructurée. Comme le souligne Jean-Marc Chouvel [33], la musique est par essence temporelle, et si l'analyse musicale peut exister hors du temps, procéder à une analyse en temps réel pourrait nous aider à mieux comprendre le phénomène musical.

L'émergence de la Théorie de l'information de Claude Shannon [146] et de l'emploi de l'informatique pour le traitement du signal permettent d'une part le développement de nombreux outils informatiques pour l'analyse à partir de l'audio : nous décrirons ces outils dans une partie ultérieure. D'autre part, cela permet de théoriser la communication en explicitant les notions d'*émetteur*, de *récepteur*, de *canal*, de *code* et de *message*, ce qui permet de voir la musique comme un phénomène temporel quantifiable, émis et perçu dans son contexte.

Par ailleurs, le type de média peut inférer sur les processus de traitement opérés pour l'analyse sur l'objet musical. Par exemple, Nathalie Herold [66] propose une analyse timbrale, justifiée par l'exhaustivité de l'information timbrale lors de la perception musicale, étroitement liée à l'objet musical sonore.

La musique étant intimement liée à sa dimension temporelle et au phénomène perceptif associé, il y a tout intérêt à traiter la musique à partir du son plutôt qu'à partir de la partition, ou du moins à combiner les deux analyses. A partir du XXe siècle, de nombreux compositeurs comme John Cage décident d'étendre la musique à l'ensemble du sonore [158]. Dans les années 1940, Pierre Schaeffer et Pierre Henry développent la musique concrète, dont la démarche consiste à travailler directement les sons, enregistrés, modifiés, pour composer la musique. Il

devient alors difficile de proposer une représentation de telles musiques sous forme de partitions, et de proposer à partir de partitions des analyses de ces musiques. Suite à la création en 1958 du Groupe de Recherche Musicales (GRM), Pierre Schaeffer publie en 1966 le *Traité des Objets Musicaux* [141], dans lequel est défini la notion *d'objet sonore*. Dans le Dictionnaire de la musique [159], la définition est reprise en ces termes : "[L'objet sonore] désigne un phénomène sonore perçu dans le temps comme un tout, une unité, quels que soient ses causes, son sens, et le domaine auquel il appartient (musical ou non) [...] la notion d'objet sonore généralise la notion de note (en tant qu'unité combinatoire) à tout l'univers sonore, en incluant les sons qui n'ont pas de hauteur définie, où dont le profil, les caractères, ne rentrent pas dans les critères habituels de la musique traditionnelle. La note de musique devient alors un cas particulier d'objet sonore." Ainsi, si l'objet sonore est employé dans un contexte musical, il permet d'étendre l'objet musical à l'ensemble des dimensions sonores de celui-ci, en gardant centrale la notion de perception de celui-ci par un sujet. Le Laboratoire Musique et Informatique de Marseille (MIM) présente les Unités Sémiotiques Temporelles (UST) [43]. Cela répond à un besoin de représenter le matériau sonore au-delà de ses composantes de hauteur et de durée, mais également de considérer le sens sémiotique de celui-ci. Ainsi, le MIM décrit dix-neuf UST chacune accompagnées de définitions morphologiques et sémantiques. Les UST ont la particularité que le sens est intégré dans la représentation : ces descriptions du sonore permettent de nouvelles analyses sans partition. Ainsi le MIM présente¹ des Analyses en UST tel que *Misere* d'Arvo Part, ou encore *Sequenza III* de Luciano Berio.

On choisit ainsi de se concentrer dans le cadre précis de cette thèse sur l'objet musical sonore et son analyse en temps réel, c'est-à-dire au fur et à mesure de l'acquisition de ses éléments constitutifs dans le temps.

1.1.2 Analyse de la forme

Dans le cadre de nos recherches, on distingue en réalité la notion de structure de la notion de la forme. Les notions de formes et de structures et leurs étymologies sont largement discutées et découlent d'un historique que nous ne présentons pas largement dans le cadre de cette thèse : cette question est discutée par J.M. Chouvel [31], voici cependant le contexte dans lequel nous nous situons. Les prémices d'une analyse formelle sont présentées dans [130]. A. Riotte présente trois niveaux d'analyse chez J. Barraqué : le premier niveau est la caractérisation des éléments primaires en fonction de leur ordre séquentiel : autrement dit, la prise en

1. <https://www.labo-mim.org/site/index.php?2008/08/11/24-les-ust>

compte de la chronologie de l'œuvre musicale dans l'analyse. Le deuxième niveau consiste à remplacer les éléments primaires par des principes événementiels (*event principles*) en préservant l'ordre temporel. Enfin, J. Barraqué effectue son processus analytique sur les principes événementiels. Il travaille alors sur des cellules *cells*, définies comme les plus petites unités ayant un sens et dont l'aspect chronologique est conservé, ce qui permet d'aboutir à une description morphologique de l'analyse de l'œuvre musicale. J.M. Chouvel [31] définit ainsi la forme comme intrinsèquement liée à la notion de mémoire et d'attention. La notion de forme tire ses concepts de la *Gestalttheory*, soit *Théorie de la forme*, qui met en exergue des règles de formations qui nous permettent de percevoir une entité non seulement comme la somme de ses éléments, mais également grâce à la relation entre ces éléments, qui constitue le tout.

La différence entre forme et structure est définie par B. De Schlœzer [40] de la manière suivante : "la structure est l'agencement de diverses parties en vue de constituer un tout, tandis que la forme est précisément ce tout en tant que tel, considéré dans son unité". H. Schenker [142] définit également la structure musicale comme "l'organisation hiérarchique des sons dans une œuvre musicale". La notion de hiérarchie est en effet fondamentale dans la définition de la structure. On peut ainsi définir différents niveaux de structure comme les niveaux contenant des éléments de différentes échelles temporelles. Par exemple, une œuvre musicale pourra révéler un niveau de structure à l'échelle de la note, mais également un niveau de structure à l'échelle du motif, ou de la phrase. De par leurs rapports dans la construction des représentations formelles que l'on proposera dans la suite de cette thèse, on parlera alors de niveaux structurels hiérarchisés.

1.1.3 Analyse sémiologique

Ainsi, la notion de temporalité dans la représentation formelle est fondamentale à notre analyse. Par ailleurs, comme les sémiologues, nous cherchons à comprendre d'une part ce qui fait entité, autrement dit comment segmente-t-on, et d'autre part ce qui fait identité, à savoir comment compare-t-on. Nous nous inscrivons ainsi dans la lignée des théories linguistiques de Ferdinand de Saussure [140]. Il développe les notions duales de paradigme et de syntagme, qui correspondent respectivement aux éléments répétés, identités dans l'œuvre, et à la répartition temporelle de ces éléments. Seulement, la musique se différencie de la linguistique en cela qu'elle n'a pas de mots pour la construire et est propice à une variété de styles innombrables. Nicolas Ruwet, dans [134] propose une des premières analyses paradigmatiques sur un *Geisslerlied*, un chant allemand populaire de flagellation monophonique, datant de 1349. Ainsi, il regroupe verticalement sur la partition les éléments ap-

pelés "séquences équivalentes", distinguant la notion de paradigme dans l'œuvre musicale.

J.-J. Nattiez, à la suite de J. Molino [105][111] présente la notion de tripartition, appliquant à l'analyse musicale les processus sémiologiques poïétiques, c'est-à-dire de création, neutre (ou immanent) qui est le fait en tant que tel, et esthétique, c'est-à-dire de réception. Il différencie également les analyses selon qu'elles sont internes, c'est-à-dire qu'elles proviennent du contenu de l'œuvre au niveau neutre, ou bien qu'elles sont externes, c'est-à-dire qu'elles sont basées sur l'analyse de documents et de processus extérieurs. Il présente ainsi six catégories pour l'analyse. L'analyse de niveau neutre est factuelle et s'attache à une analyse à partir du matériau même. L'analyse poïétique inductive qui consiste à déterminer le discours musical et les intentions du compositeur à partir de la partition. L'analyse poïétique externe se base sur les documents laissés par le compositeur, tel que les notes et brouillons. L'analyse esthétique inductive analyse le discours du point de vue de la perception, à partir du matériau musical tandis que l'analyse esthétique externe se base sur le ressenti de l'auditeur, dans le cadre d'expériences de psychologie expérimentale par exemple. À ces analyses s'ajoutent l'analyse communicationnelle qui correspond à l'étude de l'articulation entre les analyses précédentes. Selon J.J. Nattiez, la théorie des diagrammes formels de Jean-Marc Chouvel [33] serait un exemple d'analyse esthétique inductive. En réalité, l'algorithme cognitif modélise le point de vue de l'auditeur mais est appliqué à partir de l'œuvre même, il s'agirait donc plutôt d'une analyse au niveau immanent. La musique y est étudiée comme un phénomène et non pas en tant qu'objet et avec un sens de lecture spécifique, en tant que phénomène temporel.

1.2 Vers une modélisation de la cognition et de la psychologie de l'écoute

L'analyse musicale vise à une meilleure compréhension de la musique. Elle y est souvent considérée comme un objet spatialisé dans une trace, en particulier sous forme de partition. Il convient pourtant de restituer ce phénomène dans le temps, en tant que processus cognitif. Les sciences cognitives se sont beaucoup intéressées au phénomène musical, et il apparaît déterminant pour l'analyse de mieux comprendre et de mieux représenter l'intentionnalité temporelle au cœur des œuvres.

1.2.1 Psychologie de l'écoute

François Delalande [42] a montré l'influence du sujet dans la compréhension du musical et a étudié différents types d'écoutes en fonction de l'auditeur. L'amusie [24] témoigne par exemple dans certains cas d'une audition parfaite, où l'individu concerné est pourtant incapable de distinguer une mélodie. Les illusions auditives par ailleurs telles que la gamme de Shepard [148] et les illusions de Jean-Claude Risset [131] sont d'autres exemples d'effets perçus distincts de la logique acoustique qui les produit. On retrouve des effets "hallucinatoires" également dans des œuvres musicales telles que les chants Sardes *Miserere quaresimale*, qui sont chantés de sorte à rendre audible la *quintina*, une cinquième voix pourtant non présente physiquement.

Certains chercheurs essayent de mieux comprendre les effets de la musique : L. Meyer [103] cherche ainsi à expliquer l'émotion dans la musique en la liant aux phénomènes d'attention et à la sémantique musicale en s'inspirant de la *Gestalt Theory* et des théories pragmatiques de Charles Sanders Peirce et John Dewey [46] à propos des réponses émotionnelles liées au sens. Gisèle Brelet [23] soutient par ailleurs que la musique suscite en nous "*l'être que nous devrions être*". Le psychologue John Sloboda [150] a ainsi réalisé une expérience au cours de laquelle 83 personnes ont recensé des œuvres leur ayant fait ressentir des émotions particulières, en indiquant les éléments musicaux à l'origine de ces émotions. Par une analyse très précise de la partition, ils ont pu déduire que certaines formes musicales spécifiques donnaient lieu à des émotions spécifiques. Par exemple les harmonies inattendues provoquaient des frissons tandis que les appoggiatures provoquaient les pleurs.

1.2.2 Prise en considération de la musicologie cognitive

Les considérations de la psychologie cognitive et de l'analyse musicale débouchent sur la musicologie cognitive. Celle-ci est présentée par O. Laske [85], qui cherche à comprendre les rapports entre les processus cognitifs et le phénomène musical. Dans la préface de *Musique et cognition, perspectives pour l'analyse et la performance musicale* [83], P. Lalitte explique que les analyses psychologiques et musicales vont bien souvent à contre-courant. En effet, il énonce qu'après la seconde guerre mondiale, l'analyse musicale est centrée "sur l'objet" tandis que l'analyse psychologique est "centrée sur le sujet". De ce fait, les musicologues critiquent les conditions expérimentales des laboratoires, où les analyses sont effectuées sur un *stimuli* bien trop limité pour être de l'ordre du musical. Pendant ce temps et jus-

qu'aux années 1980, P.Lalitte énonce que la musicologie analytique se concentre plus sur les théories grammaticale et syntaxiques de la mouvance de la linguistique que sur les théories perceptives. La fondation de l'ESCOM (*European Society for the Cognitive Sciences of Music*) par I. Deliège [44] permet alors de développer la musicologie cognitive et joindre les deux domaines que sont la psychologie cognitive, très expérimentale, et la modélisation théorique des phénomènes cognitifs.

Ainsi, de nouveaux systèmes d'analyses tels que des *machines à écouter*, selon le titre du livre d'Émile Leipp [87], émergent. Bien que le sonagramme ait permis de mieux comprendre le monde sonore, il manque encore à l'analyse musicale un *morphoscope* [102] opérationnel, permettant de visualiser l'évolution temporelle de l'œuvre en tenant compte des impératifs cognitifs.

Ainsi, Jean-Marc Chauvel présente les concepts de l'Algorithme Cognitif et du diagramme formel [33]. L'algorithme Cognitif est un modèle qui permet de structurer et d'effectuer une analyse paradigmatique de l'œuvre musicale dans le temps au fur-et-à-mesure de l'acquisition du signal. Il produit à partir d'une représentation musicale un ensemble de diagrammes formels qui représentent les paradigmes musicaux en fonction du temps sur différentes échelles temporelles. L'algorithme Cognitif est présenté de manière exhaustive au chapitre 2. Dans ce modèle, le temps, bien que non nécessaire à l'analyse musicale, est une contrainte forte. La question se pose de savoir si cette contrainte suffit à proposer un modèle qui se rapproche du fonctionnement cognitif lors d'une écoute musicale.

Dans le cadre de cette thèse, on cherche à automatiser l'algorithme cognitif, ce qui nous pousse à définir une structure de données hiérarchisée et à proposer une étude de la similarité et de la segmentation entre des objets audio ou symboliques à l'aide des outils informatiques existants.

1.3 Musicologie computationnelle

Les avancées technologiques en informatique ont pu permettre le développement de l'analyse musicale computationnelle. Max Matthews propose les premières synthèses musicales en 1957 avec son logiciel *Music* [98] et des premières analyses musicales informatisées s'en suivent à la fin des années 1960. On observe par ailleurs une dichotomie d'une part entre les outils de la physique par opposition à ceux des mathématiques discrètes, et d'autre part un conflit historique entre l'informatique ,liée au signal, et le sérialisme, ainsi qu'entre le GRM avec l'utilisation des spectres, par opposition à l'informatique liée au symbolique et la *Set Theory* d'Allen Forte [55]. Une question fondamentale et qui se posera au cours de cette thèse est ainsi

de savoir comment faire un pont entre ces deux modélisations de la musique. Par ailleurs, la question que nous nous posons est ainsi de savoir comment automatiser grâce à des outils informatiques l'analyse et plus spécifiquement s'il est possible d'automatiser l'analyse paradigmatique.

1.3.1 Traitement du signal

Le son est un phénomène physique correspondant à la propagation de vibrations mécaniques d'un fluide sous forme d'ondes sonores. La représentation temporelle d'une onde sonore pure, c'est-à-dire composée d'une seule fréquence, présente plusieurs caractéristiques. L'une d'elle est la périodicité, qui correspond à la perception de la hauteur. La période est la durée qui sépare deux répétitions successives du phénomène. Il s'agit de l'inverse de la fréquence. L'onde possède également une amplitude maximale, qui correspond à la perception du volume sonore. Enfin la forme d'onde correspond à la perception du timbre.

On peut voir en haut de la figure 1.1 la forme d'onde du la 440 Hertz, qui est une sinusoïde pure.

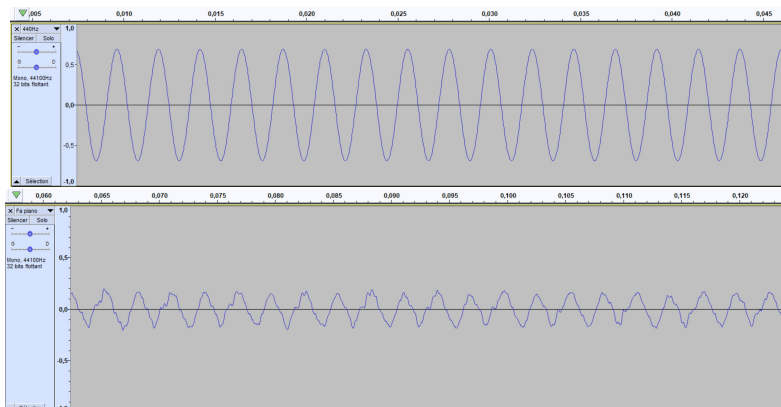


FIGURE 1.1 – Forme d'onde du *la* 440 Hz (sinusoïde pure, haut) et Forme d'onde du *fa* 349,23 Hz (somme de sinusoïdes, bas).

Un son complexe est la somme de plusieurs sinusoïdes pures de fréquences et d'amplitudes données. La figure 1.1, présente également la forme d'onde d'un *fa* joué au piano : sa forme est due à la somme de différentes sinusoïdes, qui avec des amplitudes spécifiques, donnent ce timbre particulier. En figure 1.2, on peut observer la forme d'onde du *Rondo* de la *Sonate pour piano n°16* de Wolfgang Amadeus Mozart.

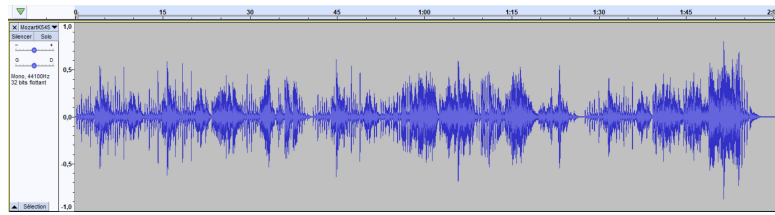


FIGURE 1.2 – Forme d’onde du *Rondo* de la *Sonate pour piano n° 16* de Wolfgang Amadeus Mozart, obtenus avec le logiciel *Audacity*.

Une deuxième représentation du son est le spectre, qui présente l’amplitude pour chacune des fréquences qui constituent le son. La figure 1.3 présente en haut un spectre d’une sinusoïde pure en *la 440 Hertz* et en bas le spectre d’un *fa 349,23 Hertz* joué au piano.

Les pics correspondent aux fréquences qui constituent le son. Dans le cas d’une sinusoïde pure, telle que le *la 440 Hertz*, il y a un unique pic au niveau de la fréquence correspondante. Le son de piano quant à lui est beaucoup plus complexe et il est constitué en réalité de nombreux pics. Le premier pic est appelé fondamental et les pics suivants correspondent aux harmoniques ou partiels. Cette décomposition en somme de sinusoïdes est possible grâce à la transformée de Fourier, présentée au chapitre 3.

Afin d’analyser des œuvres, notamment les œuvres musicales non écrites, on peut représenter l’évolution du spectre en fonction du temps : il s’agit du sonagramme.

Le sonagramme est un diagramme qui représente la puissance sonore en fonction du temps (en abscisse) et de la fréquence (en ordonnée). La couleur du point représente sa puissance sonore. Cela permet donc de voir l’évolution au cours du temps des paramètres fréquentiels et l’intensité d’un son. On peut ainsi, en observant un sonagramme, reconnaître par exemple le timbre d’un instrument. Initialement, les sonagrammes étaient obtenus à l’aide du sonagraphe [125], créé en 1941. Aujourd’hui, ils sont obtenus par calcul de la transformée de Fourier aux différents instants de la musique.

La figure 1.4 présente les sonagrammes du *la 440 Hertz* en sinusoïde pure (haut), du *fa 349,23 Hz* joué au piano (milieu) et du *Rondo* de Mozart (bas).

Aujourd’hui en informatique musicale, la stratégie employée de manière générale est d’extraire des descripteurs audio du signal afin de faciliter le repérage des ruptures d’homogénéité et de répétition. Cela permet de s’attaquer à une écri-

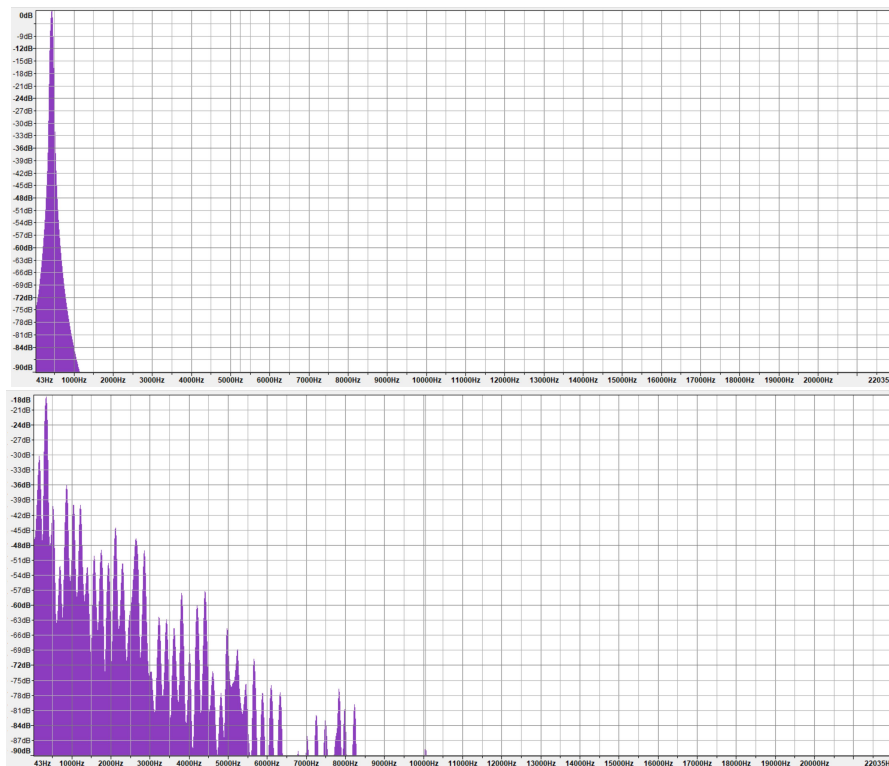


FIGURE 1.3 – Spectre d’une sinusoïde pure en la 440 Hz (haut) et d’un fa 349,23Hz joué au piano (bas), obtenus avec le logiciel *Audacity* (fenêtre de Hann, 1024 échantillons).

ture de plus haut niveau qui permettrait de classer sous une même catégorie deux éléments sensiblement différents. On peut citer parmi les descripteurs majoritairement utilisés : les Mel Frequency Cepstral Coefficients pour le timbre [92].

Nous pouvons également citer les chromas, qui permettent d’associer au signal l’énergie moyenne de la gamme. Pour le rythme, on analyse les émissions des notes, c’est-à-dire les attaques, et les sons percussifs par l’analyse de l’évolution du spectre d’intensité sonore [145]. On peut de manière générale calculer de nombreux descripteurs permettant de donner par exemple les valeurs de la pulsation, du tempo, de la métrique, de l’harmonie et combiner ces différentes approches dans une analyse multi-strate avec éventuellement l’utilisation du Deep Learning, c’est-à-dire l’utilisation de réseaux de neurones artificiels pour l’apprentissage automatique à partir d’un ensemble de données importants, comme dans [165].

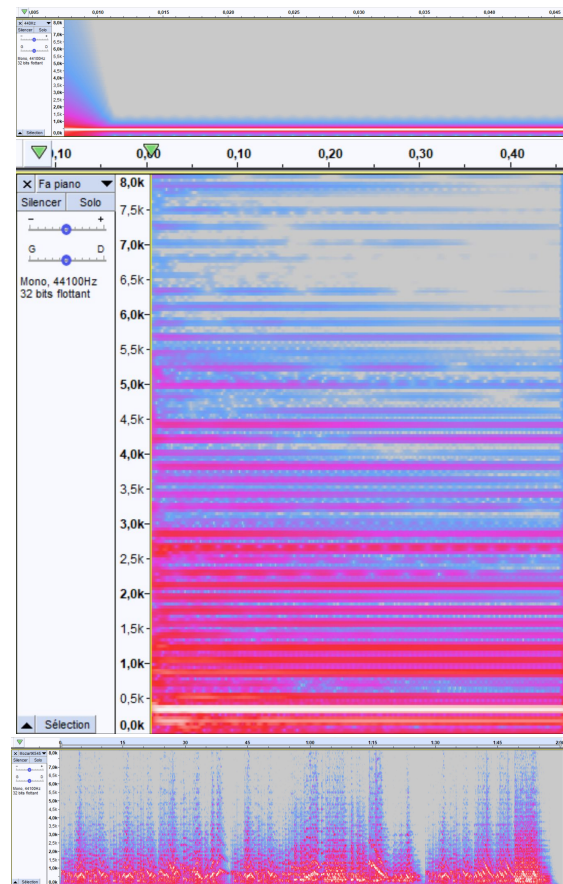


FIGURE 1.4 – Sonagramme du la 440 Hz joué au piano (haut), du fa 349,23 Hz joué au piano (milieu) et du *Rondo* de Mozart (bas), obtenus avec le logiciel *Audacity* (fenêtre de Hann, 1024 échantillons).

1.3.2 Logiciels et algorithmes d'analyses

De nombreux outils ont été développés pour réaliser des analyses d'œuvres musicales.

L'acousmographe [57] notamment, est un logiciel développé par l'Ina-GRM depuis 1989. Il propose des représentations des musiques électro-acoustiques à partir de l'analyse spectrale du signal associé. On peut également annoter les différentes représentations et ce logiciel peut être utilisé de manière générale pour tout type de fichier audio.

EAnalysis [38] est un logiciel développé par Pierre Couprie depuis 2010 et toujours en développement. Ce logiciel permet de visualiser de différentes manières

et sous plusieurs vues le sonagramme ainsi que la forme d'onde. L'utilisateur peut aussi obtenir des données analytiques sur ces vues et le logiciel permet par ailleurs de proposer une représentation graphique de l'œuvre musicale à l'aide d'outil d'édition, ce qui est particulièrement pratique pour les œuvres électro-acoustiques.

Benjamin Martin (2012,[96]) propose un algorithme pour analyser la structure musicale à partir de l'audio. Il reprend un algorithme de comparaison de chaînes de caractères dans les séquences d'ADN, BLAST [4] pour éviter lors de la phase de classification d'effectuer toutes les comparaisons possibles et ainsi gagner en temps de calcul. Cela impose par ailleurs l'utilisation de descripteurs robustes à l'altération de motifs et à la transposition. Il a également utilisé cet algorithme [97] pour la reconnaissance de reprises musicales dans de larges bases de données. Benjamin Martin propose également un modèle hiérarchique du système en représentant l'œuvre par un arbre qu'il produit en recherchant de manière descendante les sous-structures des structures détectées. Néanmoins, il ne s'attaque pas aux problématiques de la temporalité ni de l'effet sur le cognitif. Par ailleurs, il se restreint aux styles musicaux conventionnels qui permettent une certaine répétition dans la structure de l'œuvre. La thèse de Gabriel Sargent [139] propose une estimation de la structure d'un morceau de musique à partir de l'audio par une approche différente, en s'inspirant des notions linguistiques de paradigme et de syntagme pour définir ce qu'il appelle une structure sémiotique : il crée un principe méthodologique d'analyse structurelle à partir de l'étude interne des blocs, grâce au modèle système-contraste [15]. Il effectue également une analyse paradigmatique qui étudie les relations entre les blocs d'une même classe et les oppositions entre blocs de classes différentes, afin de permettre l'étiquetage des segments, et une analyse syntagmatique qui étudie les blocs vis-à-vis de leurs voisins. Dans son étude, il n'y a pas de hiérarchisation et il choisit lui-même la taille des segments auxquels il souhaite se référer. Gabriel Sargent se limite également aux styles musicaux conventionnels.

Antescofo, développé par l'IRCAM [35] est un logiciel de suivi de partitions synchronisées au jeu du musicien. Il analyse des caractéristiques par toutes sortes de descripteurs et agit en fonction d'événements inscrits sur la partition. Il est également équipé d'un système qui permet de prévoir le tempo du musicien afin de s'adapter à celui-ci. Il modélise ainsi un système d'écoute qui prend en compte l'évolution temporelle de l'œuvre et est capable de reconnaître un élément qui a déjà surgi par le passé. OSSIA SCORE [29], développé au sein du SCRIME² et du collectif ossia.io³ propose une modélisation hiérarchique et temporelle en intégrant un modèle hybride du temps pour écrire et exécuter la partition.

2. <https://scime.u-bordeaux.fr/>

3. <https://ossia.io>

La matrice d'autosimilarité [53] consiste en une matrice symétrique dont l'ordonnée comme l'abscisse représente le déroulement temporel du morceau et où chaque valeur $s(i, j)$ correspond à la distance entre les descripteurs calculés à l'instant i et à l'instant j . La différence majeure avec l'algorithme cognitif est que cette matrice est calculée entièrement sur l'ensemble de l'œuvre et non pas dynamiquement, ni en tirant conséquence des similitudes repérées et sans explicitation de la structure.

Une étude approfondie des logiciels informatiques et projets d'analyse musicale assistée par ordinateur est présentée par Christophe Guillotel-Nothmann dans [64], et détaille les logiciels en fonction de leur contexte chronologique, géographique, de leurs données d'entrées audio/vidéo, symbolique ou langage de contrôle, leurs catégories analytiques et leurs répertoires. Il en déduit quatre axes de recherches qui sont la pérennité des données et des standards de celles-ci, la méthodologie et la conception d'outils globaux, l'apport de ces technologies pour le développement de la connaissance des œuvres et enfin l'extraction d'une sémantique et l'approche pragmatique que l'on pourrait tirer d'une analyse informatisée. Il existe donc de nombreuses méthodes d'analyse musicale automatisées, mais soit elles procèdent à l'analyse de l'œuvre directement sur son ensemble sans considération du déroulement temporel de l'œuvre, soit le matériau d'analyse a une taille imposée et on ne trouve pas de construction ascendante de l'analyse de l'œuvre.

On souhaite ainsi proposer un logiciel qui permet une construction d'une part soucieuse de l'ordre chronologique d'apparition des matériaux, d'autre part dont le matériau d'analyse est construit de manière hiérarchique par strate reposant sur la construction des niveaux inférieurs.

Conclusion

Le travail de cette thèse s'inscrit donc comme une réponse aux besoins émergents de la musicologie cognitive d'expliquer le phénomène musical en tant qu'objet perçu par un auditeur. L'algorithme cognitif présente ainsi une représentation formelle hiérarchisée de l'œuvre, obtenue à partir d'une description immanente de celle-ci mais dans le but de modéliser les processus cognitifs. L'objectif de ce travail est d'automatiser l'Algorithme Cognitif de Jean-Marc Chouvel et la production des diagrammes formels par le développement d'un nouveau logiciel d'analyse musicale intitulé MORFOS. À travers cette automatisation, on cherche à explorer un modèle pertinent pour la représentation multi-échelle de la forme et les liens que cette représentation explicite entre les différents niveaux pour la structure et l'organi-

sation interne : ce modèle est présenté au chapitre 2. Ce logiciel prend en entrée des fichiers au format audio et applique donc les problématiques liées au traitement du signal tels que l'extraction de spectre et de descripteurs. Nous obtenons néanmoins rapidement une modélisation symbolique de l'œuvre musicale analysée, c'est pourquoi le logiciel soulève également le problème du rapport entre l'objet musical en tant que signal et sa représentation symbolique. L'implémentation est présentée au chapitre 3.

Chapitre 2

L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

Dans ce chapitre, nous présentons six notions fondamentales autour desquelles MORFOS est construit :

Le *Diagramme Formel* est la représentation visuelle à un niveau de structure temporelle, fournie par l'*Algorithme MORFOS* à partir d'une représentation audio ou écrite de l'œuvre musicale. Une multiplicité de diagrammes formels peuvent être créés en fonction de critères spécifiques de l'analyse et des niveaux hiérarchiques considérés. Ils peuvent être organisés de manière *hiérarchisée* et assemblés dans ce que l'on appellera *Diagramme Formel Multi-échelle* et qui permet de représenter et d'analyser la structure temporelle ainsi qu'inter-échelle de l'œuvre musicale. Cette représentation qui est obtenue au fur et à mesure de l'acquisition du signal dans le temps est intrinsèquement liée à une modélisation de l'écoute, ou plutôt *des écoutes*, propres à chaque individu. L'écoute active, ainsi que l'Algorithme MORFOS, permettent la création d'une représentation de l'œuvre musicale grâce aux phénomènes de *similarité* et de *segmentation*. Seulement les critères de ces deux phénomènes dépendent de la sensibilité de chacun et il en est également ainsi de la projection faite de l'analyse, qui peut être *multi-dimensionnelle*. Ainsi, une représentation donnée d'un ensemble de diagrammes formels hiérarchisés sera obtenue à partir de la projection selon la dimension spécifiée.

2.1 Le Diagramme Formel

L'Algorithme Cognitif et la notion de Diagramme Formel sont introduits dans le chapitre 1. On présente dans un premier temps la représentation telle que proposée par Jean-Marc Chouvel [33] avant de présenter notre formalisation, adaptée à l'automatisation de ce processus analytique.

2.1.1 Présentation du Diagramme Formel de J.-M. Chouvel

Un **diagramme formel** est la représentation des matériaux d'une œuvre en fonction du temps. Le choix de ce terme, qui peut être discuté de part son usage musicologique, est défini ici de la manière suivante.

Un *matériau* est défini comme un paradigme au sens linguistique du terme, c'est-à-dire une classe d'éléments musicaux substituables. En ce sens, on pourrait parler de diagrammes "paradigmatiques". Le matériau se distingue de l'*objet*, qui est une occurrence du matériau dans une œuvre donnée. Par exemple un matériau peut être la note *La*, tandis qu'un objet va être la note *La* jouée à un instant précis dans une représentation audio d'un morceau donné, il s'agit d'une variation eidétique.

Le Diagramme Formel permet d'apporter une synthèse visuelle de l'œuvre musicale propice à l'interprétation musicologique. En effet, il décrit la forme musicale au sens présenté dans le chapitre 1 et permet donc une analyse paradigmatique et syntagmatique spécifique à chaque œuvre musicale en mettant en exergue ses matériaux constitutifs et leurs relations dans le temps.

La figure 2.1 représente un diagramme formel du *Geisslerlied Maria muoter reinû maît*. Il s'agit d'un chant allemand populaire de "flagellation" monophonique datant de 1349. Cette œuvre est le premier exemple pris par Nicolas Ruwet [134] pour effectuer une analyse paradigmatique, et elle est reprise par Jean-Marc Chouvel dans [33] en illustration du concept de diagramme formel.

L'axe des ordonnées représente bien les matériaux tandis que l'axe des abscisses représente le temps. À partir d'un tel diagramme, une analyse musicologique de l'œuvre peut être effectuée. Chouvel présente ainsi que "Toute la stratégie de la partition semble axée sur le resserrement temporel entre ce questionnement ultime du sib-la et la réponse de la stabilité sur le fa". On constate en effet sur les quatre premières mesures un resserrement autour de la note "la" (fa vers la, fa-la, do-la, si-la) avant de conclure sur la tonique fa. Ce resserrement peut être retrouvé lorsque

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

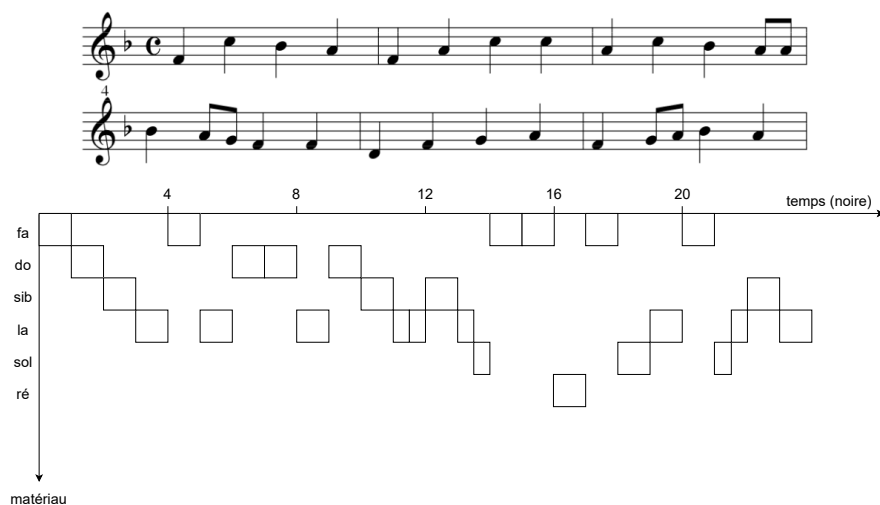


FIGURE 2.1 – Partition (haut) et diagramme formel (bas) du *Geisslerlied Maria muoter reinû maît* au niveau structurel de la note.

le diagramme formel est produit sur un segment musical plus large à l'échelle de la mesure, ce qui nous permet de constater un mimétisme de forme entre les différentes échelles. Ce resserrement est représenté figure 2.2.

La figure 2.3 représente également un diagramme formel à l'échelle des motifs mélodiques du Rondo du Menuet K545 de Wolfgang Amadeus Mozart produit par Jean-Marc Chouvel et présenté dans [33], ainsi que la partition analysée.

Le refrain du Rondo apparaît sur le diagramme, représenté par les cinq premiers matériaux suivants :

- un motif contenant les trois croches et un demi-soupir ;
- quatre doubles croches dont trois notes ascendantes et une descendante ;
- quatre doubles croches ascendantes ;
- quatre doubles croches descendantes ;
- une noire pointée croche, suivie d'une croche et d'un demi soupir.

Ce refrain marque un retour à la stabilité dans l'œuvre musicale, par opposition aux développements autour des mesures 32 à 40 et 44 à 51, marqués par la découverte de nombreux matériaux. Cela nous amène à la notion de *Front de découverte*, qui correspond à la ligne de découverte de nouveaux matériaux, par opposition à la notion de *Fond de répétition*, qui est la zone du diagramme où se répètent des matériaux déjà entendus. Ces notions sont en réalité intrinsèquement liées à la mémorisation [11] et à la quantité d'information, notamment la quantité d'information qu'un auditeur est en mesure de traiter [124]. L'hypothèse peut ainsi être émise qu'il existe un juste milieu entre la répétition et la découverte afin d'en-

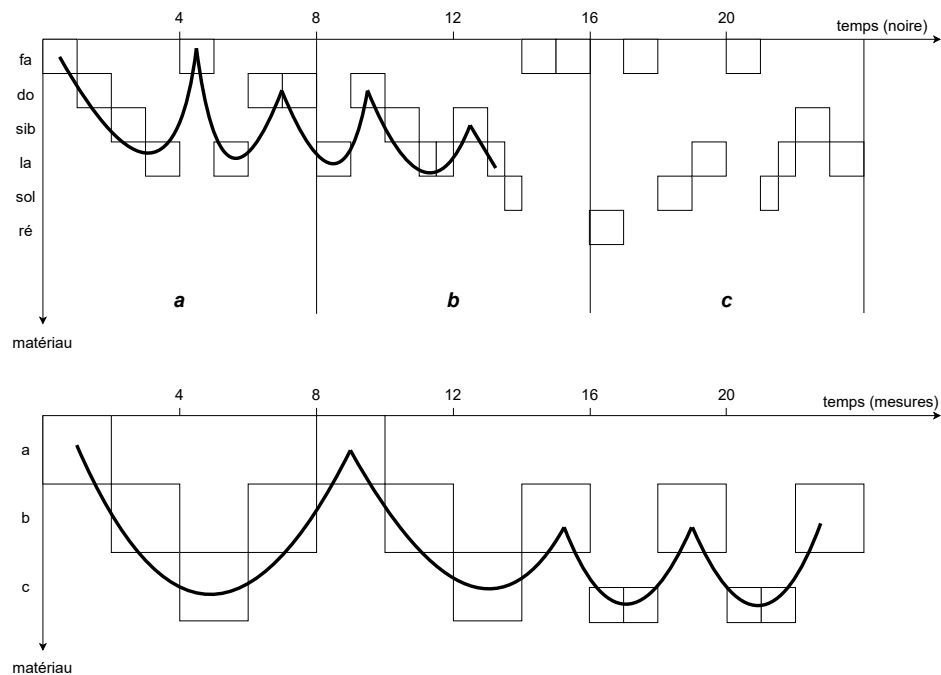


FIGURE 2.2 – Représentation du resserrement sur les premières mesures du *Geisslerlied* à l'échelle de la note (haut). L'extrait est regroupé en trois parties a, b et c, qui sont reprises tout au long de la pièce telles que représentées sur la figure du bas. On retrouve la figure de resserrement à l'échelle de la mesure.

tretenir l'intérêt et la compréhension de l'auditeur, et la forme de cette courbure permet de nous instruire sur la stratégie du compositeur à ce propos.

Cette représentation des diagrammes formels polyphoniques pour l'œuvre analysée est tracée "à la main". Elle montre en particulier le tuilage des motifs de croches au début du Rondo, et permet de lire sur le diagramme la superposition des matériaux à un instant t donné.

Enfin, alors que cela semblait spontané dans le *Geisslerlied* à l'échelle de la note, l'ambiguïté liée à la segmentation et la similarité peut déjà être soulignée. En effet, un premier constat est que le choix de la segmentation des matériaux découle de connaissances musicologiques : il n'y aura pas de segmentation par exemple à l'échelle de la phrase, au milieu d'une cadence parfaite. Par ailleurs, les objets considérés comme appartenant à la même classe de matériaux ne sont pas strictement similaires. Sans même considérer les variations liées à l'interprétation, le motif des trois croches suivies d'un demi silence est considéré comme étant quatre fois le même matériau alors qu'il s'agit à chaque fois de notes différentes.

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle :
formalisation de MORFOS



FIGURE 2.3 – Partition (haut) et diagramme formel par Jean-Marc Chouvel (bas) du Rondo K545 de W.A. Mozart. Les premiers matériaux mélodiques élémentaires considérés sont encadrés en pointillés sur la partition.

Les variations ne sont par ailleurs ici pas annotées sur le diagramme. La difficulté d'un modèle qui automatise ces opérations est donc de déterminer les critères et les opérations permettant de déterminer une similarité et une segmentation, indépendamment d'une étude musicologique préalable. Ceci a en revanche l'intérêt de pouvoir apporter une représentation indépendante des connaissances du musicologue qui effectuera l'analyse à partir de celle-ci et donc d'éventuellement présenter des diagrammes formels qui illustreront des subtilités que le musicologue aura écartées par des biais cognitifs. D'autres biais en revanche auront pu être introduits par les types de mesures choisies pour les calculs de similarité et de segmentation.

Nous présentons ainsi dans la suite un modèle informatique du Diagramme Formel et la formalisation que l'on a choisie pour le réaliser.

2.1.2 Définitions de notre modèle

Dans notre modèle, nous gardons la définition du diagramme formel comme étant la représentation de matériaux de l'œuvre musicale et l'explicitation de leurs relations au cours du temps. On notera que la terminologie ici de matériau et d'objet s'éloigne de celle de Jean-Marc Chouvel. En effet, les critères de segmentation automatisés ne permettent pas toujours d'obtenir des objets (et par extension des matériaux) qui font sens en soi.

2.1.3 Algorithme

Pour obtenir un diagramme formel, l'algorithme 1, provenant de l'ouvrage *Analyse Musicale* de Jean-Marc Chouvel [33] est présenté.

Algorithme 1 Calcul d'un diagramme formel

pour chaque fragment entendu :
comparaison avec les fragments précédents ;
s'il est différent :
 inscription en mémoire « matériau » ;
dans tous les cas : inscription en mémoire « forme » ;

On relève que le terme de *fragment* apparaît ici et peut être associé par définition aux termes "partie", "fragmentation", et donc à la notion de *segmentation* (*test de reconnaissance syntagmatique*). Un pré-calcul, non explicité ici, est donc associé à l'obtention de fragments à partir d'une *représentation initiale*. L'algorithme effectue ensuite une comparaison, qui est associée à la notion de calcul

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

de *similarité* (test de reconnaissance paradigmatique). Jean-Marc Chouvel désigne ainsi le fragment de la manière suivante : "Il ne comporte aucune restriction particulière au niveau de la notion de fragment si ce n'est celle d'une homogénéité suffisante pour autoriser la fonction de comparaison". Ainsi, avant d'être dans une *classe*, les objets, non classés, sont des *fragments*. Il s'agit donc d'un état entre la segmentation et l'étude de la similarité.

Enfin, en fonction des résultats de ce deuxième test de reconnaissance paradigmatique, le fragment entendu est inscrit dans une ou deux des mémoires *matériau* et *forme*. Ici, la *mémoire matériau* correspond au stockage des matériaux de l'œuvre musicale et nécessite l'inscription sur l'axe des matériau d'un nouveau matériau jamais entendu jusque là. La mémoire forme quant à elle correspond à l'agencement des matériaux au cours du temps et nécessite l'inscription systématique du fragment dans le temps, en fonction du matériau correspondant. Les critères et opérations de segmentation et similarité ne sont pour le moment pas précisés.

Prenons l'exemple du *Geisslerlied* présenté précédemment en figure 2.1. La représentation initiale est celle de la partition. Alors que nous lisons la partition, nous segmentons chaque note lue. Pour chaque fragment entendu, ici la note, celle-ci est comparée avec les notes entendues précédemment. Si elle est différente de tous les fragments précédents (c'est le cas par exemple pour chacune des quatre premières notes), alors elle est inscrite en mémoire matériau, c'est-à-dire qu'une nouvelle ligne est créée pour le matériau correspondant dans le diagramme formel. Dans tous les cas, elle est inscrite dans la mémoire forme, l'objet étant représenté par exemple par un carré à l'instant t correspondant sur l'axe du temps dans le diagramme. La figure 2.4 représente l'inscription des 5 premiers fragments du *Geisslerlied*. Le processus est décrit de gauche à droite et de haut en bas : tout d'abord, le diagramme formel et les mémoires matériau et forme sont vides car nous n'avons acquis aucun fragment. Le premier fragment, *fa*, est détecté et comparé à l'existant dans la mémoire matériau. Celle-ci est vide, le *fa* est donc inscrit à l'instant 1 l'objet dans le diagramme ainsi que dans la mémoire matériau et une nouvelle ligne est créée pour ce matériau dans le diagramme formel. Le même procédé vaut pour les notes *do*, *si bémol* et *la*, qui sont inscrites, chacune à son tour en mémoire matériau et aux instants 2, 3 et 4 du diagramme formel. Un nouveau *fa* est ensuite à nouveau acquis, et comparé aux matériaux déjà existants est retrouvé. Il est ainsi inscrit uniquement en mémoire forme, c'est-à-dire à l'instant 5 dans la ligne correspondante.

L'algorithme peut être décrit en quatre étapes distinctes :

- l'acquisition au fur et à mesure d'une représentation initiale ;

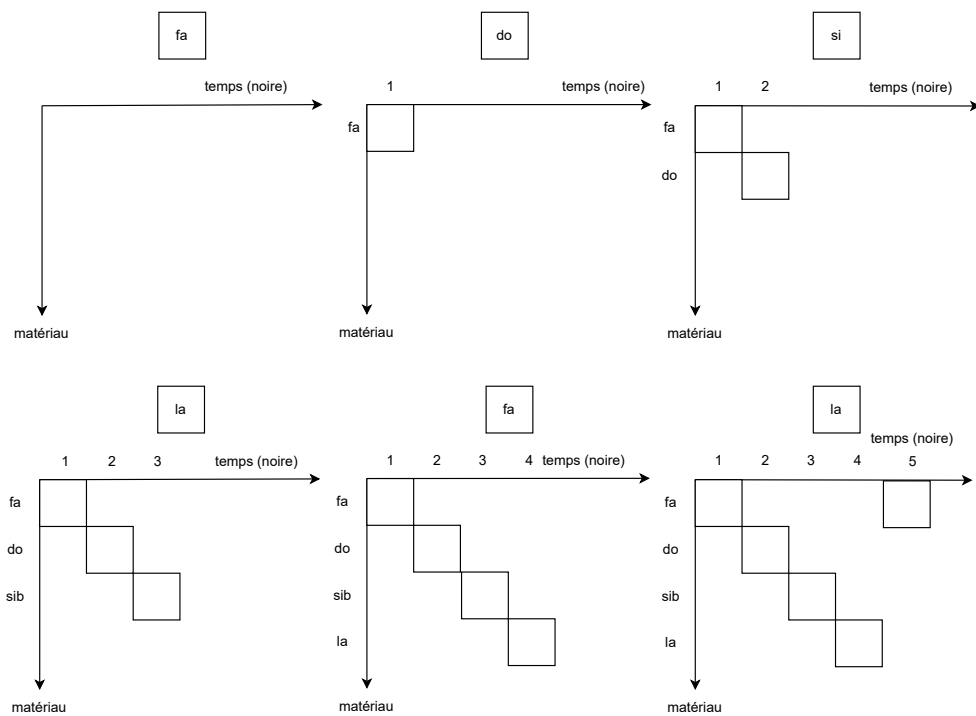


FIGURE 2.4 – Création en temps-réel du diagramme formel à l'échelle de la note du *Geisslerlied*.

- la segmentation en fragments de cette première représentation ;
- la classification du fragment actuellement écouté ;
- l'acquisition au fur et à mesure du diagramme formel.

L'Algorithme MORFOS et les deux opérations de *segmentation* et *classification* sont représentés en figure 2.5.

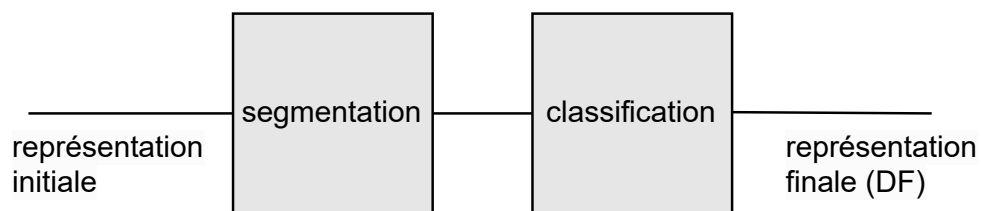


FIGURE 2.5 – Diagramme de l'algorithme MORFOS.

Ainsi, un diagramme formel est obtenu à partir d'une représentation initiale, en appliquant des opérations de segmentation et de classification.

2.1.4 Formalisation

Nous distinguons le logiciel global MORFOS de l'Algorithme MORFOS permettant d'obtenir le diagramme formel. MORFOS contient des phases de pré-traitement avant de fournir un flot de données à l'Algorithme MORFOS. Il s'agit donc du système {pré-traitement, Algorithme MORFOS}.

Deux types de représentations sont prises en compte à l'entrée de MORFOS :

- les représentations audio (signal numérisé) ;
- les représentations symboliques ordonnées (alphabet).

Les représentations audio sont le signal brut obtenu à partir d'un enregistrement ou une synthèse de l'œuvre musicale.

La représentation symbolique est la succession de caractères correspondant aux objets représentés par les labels des matériaux initiaux associés (avant segmentation). Ces labels n'ont pas d'intérêt sémantique. Ils sont définis sur un alphabet \mathcal{A} . On définit ainsi une *représentation initiale* r comme un mot

$$r = x_1 \dots x_i \dots x_l$$

avec x_i des lettres définies sur l'alphabet \mathcal{A} .

la représentation initiale est un mot ordonné strictement selon l'ordre chronologique des caractères tels que $\forall (x, y) \in r, x < y$ implique que la première apparition de x à eu lieu avant celle de y .

Nous choisissons ainsi de nommer les matériaux alphabétiquement ou numériquement par ordre d'apparition. Le premier matériau apparu lors d'une analyse sera nommé a , le second, b , et ainsi de suite. Les deux écritures alphabétiques et numériques seront considérées équivalentes.

L'ordonnement des caractères signifie donc que l'ordre d'apparition des matériaux est déjà pris en compte par le calcul d'une similarité *relative* pour l'obtention de la représentation initiale symbolique. Cette représentation symbolique est obtenue soit manuellement par l'analyse préalable d'un musicologue sur une représentation donnée, soit de manière automatique par exemple directement par MORFOS.

Que l'entrée soit audio ou symbolique, nous choisissons donc de formaliser un morceau de musique comme un flot d'informations, c'est-à-dire, une succession d'objets.

Dans notre modèle, les objets se succèdent strictement. Par ailleurs, une œuvre musicale étant bornée par son début et sa fin, pour toute partie non vide p de r , il

existe un élément x de p n'ayant aucun antécédent dans p selon la relation R définie telle que $yRx \Leftrightarrow x$ est le premier nouveau matériau qui succède chronologiquement à y .

Ainsi, on définit le *rang* du matériau x comme la fonction $rg(x)$ définie par $rg(x) = \cup\{rg(y) + 1 | yRx\}$. Cela correspond au nombre ordinal associé à l'ordre d'apparition du matériau dans l'extrait analysé.

L'*indice* d'un objet labélisé dans un mot r est également défini comme le numéro de l'emplacement de l'objet dans le mot r : il est ainsi indexé par son indice. Les indices et le rang commencent à 1 dans ce manuscrit. On admet alors la notation $x_i = r[i]$.

La représentation initiale $r = x_1...x_i...x_l$ est segmentée en un ensemble de facteurs dits *fragments*. On définit alors l'opération de segmentation $sgm(r, cs)$ prenant en argument la représentation initiale r et un ensemble de critères de segmentation cs . Le résultat de cette fonction est une *représentation segmentée* s qui est le regroupement de lettres de \mathcal{A} en facteurs w_j par associativité du produit du monoïde libre \mathcal{A}^* sur \mathcal{A} tel que

$$s = sgm(r, cs) = w_1...w_j...w_k$$

On calcule ensuite la similarité interne entre les fragments w_j de s . Pour cela, on définit une opération de classification $cls(s, cc)$ qui prend en argument une représentation segmentée s et un ensemble de critères de classification cc . La classification peut éventuellement procéder à des opérations de transformation sur les fragments w_i avant de les comparer¹.

Ainsi, la *représentation finale* est définie par le mot f telle que

$$f = cls(s, cc) = cls(sgm(r, cs), cc) = y_1...y_j...y_k$$

où les y_j sont des lettres définies sur un nouvel alphabet \mathcal{B} et correspondent aux *objets* qui constituent l'œuvre musicale, dont la lettre est le label associé au *matériau* correspondant.

La figure 2.6 illustre la formalisation en fonction des représentations dans MORFOS.

Dans notre modélisation actuelle, les w_i ne peuvent pas se recouvrir. Les musiques polyphoniques au sens contrapuntique ne sont donc pas représentées pour le moment.

1. On trouvera un exemple de transformation des fragments dans la partie suivante sur l'exemple de l'analyse des représentations obtenues à partir d'un fichier audio du *Geisslerlied*

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

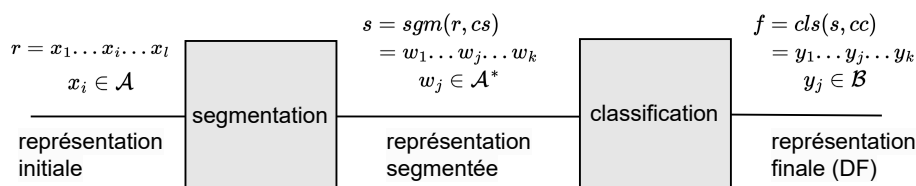


FIGURE 2.6 – Formalisation des différentes représentations initiale, segmentée et finale dans l'Algorithme MORFOS.

2.1.5 Exemples du *Geisslerlied* et du *Rondo* extrait du *Menuet K545* de W.A. Mozart

Considérons deux exemple concrets. Le premier est l'analyse de la représentation à partir d'un fichier audio du *Geisslerlied*, et le deuxième est l'analyse à partir d'une représentation symbolique du *Rondo* du *Menuet K545* de W.A. Mozart.

Exemple à partir de l'audio

On considère une analyse du *Geisslerlied* à partir d'une représentation audio de celui-ci. La première représentation qui est fournie au logiciel est un flux d'information qui est obtenu suite aux calculs des étapes suivantes dans MORFOS :

- tout d'abord, le fichier audio est numérisé,
- puis segmenté en *frames* de tailles fixes (ici, 1024 échantillons)
- puis une succession de spectres est calculé à partir de ces *frames*.

Le *Geisslerlied* à été obtenu à partir de la synthèse d'un fichier MIDI à 60 bpm auquel nous avons ajouté une seconde supplémentaire de silence au début du morceau. Le fichier dure ainsi 13 secondes. Il y a donc 560 spectres de 1024 échantillons obtenues à une fréquence d'échantillonnage de 44100Hertz. Chaque noire à 60bpm est constituée par ailleurs de 43 spectres successifs.

la représentation initiale est

$$r = x_1 \dots x_i \dots x_{560}$$

avec x_i défini sur l'alphabet \mathcal{A} qui est constitué de l'ensemble des spectres successifs du fichier audio.

On cherche à segmenter cette représentation avec la fonction *sgm* de telle sorte à obtenir des groupements de spectres. Ici, le critère *cs* est que l'algorithme segmente

lorsque il y a une différence importante entre l'objet entendu et le précédent :

$$s = sgm(r, cs) = (m_1 m_2 \dots m_{43}) \dots (m_{517} \dots m_{560}) = w_1 \dots w_j \dots w_{28}$$

Avec w_j des facteurs définis sur \mathcal{A}^* et qui correspondent à la concaténation de sous-ensembles consécutifs de spectres. A partir de cette représentation, une classification cls est effectuée sur les w_j obtenus afin de comparer l'information audio correspondante aux groupements des descripteurs. Pour cela, un pré-traitement est effectué par ailleurs et correspond au moyennage des spectres contenus dans chaque fragment.

$$f = cls(s, cc) = cls(sgm(r, cs), cc) = y_1 \dots y_j \dots y_{28} = \text{ABCDEBECCECDEEDEFBBGBFEBFEDE}$$

Les y_j sont définis sur l'alphabet \mathcal{B} qui contient les labels associés aux matériaux, qui sont les classes des w_j .

Exemple à partir d'une représentation symbolique

Un deuxième exemple, cette fois-ci à partir d'une représentation symbolique de l'œuvre est l'analyse du *Rondo* de W.A. Mozart étudié précédemment. On nomme ici *terme* les éléments plus complexes qu'un motif (qui est le niveau cellulaire, une suites de 2 ou 3 intervalles, de 3 ou 4 durées). À partir d'une étude de la partition en figure 2.3, la séquence à l'échelle des termes (cadres noirs) suivante est obtenue sur \mathcal{A} qui contient les labels des matériaux de la représentation initiale :

$$r = x_1 \dots x_i \dots x_{41} = \text{abacabacdeabfgabachijklmhinopqabacrsrsttu}$$

L'application de la fonction de segmentation $sgm(r, cs)$ (dont les critères cs seront détaillées plus tard) nous permet d'obtenir l'ensemble des facteurs suivants de \mathcal{A}^* :

$$\begin{aligned} s &= sgm(r, cs) \\ &= (ac)(ab)(ac)(de)(ab)(fg)(ab)(ac)(hi)(jklm)(hi)(nopq)(ab)(ac)(rs)(rs)(ttu) \\ &= w_1 \dots w_j \dots w_{18} \end{aligned}$$

Enfin, la fonction de classification $cls(s, cc)$ entre les différents facteurs (dont l'égalité stricte entre les chaînes de caractères suffit pour cet exemple) nous permet d'obtenir le nouveau mot f sur un ensemble \mathcal{B} qui contient les labels des matériaux de la représentation finale :

$$f = cls(s, cc) = cls(sgm(r, cs), cc) = y_1 \dots y_j \dots y_{18} = \text{ABABCADABEFEGABHHI}$$

Notons que la lettre A de la représentation finale appartient à \mathcal{B} et décrit une autre information que la lettre a de la représentation initiale, qui appartient à \mathcal{A} . Il y a une relation d'équivalence entre $y_1 = A \in \mathcal{B}$ et $w_1 = (ab) \in \mathcal{A}^*$

2.1.6 Équivalence entre les représentations symboliques et le Diagramme Formel

Pour le moment, nous avons montré comment nous passons d'une représentation initiale à une représentation finale en passant par des opérations de segmentation et de classification. En quoi la représentation finale est-elle relative à un diagramme formel ?

La relation d'ordre des lettres de la représentation initiale nous permet d'avoir une équivalence assez immédiate. En effet, il nous faut pour cela lire le mot de gauche à droite. Si le rang de la lettre qui est actuellement lue est supérieur à la taille de la mémoire matériau, alors une nouvelle ligne est créée. Dans tous les cas, la lettre est inscrite à son instant d'apparition dans la mémoire forme.

Pour le *Geisslerlied*, avec un silence au début du fichier audio, l'algorithme produit comme vu précédemment la séquence suivante :

ABCDEBECCECDEEDEFBBGBFEBFEDE

avec le tableau de correspondance suivant :

label	matériau	rang
A	<i>silence</i>	0
B	<i>fa</i>	1
C	<i>do</i>	2
D	<i>sib</i>	3
E	<i>la</i>	4
F	<i>sol</i>	5
G	<i>ré</i>	6

Le diagramme formel *final* obtenu est présenté en figure 2.7.

À partir de la séquence *ababcadabefegabhhi* obtenue suite au traitement du *Rondo* de W.A. Mozart, le diagramme formel final est également présenté en figure 2.8

2.1.7 Formalisation temps réel

Cependant, les diagrammes formels ne sont pas obtenus en un seul bloc, mais en temps réel, c'est-à-dire au fur et à mesure de l'acquisition du signal dans MORFOS

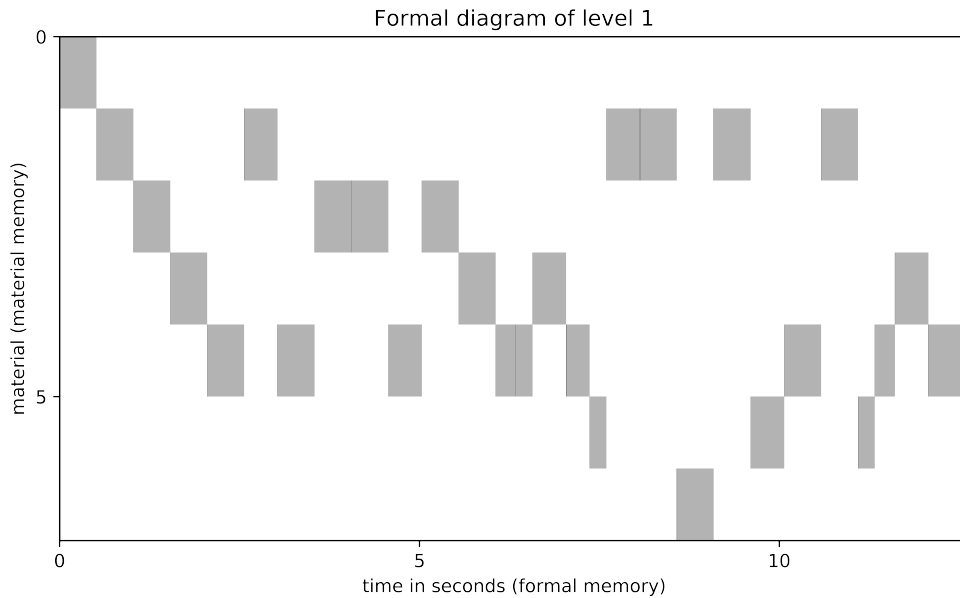


FIGURE 2.7 – Diagramme formel final automatisé du *Geisslerlied*. Le premier matériau correspond au silence qui débute le fichier audio.

et de la représentation initiale au niveau de l'Algorithme MORFOS. Cela signifie que les trois représentations r , s et f co-existent en réalité dès l'acquisition des premiers objets dans le logiciel.

Il y a donc initialement les trois représentations r , s et f qui correspondent initialement chacune au mot vide. Un indicateur k est également défini et correspond à l'indice de la lettre où a eu lieu la dernière segmentation. Ils sont représentés sous la forme du quadruplet (r, s, f, k) , initialisé par $(\epsilon, \epsilon, \epsilon, 0)$. Nous allons ainsi présenter une description de l'Algorithme Cognitif par une suite d'états. Ce quadruplet peut changer d'état en fonction de trois opérations distinctes :

- l'*acquisition temps réel* notée atr ;
- la *segmentation temps réel* notée str ;
- la *classification temps réel* notée ctr .

Lors de son acquisition, un nouvel objet x provenant du flux est concaténé à r :

$$(r, s, f, k) \xrightarrow[x]{atr} (r.x, s, f, k)$$

Où "." est l'opération de concaténation entre deux mots (lorsque celle-ci est ambiguë), ou bien entre un mot et une lettre.

Si cela est pertinent, une opération de segmentation peut suivre l'acquisition.

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

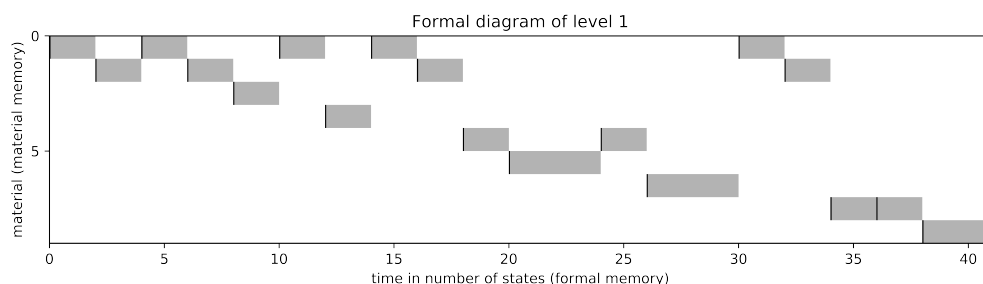


FIGURE 2.8 – Diagramme formel final automatisé du Rondo K545 de W.A. Mozart.

Comme le nouvel objet acquis peut concerner le retour au début d'un nouveau groupe, la segmentation est toujours opérée entre l'objet nouvellement acquis et le précédent.

Lors de la segmentation, s est concaténé au fragment constitué de l'ensemble des lettres acquises depuis la dernière segmentation (indiquée par l'indice correspondant à la lettre k) jusqu'à la lettre précédemment acquise d'indice $n - 1$:

$$(r, s, f, k) \xrightarrow{str} (r, s.([r]_{k,n-1}), f, n)$$

Où $[r]_{k,n-1}$ correspond au sous mot de r allant de l'indice k à l'indice $n - 1$ inclus.

Après chaque segmentation, une opération de classification ctr est appliquée (après une éventuelle transformation) sur le dernier facteur concaténé à s . Le label correspondant obtenu est ensuite concaténé dans \mathcal{B} à f . Si on note w le dernier fragment de la représentation segmentée, on a alors :

$$(r, s.w, f, k) \xrightarrow{ctr} (r, s.w, f.ctr(w), k)$$

L'acquisition peut être suivie par une acquisition ou une segmentation, tandis que la segmentation est nécessairement suivie par une classification, et la classification est nécessairement suivie par une acquisition.

Tous les changements d'état sont procédés lorsque l'algorithme a effectué l'opération d'acquisition sur l'intégralité de l'information initiale. Alors la dernière acquisition est nécessairement suivie d'une segmentation et d'une classification qui terminent l'algorithme.

Ainsi après avoir effectué l'intégralité des changements d'états dans le quadruplet, f contient la représentation finale équivalente au diagramme formel final.

On présente la succession des opérations dans l'automate en figure 2.9.

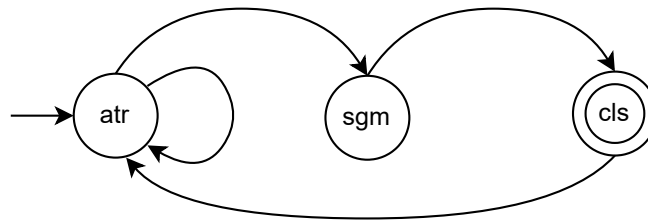


FIGURE 2.9 – Automate des opérations de l’Algorithme MORFOS. Chaque état correspond à une opération effectuée *atr*, *sgm* ou *cls*.

2.1.8 Pré-traitement

On constate, autant pour l’acquisition à partir d’un fichier audio qu’à partir d’une représentation symbolique, que l’étape permettant d’obtenir la première représentation r consiste en réalité déjà en une étape de segmentation, voire également de similarité. Seulement, celles-ci sont gérées de manière distincte de l’Algorithme MORFOS tel que nous le présentons en tant qu’algorithme automatisé.

En effet, concernant le fichier audio, il y a une première segmentation à l’échelle de la fenêtre qui est arbitraire, et si il y a bien des objets cohérents pour effectuer un calcul de similarité, il s’agit d’objets à une échelle si petite et dont la segmentation est trop arbitraire pour que ces objets aient véritablement un sens en tant que paradigme.

Concernant la représentation symbolique, l’obtention de celle-ci nécessite une pré-analyse, soit par un logiciel, soit par un musicologue, à partir d’un fichier audio ou d’une représentation diverse de l’œuvre musicale. L’étude de la similarité et de la segmentation a été opérée par exemple par un musicologue dans le cadre de l’analyse du Rondo du Menuet K545 de W.A. Mozart.

La figure 2.10 représente ainsi les pré-traitements effectués avant la boucle de l’Algorithme MORFOS. Le pré-traitement est externe à MORFOS pour une entrée symbolique, et une phase préliminaire de MORFOS pour une entrée audio. Dans les deux cas, les calculs de segmentation et de similarité sont potentiellement faits simultanément.

Ce module de pré-analyse peut néanmoins tout à fait être considéré comme une instance différenciée de l’Algorithme MORFOS. Il permet en effet à partir d’une première représentation qui est le fichier audio ou la partition, d’obtenir après un calcul de segmentation et de similarité la représentation initiale r .

Ainsi, les représentations initiales r ont elles-mêmes une équivalence avec un

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

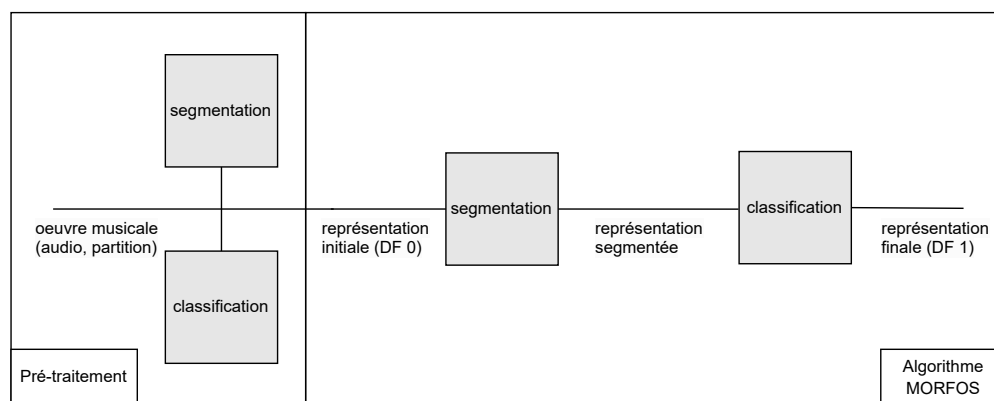


FIGURE 2.10 – Représentation du logiciel MORFOS constitué d'une phase de pré-traitement et de l'Algorithme MORFOS.

diagramme formel initial. La figure 2.11 présente le diagramme formel initial associé à la représentation initiale du *Geisslerlied*, obtenue en suivant le procédé présenté précédemment à partir d'un fichier audio.

Notons que le diagramme formel initial du *Geisslerlied* fait doublon avec le *diagramme formel final* : les subdivisions en *frames* n'ont quasiment pour seul intérêt que d'apporter une information plus détaillée de la dynamique mais le diagramme général obtenu est très similaire au diagramme de niveau 1, à la différence que chaque objet du niveau 1 est subdivisé en une quarantaine d'objets au niveau 0.

En effet, la segmentation est effective lorsque qu'un objet est considéré différent de l'objet qui le précède, ce qui signifie qu'à chaque fois qu'il n'y a pas segmentation, il y a alors similarité entre chacun des objets qui se succèdent. Cependant, bien qu'il y ait similarité entre chacun des objets successifs, la similarité relative à tous les objets entendus précédemment n'est pas calculée. Pour avoir le label correspondant, il faut soit comparer les objets entre eux, soit attendre le calcul de similarité après obtention de la représentation segmentée s . Ce label n'est cependant pas nécessaire pour le calcul de la segmentation dans l'Algorithme MORFOS car la segmentation est effectuée à partir de l'information du signal. Le calcul de similarité au niveau initial est donc redondant avec la segmentation et n'est pas nécessaire.

La représentation automatisée du diagramme formel initial du *Rondo* obtenue à partir de la représentation initiale symbolique extraite du *Rondo K.545* de W.A. Mozart est également donnée en figure 2.12.

Ici, le diagramme formel initial du *Rondo* a un intérêt sémantique bien que le

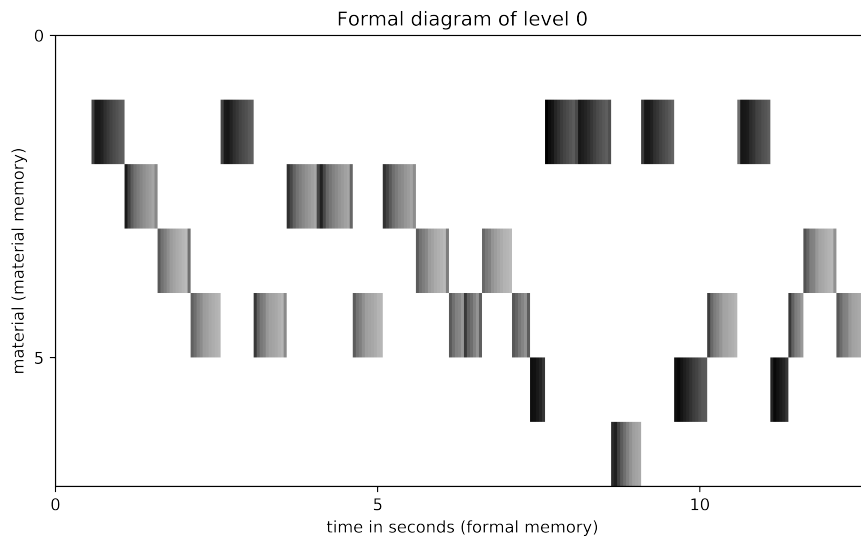


FIGURE 2.11 – Diagramme formel initial automatisé du *Geisslerlied* obtenu à partir d’un fichier MIDI synthétisé avec un son de piano à 120bpm. Chaque objet est une frame de 0,023s : chaque bloc d’une seconde est donc en réalité divisé en une quarantaine de sous-blocs. Les nuances de gris correspondent ici à l’information de la dynamique (plus le bloc est foncé, plus la dynamique est élevée). Comme pour le diagramme formel final présenté précédemment, un premier matériau correspond au silence qui débute le fichier audio : la dynamique est nulle et les objets sont donc blancs.

processus de similarité et de segmentation soit externe au logiciel.

La figure 2.13 présente les deux modélisations possibles en fonction du traitement audio ou symbolique de l’œuvre musicale. Si la partie de l’Algorithme MORFOS semble quasiment identique, nous avons déjà pu constater que les contraintes qui induisent les tests de similarité et de segmentation en fonction du modèle diffèrent. Mais la plus grande différence apparaît dans la possibilité au niveau du modèle du signal d’acquérir à la fois une information audio et symbolique, tandis que le deuxième modèle est uniquement symbolique.

2.1.9 Prise en compte des vecteurs en entrée de MORFOS

En réalité, nous ne considérons pas seulement deux mais quatre types de représentations pour une œuvre musicale que notre logiciel peut prendre en entrée :
 — les représentations audio (signal numérisé) ;

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

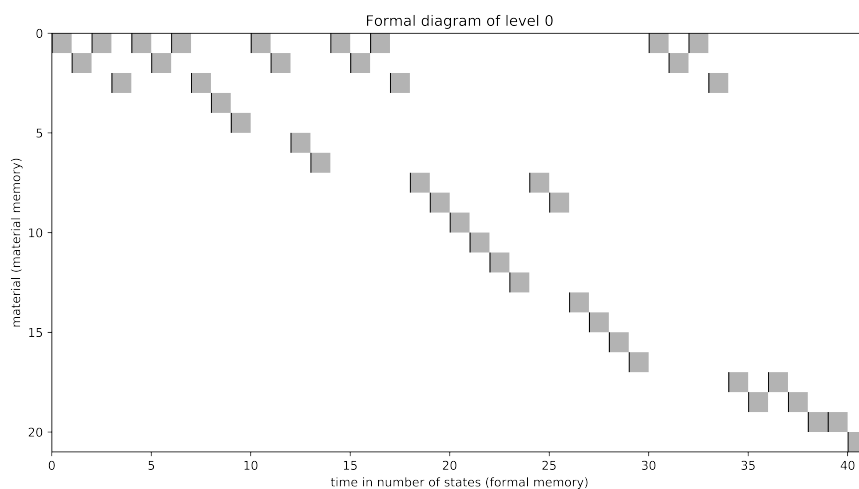


FIGURE 2.12 – Diagramme formel initial automatisé du Rondo K545 de W.A. Mozart.

- les représentations symboliques ordonnées (alphabet) ;
- **les représentations symboliques non ordonnées (partition diverse) ;**
- **les représentations vectorielles (descripteurs).**

Les représentations symboliques non ordonnées sont une succession de symboles obtenus via un logiciel externe tel que l'annotation proposée par F. Bimbot dans le cadre du projet Quaero [16]. Les labels sont ici obtenus par le calcul d'une similarité *absolue* par rapport à un dictionnaire externe pour l'obtention de la représentation initiale symbolique. Il peut également s'agir d'une partition représentée par une succession de caractères. Il faut alors réordonner les caractères avant de fournir la représentation à l'algorithme MORFOS.

Les représentations vectorielles sont quant à elles des successions de vecteurs obtenues par l'utilisation de logiciels externes permettant par exemple d'obtenir une représentation de l'œuvre sous forme de descripteurs audio.

Dans les deux cas, la segmentation est considérée comme déjà effectuée afin d'avoir un contenu (descripteur ou label) significativement cohérent.

Ainsi, ces représentations sont en réalité des représentations segmentées qui sont introduites à ce niveau là de l'algorithme. Les vecteurs, en tant que descripteurs, ont les mêmes règles de similarité que les représentations segmentées obtenues à partir du fichier audio, et les labels non ordonnés ont les mêmes règles de similarité que les représentations segmentées symboliques.

On présente l'introduction de ces représentations au niveau du logiciel en fi-

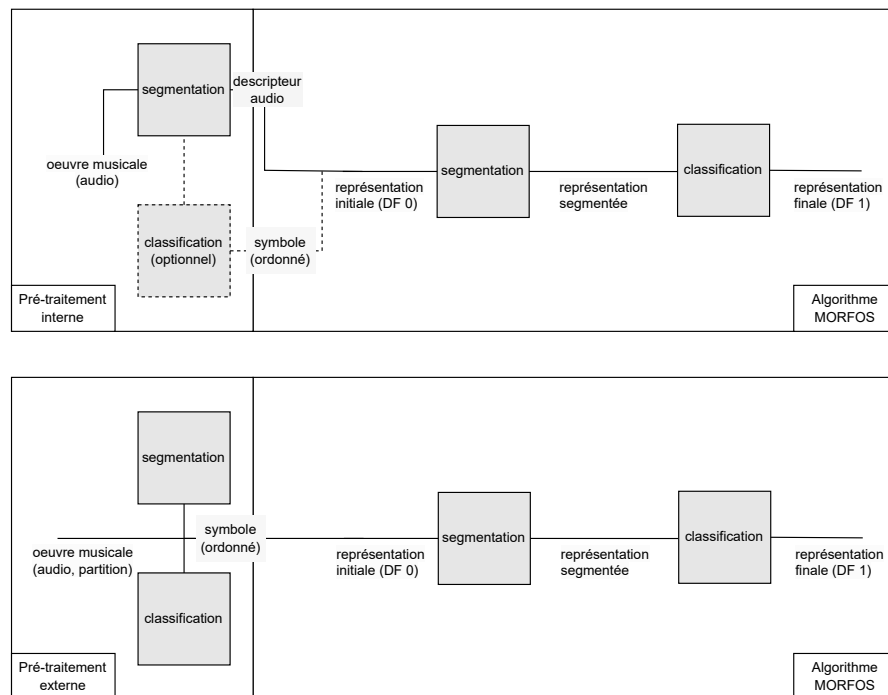


FIGURE 2.13 – Représentation de MORFOS : une phase de pré-traitement interne permet de fournir un ensemble de descripteurs audio et éventuellement une représentation symbolique à partir d'un fichier audio (diagramme du haut), et d'une phase de pré-traitement externe fournissant une représentation symbolique à partir d'une représentation audio ou symbolique (diagramme du bas). Ces représentations initiales sont ensuite fournies à l'Algorithme MORFOS dont les contraintes de segmentation et de classification divergent en fonction du format vectoriel ou symbolique.

gure 2.14.

2.1.10 Polyphonie et multiplicité des échelles de temps

Les mécanismes de traitement décrits ci-dessus prennent en entrée un flot d'informations de telle sorte que la fin de l'objet précédent marque le début de l'objet suivant. Ils ne permettent donc pas une analyse polyphonique au sens contrapuntique. Ceci est justifié par le fait que nous avons choisi lors de l'automatisation de l'algorithme de laisser de côté le problème de la séparation des voix pour nous concentrer sur celui de la hiérarchisation.

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

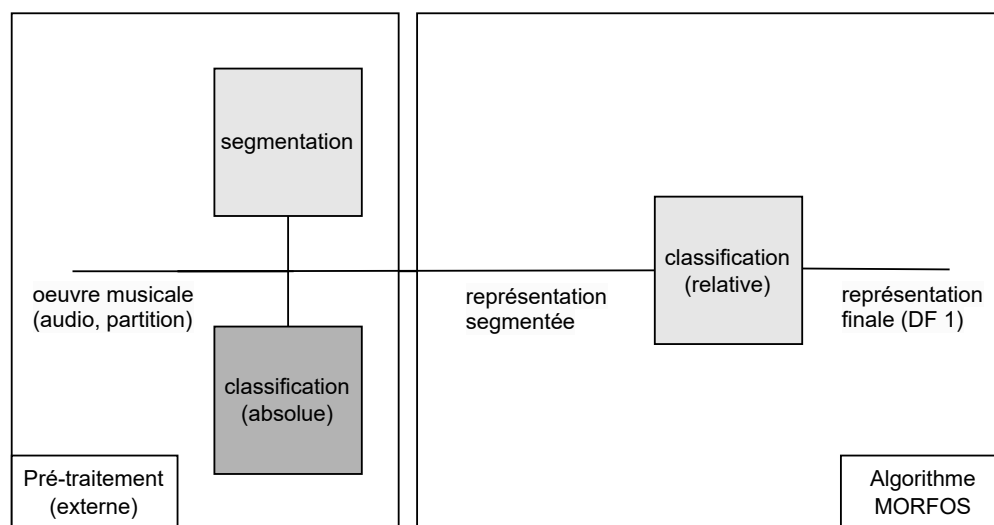


FIGURE 2.14 – Organisation de MORFOS pour les représentations vectorielles. Celles-ci sont introduites en tant que représentation segmentée et sont soumises directement à l'opération de classification.

Les objets peuvent être de différentes échelles temporelles : du microscopique (de l'ordre de la milliseconde) au macroscopique (de l'ordre de la partie musicale, voir de l'œuvre musicale entière.)

L'algorithme produit à priori un diagramme formel en fonction de matériaux pour une dimension et pour une échelle temporelle donnée. Ainsi, il est possible d'obtenir un diagramme formel à l'échelle motivique de matériaux mélodiques, ou encore un diagramme formel à l'échelle de la phrase de matériaux rythmiques.

2.2 L'Algorithme MORFOS : Hiérarchisation

Nous avons montré comment obtenir un diagramme formel à partir d'une représentation donnée sur un alphabet ordonné, à une échelle de temps précise. Remarquons cependant (voir chapitre 1) que le cerveau humain est capable de structurer le temps à différentes échelles : en effet lorsque nous écoutons une œuvre musicale, nous pouvons proposer une segmentation autant à l'échelle du motif qu'à l'échelle de la partie musicale [51]. Cependant, alors que le cerveau humain ne peut pas retenir l'intégralité des motifs qui constituent une œuvre de plusieurs minutes, un avantage d'un système informatique est que sa mémoire est presque infallible. Ainsi, un ensemble de diagrammes formels peuvent être produits en simultanés,

au fur-et-à-mesure de l'acquisition du signal.

2.2.1 Hiérarchisation du modèle

Ainsi, il est possible de proposer, à partir du processus de création d'un diagramme formel, une représentation hiérarchisée des diagrammes formels appelée *diagramme formel multi-échelle*.

À partir des diagrammes d'échelles temporelles les plus petites, l'algorithme construit les diagrammes d'échelles temporelles les plus grandes, de telle sorte que les ensembles d'objets de niveau inférieur constituent un unique objet de niveau supérieur. Par exemple, un ensemble de notes d'un diagramme formel constitue une phrase pour le diagramme formel d'échelle supérieure, et un ensemble de phrases constitue une partie du diagramme formel d'échelle supérieure.

2.2.2 Formalisation

Remarquons tout d'abord que l'entrée et la sortie à un niveau donné de l'Algorithme MORFOS sont en réalité la même structure de données. L'entrée d'une instance de l'algorithme MORFOS peut ainsi être la sortie d'une instance de niveau inférieur. On ne parle alors plus de *représentation initiale* et de *représentation finale* mais de *représentation de niveau n* et de *représentation de niveau $n + 1$* . La représentation segmentée est appelée *représentation segmentée de niveau n* .

On définit ainsi $r^{(n)}$ notre *représentation de niveau n* équivalente au *diagramme formel de niveau n* comme un mot

$$r^{(n)} = x_1^{(n)} \dots x_i^{(n)} \dots x_{l_n}^{(n)}$$

avec $x_i^{(n)}$ des lettres définies sur l'alphabet ordonné $\mathcal{A}^{(n)}$.

Cette représentation de niveau n est segmentée en un ensemble de fragments. L'opération de segmentation $sgm(r^{(n)}, cs(n))$ est définie et prend en arguments la représentation initiale $r^{(n)}$ et un ensemble de critères de segmentation cs fonction du niveau n : en effet, les critères de segmentation peuvent varier d'une échelle de temps à une autre et donc d'un niveau à un autre. Cela donne alors $s^{(n)}$ une *représentation segmentée de niveau n* qui est le regroupement de lettres de $\mathcal{A}^{(n)}$ en **facteur** $w_j^{(n)}$ appartenant à $\mathcal{A}^{(n)*}$ par associativité du monoïde \mathcal{A} sur $\mathcal{A}^{(n)*}$ tel que

$$s^{(n)} = sgm(r^{(n)}, cs(n)) = w_1^{(n)} \dots w_j^{(n)} \dots w_{l_{n+1}}^{(n)}$$

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

On calcule ensuite la similarité interne entre les fragments $w_j^{(n)}$ de $s^{(n)}$. Pour cela, on définit une opération de classification $cls(s^{(n)}, cc(n))$ qui prend en arguments la représentation segmentée de niveau n $s^{(n)}$ et un ensemble de critères de classification cc dépendants du niveau n . Ainsi, la *représentation de niveau $n+1$* est le mot $r^{(n+1)}$ obtenu telle que

$$r^{(n+1)} = cls(s^{(n)}, cc(n)) = cls(sgm(r^{(n)}, cs(n)), cc(n)) = x_1^{(n+1)} \dots x_j^{(n+1)} \dots x_{l_{n+1}}^{(n+1)}$$

où les $x_j^{(n+1)}$ sont des lettres définies sur un nouvel alphabet ordonné $\mathcal{A}^{(n+1)}$.

Ainsi, un diagramme formel multi-échelle est obtenu à partir d'une représentation en entrée, puis la segmentation et la classification successives des représentations obtenues. La figure 2.15 représente l'organisation de MORFOS, permettant d'obtenir l'ensemble des diagrammes formels hiérarchisés. L'algorithme termine lorsque l'ensemble des objets de niveau 0 sont traités. Le diagramme formel multi-échelle est l'ensemble qui contient tous les diagrammes formels $dfme = r^{(j)}_{j \in [0, n_{max}]}$ du niveau 0 à n_{max} .

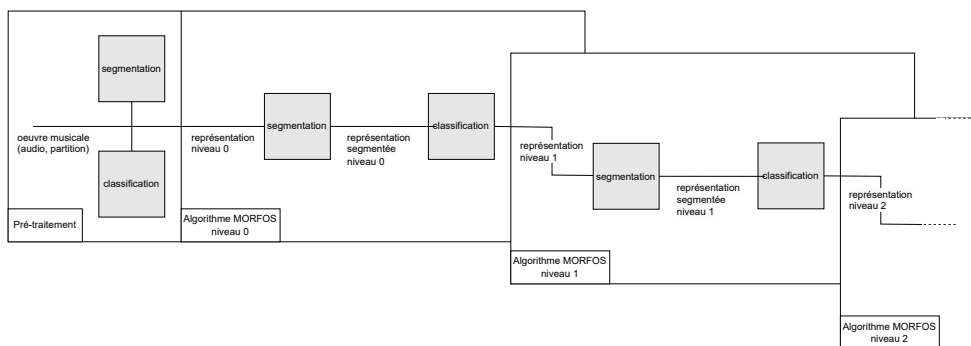


FIGURE 2.15 – Organisation de MORFOS : le logiciel est constitué de la phase de pré-traitement ainsi que de l'Algorithme MORFOS itéré à plusieurs niveaux.

2.2.3 Exemple du *Rondo* extrait du *Menuet K545* de W.A. Mozart

Prenons un exemple avec le *Rondo* de Mozart. On a présenté dans la partie précédente la représentation de niveau 0 et la représentation de niveau 1 obtenue à partir de la segmentation et classification de celle de niveau 0. On note $\mathcal{A}^{(n)}$ l'alphabet des labels associés aux matériaux de la représentation de niveau n .

On avait ainsi

$$r^{(0)} = x_1^{(0)} \dots x_i^{(0)} \dots x_{41}^{(0)} = abacabacdeabfgabachijklmhinopqabacrsrsttu$$

avec l'ensemble des $x_i^{(0)}$ des lettres définies sur $\mathcal{A}^{(0)}$

Par application des fonctions de segmentation $sgm(r^{(0)}, cs(0))$ sur $r^{(0)}$ et de classification $cls(s^{(0)}, cc(0))$ sur

$$\begin{aligned} s^{(0)} &= sgm(r^{(0)}, cs(0)) \\ &= (ab)(ac)(ab)(ac)(de)(ab)(fg)(ab)(ac)(hi)(jklm)(hi)(nopq)(ab)(ac)(rs)(rs)(ttu) \\ &= w_1^{(0)} \dots w_j^{(0)} \dots w_{18}^{(0)} \end{aligned}$$

avec l'ensemble des $w_j^{(0)}$ des facteurs définis sur $\mathcal{A}^{(0)*}$ on obtient

$$r^{(1)} = cls(s^{(0)}, cc(0)) = x_1^{(1)} \dots x_j^{(1)} \dots x_{18}^{(1)} = ABABCADABEFEGABHHI$$

avec l'ensemble des $x_j^{(1)}$ des lettres définies sur $\mathcal{A}^{(1)}$, donc $A \in \mathcal{A}^{(1)}$ et $A \in \mathcal{A}^{(0)}$ et $A \neq a$.

L'algorithme effectue ainsi le même procédé sur $r^{(1)}$.

Par application de la fonction de segmentation $sgm(r^{(1)}, cs(1))$ sur $r^{(1)}$, cela donne :

$$s^{(1)} = sgm(r^{(1)}, cs(1)) = (AB)(AB)(CAD)(AB)(EF)(EG)(AB)(HHI) = w_1^{(1)} \dots w_j^{(1)} \dots w_8^{(1)}$$

avec l'ensemble des $w_j^{(1)}$ des facteurs définis sur $\mathcal{A}^{(1)*}$. La fonction de classification $cls(s^{(1)}, cc(1))$ est appliquée ensuite sur $s^{(1)}$, ce qui permet d'obtenir :

$$r^{(2)} = cls(s^{(1)}, cc(1)) = x_1^{(2)} \dots x_j^{(2)} \dots x_8^{(2)} = AABACDAE$$

Enfin, la fonction de segmentation $sgm(r^{(2)}, cs(2))$ est appliquée sur $r^{(2)}$:

$$s^{(2)} = sgm(r^{(2)}, cs(2)) = (AAB)(ACD)(AE) = w_1^{(2)} w_2^{(2)} w_3^{(2)}$$

avec l'ensemble des $w_j^{(2)}$ des facteurs définies sur $\mathcal{A}^{(2)*}$. La fonction de classification $cls(s^{(2)}, cc(2))$ est enfin appliquée :

$$r^{(3)} = cls(s^{(2)}, cc(2)) = x_1^{(3)} x_3^{(3)} x_3^{(3)} = ABC$$

Nous n'avons aucune segmentation avant la fin du mot $r^{(3)}$, le procédé est donc complet. Les diagrammes formels correspondants sont présentés en figure 2.16.

2.2.4 Formalisation temps réel

Comme pour l'obtention d'un unique diagramme formel, la création de l'ensemble des diagrammes formels aux différentes échelles se fait en réalité en temps

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

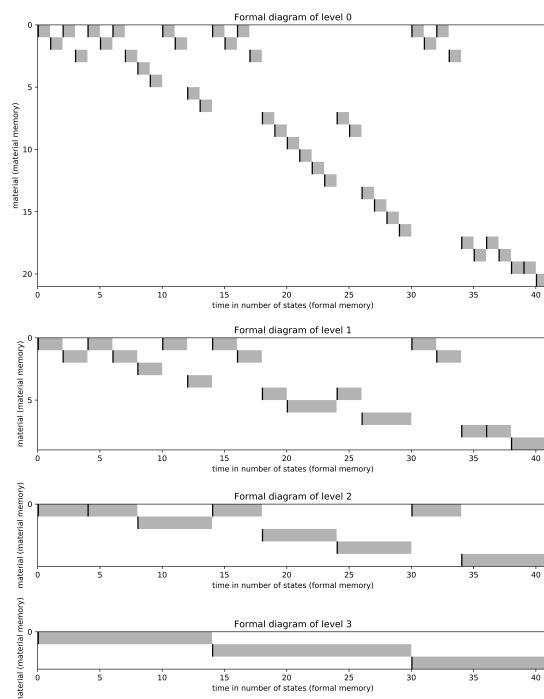


FIGURE 2.16 – Diagramme formel multi-échelle du *Rondo K545* de W.A. Mozart.

réel au fur et à mesure de l'acquisition du signal initial, et en simultanément, comme pour une écoute réelle.

Ainsi, un diagramme formel multi-échelle est défini par une suite d'états représentés par le mot $R = X_0.X_1...X_n$ avec n la taille du mot correspondant au nombre de diagrammes formels créés. Il est obtenu par la concaténation des triplets $X_j = (r^{(j)}, s^{(j)}, k^{(j)})$ où $r^{(j)}$ est la représentation de niveau j , $s^{(j)}$ est la représentation segmentée de niveau j et $k^{(j)}$ le marqueur de l'indice de la dernière segmentation au niveau j . Il n'y a pas la représentation de niveau $j + 1$ car celle-ci est contenue dans X_{j+1} .

Le Diagramme Formel Multi-échelle est donc le mot obtenu à partir de la concaténation des $\{r^{(j)}\}_{0 \leq j \leq n}$ à l'état final de R avec n le nombre total de niveau créés.

Dans un premier temps, $R = \epsilon$ est initialisé avec le mot vide. On a quatre opérations qui permettent de changer l'état de R :

- la *création d'un nouveau niveau* $X_n = (\epsilon, \epsilon, 0)$ avec l'opération cnn ;
- l'*acquisition temps réel* au niveau 0 notée atr ;
- la *segmentation temps réel* au niveau n $str(n)$;

— la *classification temps réel* au niveau n $ctr(n)$ vers le niveau $n + 1$.

Non seulement les opérations de segmentation $str(n)$ et de classification $ctr(n)$ s'appliquent au niveau n , mais elles peuvent également prendre en compte des critères dépendant du niveau auquel elles s'appliquent.

On garde par ailleurs les notations "." pour la concaténation entre deux mots lorsqu'il y a ambiguïté ou entre un mot et une lettre, et $[r^{(j)}]_{k_1, k_2}$ pour le sous mot de r allant de l'indice k_1 à l'indice k_2 inclus.

Lors de l'acquisition d'un nouvel objet x au niveau 0, on concatène $r^{(0)}$ avec le nouvel objet x qui provient du flux :

$$X_0 X_1 \dots X_n = (r^{(0)}, s^{(0)}, k^{(0)}) X_1 \dots X_n \xrightarrow[x]{atr} (r^{(0)}.x, s^{(0)}, k^{(0)}) . X_1 \dots X_n$$

Après l'acquisition, on peut effectuer si cela est pertinent une opération de segmentation. Comme précédemment, la segmentation est toujours opérée entre l'objet nouvellement acquis et le précédent.

Lors de la segmentation, on concatène $s^{(j)}$ avec le fragment constitué de l'ensemble des lettres acquises depuis la dernière segmentation (indiquée par l'indice correspondant à la lettre $k^{(j)}$) jusqu'à la lettre précédemment acquise d'indice $l - 1$ avec l la longueur du mot de la représentation $r^{(j)}$:

$$X_0 \dots (r^{(j)}, s^{(j)}, k^{(j)}) \dots X_n \xrightarrow{str^{(j)}} X_0 \dots (r^{(j)}, s^{(j)} . ([r^{(j)}]_{k, l-1}), l) \dots X_n$$

Après chaque segmentation, on applique (après une éventuelle transformation) une opération de classification ctr sur le dernier facteur concaténé à $s^{(j)}$. On concatène ensuite le label correspondant obtenu dans $\mathcal{A}^{(j)}$ à $r^{(j+1)}$. Si on note w le dernier fragment de notre représentation segmentée, on a alors :

$$X_0 \dots (r^{(j)}, s^{(j)} . w, k^{(j)}) (r^{(j+1)}, s^{(j+1)}, k^{(j+1)}) \dots X_n \\ \xrightarrow{ctr^{(j)}} X_0 \dots (r^{(j)}, s^{(j)} . w, k^{(j)}) . (r^{(j+1)} . ctr(w), k^{(j+1)}) \dots X_n$$

On constate que l'opération d'acquisition atr est en réalité un cas particulier de l'opération de classification $cls(n)$ au niveau 0 avec la fonction de classification étant la fonction identité.

Il est possible que le niveau $j + 1$ n'existe pas encore. Avant de procéder à l'opération de classification, il faut alors créer un nouveau niveau cnn qui correspond à la concaténation du diagramme formel multi-échelle de taille $n - 1$ avec le diagramme formel $X_n = (\epsilon, \epsilon, 0)$.

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle :
formalisation de MORFOS

$$X_0 X_1 \dots X_{n-1} \xrightarrow{cnn} X_0 X_1 \dots X_{n-1} . (\epsilon, \epsilon, 0)$$

L'état final du mot est obtenu lorsque l'intégralité de l'information au niveau 0 est acquise et que la classification a été effectuée au niveau le plus haut.

Ainsi, l'ensemble des n diagrammes formels qui constituent le diagramme formel multi-échelle sont les $r^{(j)}_{j \in [0, n]}$ après avoir procédé à l'intégralité des changements d'états.

L'ordre des opérations est représenté sur le schéma en figure 2.17. L'algorithme commence toujours par une succession d'acquisitions au niveau 0, avant de segmenter. La segmentation est toujours suivie par une classification, avec une création intermédiaire d'un nouveau niveau si celui-ci n'existe pas. Après la classification, il y a soit segmentation au niveau supérieur, soit acquisition au niveau 0.

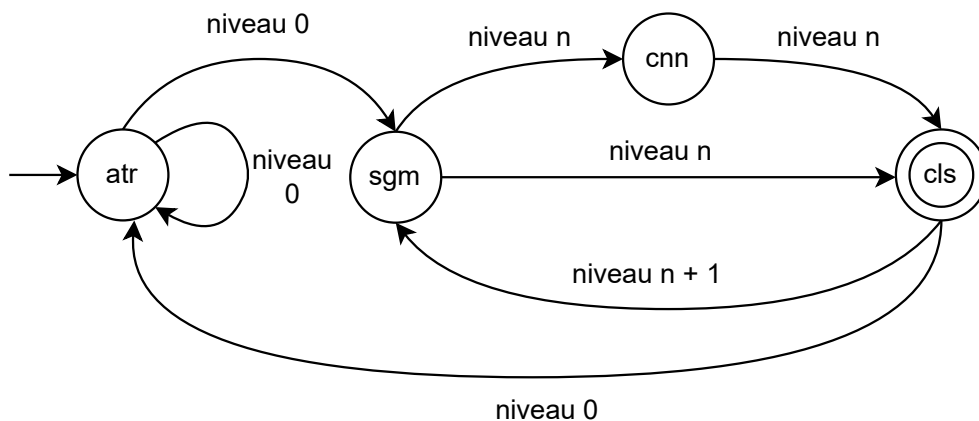


FIGURE 2.17 – Automate des opérations de l'Algorithme MORFOS. Chaque état correspond à une opération et les arcs correspondent au niveau de l'état affecté par l'opération vers laquelle la flèche pointe.

Ainsi, sur l'exemple du *Rondo* de W.A. Mozart, nous avons les changements d'états suivants :

Le diagramme formel multi-échelle est initialement représenté par le mot vide ϵ . On souhaite acquérir un nouvel objet au niveau 0. Pour cela, le niveau 0 doit d'abord être créé avec l'opération cnn :

$$\epsilon \xrightarrow{cnn} X_0$$

Un nouvel objet a est ensuite acquis au niveau 0 :

$$(r^{(0)}, s^{(0)}, k^{(0)}) = (\epsilon, \epsilon, 0) \xrightarrow[a]{atr} (r^{(0)}.a, s^{(0)}, k^{(0)}) = (a, \epsilon, 0)$$

Il n'y a pas lieu de segmenter, l'objet suivant, b , est donc acquis :

$$(a, \epsilon, 0) \xrightarrow[b]{atr} (ab, \epsilon, 0)$$

De la même manière, l'algorithme continue d'acquérir l'objet suivant, a :

$$(ab, \epsilon, 0) \xrightarrow[a]{atr} (aba, \epsilon, 0)$$

Ici, une opération de segmentation est effectuée entre b et a :

$$(aba, \epsilon, 0) \xrightarrow{sgm(0)} (aba, (ab), 3)$$

Afin de procéder à la classification de (ab) au niveau supérieur, il faut avant tout initialiser le niveau supérieur :

$$(aba, (ab), 3) \xrightarrow{cnn} (aba, (ab), 3).(\epsilon, \epsilon, 0)$$

L'opération de classification est ensuite opérée sur (ab) qui est concaténé à $r^{(1)}$:

$$(aba, (ab), 3).(\epsilon, \epsilon, 0) \xrightarrow{cls(0)} (aba, (ab), 3).(A, \epsilon, 0)$$

L'exemple complet est présenté en annexe [A](#).

Ainsi, MORFOS permet de représenter de manière récursive un ensemble de diagrammes formels sur une dimension donnée. Cela permet alors de comparer la forme au sein d'une même œuvre musicale, et de mieux comprendre l'articulation entre les différentes échelles temporelles.

2.2.5 Échelles musicales

Un musicologue pourra ainsi comparer des diagrammes formels aux différents niveaux d'une même œuvre, tel que nous l'avons fait précédemment pour le *Geisslerlied*, ou bien par exemple des œuvres qui ont la même forme à une échelle élevée de la partie musicale (tel que deux rondo), mais tout à fait distinctes à l'échelle motivique.

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

Une question, qui sera traitée plus amplement dans la partie 2.4.2 de ce chapitre, est de déterminer ce qui fait entité, de ce qui fait variation dans la forme à une échelle donnée. Ainsi, si le procédé semble être le même à toutes les échelles temporelles, il n'est pas sûr que les opérations, notamment lors des tests de similarité et de segmentation, soient en réalité les mêmes à toutes les échelles de temps.

Snyder, dans [151] présente une vingtaine de niveaux d'expériences musicales dont les événements vont du sixième de secondes à l'échelle de l'heure. Ces niveaux sont divisés en trois groupes : l'*Event Fusion* (inférieur au vingtième de seconde), les *Melodic and Rhythmic Grouping* (jusqu'aux 8 secondes environs) et la *Form* (de la dizaine de seconde à l'heure). Ces groupes peuvent également être définis comme *microscopiques*, *mésoscopiques* et *macroscopiques*. L'échelle microscopique décrit les détails de l'œuvre, tandis que plus l'échelle s'approche du macroscopique, plus les éléments intermédiaires disparaissent pour permettre une lecture de la structure profonde de l'œuvre, à la manière de l'*Ursatz* de Schenker. Y. Langner dans [84] procède à une opération de floutage du signal par moyennage de la dynamique à différentes échelles afin de mieux déterminer la structure de l'œuvre musicale.

Ainsi la relation entre les échelles et donc la hiérarchisation permet de mieux comprendre la structure de l'œuvre. Par ailleurs, on constate que la notion de périodicité est fondamentale dans la notion de structure à toutes les échelles de temps, de la fréquence du signal (à l'échelle microscopique) à l'ostinato par exemple (à l'échelle mésoscopique) jusqu'au retour du silence qui sépare deux œuvres musicales (à l'échelle macroscopique). Le retour à un événement déjà vu est donc un aspect prépondérant des critères de segmentation qui permet de faire le lien entre un niveau inférieur et un niveau supérieur et donc de participer à la représentation de la structure.

Cependant, MORFOS ne représente pas l'intégralité des niveaux présentés par Snyder : dans le cadre de la musique de notes, l'algorithme présente un unique niveau à l'échelle microscopique qui est le découpage en *frames* de l'ordre du quarantième de secondes, tandis que le deuxième niveau correspond au niveau structurel de l'ordre de la seconde et donc à l'échelle mésoscopique. Par ailleurs, il n'existe pas nécessairement des motifs pertinents à tous les niveaux en simultané et nous représentons seulement les niveaux où sont déterminés des matériaux pertinents, à partir des matériaux représentés aux niveaux inférieurs. Néanmoins, il est possible qu'il existe en réalité des matériaux à des niveaux "intermédiaires", c'est-à-dire qu'il existe un ou plusieurs matériaux qui font sens à un niveau donné sans pouvoir créer une représentation de ce niveau tel qu'il y ait une succession d'objets cohérents et ainsi une représentation continue de la structure. Notre logiciel ne

crée alors pas de représentation intermédiaire et ce matériau est assigné au niveau inférieur ou supérieur.

Néanmoins, nous précisons que le logiciel que nous avons présenté ne peut pas prétendre modéliser les effets concrets du système cognitif lors de l'écoute musicale.

2.3 Les trois plans

Le diagramme formel contient deux axes qui sont *l'axe des matériaux* et *l'axe du temps*. En l'étendant aux diagrammes formels multi-échelles, nous distinguons un troisième axe qui est celui de la hiérarchisation qui correspond à un *axe des niveaux hiérarchiques*.

Ainsi, la forme peut déjà être observée à partir du diagramme formel, mais il est possible également de regarder la relation des matériaux à travers les différents niveaux, ou encore la relation des objets dans le temps sur les différents niveaux. On distingue ainsi trois plans, qui sont :

- le *plan de la forme*, les diagrammes formels étant une *coupe* à un niveau de structure donné ;
- le *plan de la structure*, qui exprime les relation *syntagmatiques* ;
- le *plan de l'organisation*, qui explicite le système *hors-temps*.

La figure 2.18 représente un bloc en 3 dimensions (droite) obtenu à partir de la superposition des différents niveaux du diagramme formel multi-échelle (gauche). Cela permet de distinguer les 3 axes des matériaux, du temps et des niveaux, ainsi que les trois plans qui en découlent. Par souci de lisibilité, les objets des trois différents niveaux sont représentés de trois couleurs différentes : bleu pour le niveau 0, rouge pour le niveau 1 et vert pour le niveau 2. Nous gardons ce code couleur pour les diagrammes de cette section.

2.3.1 Présentation des plans

Le plan de la forme correspond au plan obtenu à partir des axes des matériaux et du temps. C'est celui du diagramme formel à différents niveaux hiérarchiques et il permet d'étudier l'évolution du langage musical dans le temps et de déduire de l'utilisation des matériaux les effets produits à l'écoute.

Le plan de la structure correspond à la projection obtenue à partir des axes du temps et des niveaux. Il permet de déterminer l'organisation temporelle entre

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle :
formalisation de MORFOS

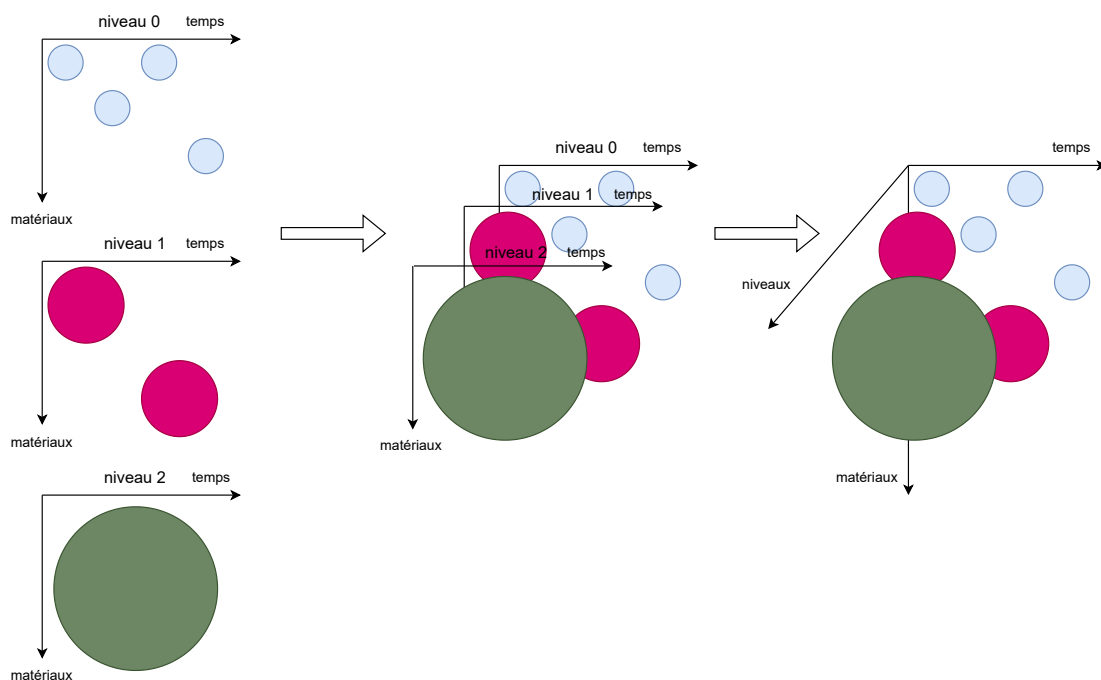


FIGURE 2.18 – Superposition des diagrammes formels dans le diagramme formel multi-échelle.

les différentes échelles de temps à différentes coupes de matériaux, et donc proportionnellement aux durées des éléments considérés.

Le plan de l'organisation correspond au plan obtenu à partir des axes des matériaux et des niveaux. Il décrit le dictionnaire du langage utilisé dans la pièce musicale et la manière dont il s'articule entre les différents niveaux de la structure à chaque instant. Il peut être interprété comme une empreinte de la pièce musicale hors-temps, comme son *système*.

La figure 2.19 présente un schéma des plans de la forme, de la structure et de l'organisation. Le plan de la forme correspond à la description de la section précédente. Le plan de la structure est représenté de telle sorte que les objets de niveau n apparaissant à l'instant t sont associés à tous les objets de niveaux $n - 1$ contenus entre l'instant $t - k$ exclu, avec $0 < k \leq t$ instant d'apparition du précédent objet de niveau n (ou de début du morceau), et l'instant t inclus. Ainsi l'objet de niveau n d'instant t est affiché au moment de la segmentation de l'objet en voie de constitution de niveau inférieur. Le plan de l'organisation est hors temps. Pour retrouver le rapport entre les matériaux des différents niveaux, il

est agrémenté de liens qui partent des matériaux de niveau n et pointent vers les matériaux de niveau $n - 1$ qui les constituent. Ces liens pourraient être ajoutés sur le plan de la structure, qui connectent les objets de niveau supérieur et les objets qui les constituent au niveau inférieur mais ils n'ajoutent pas d'information à la lecture du diagramme.

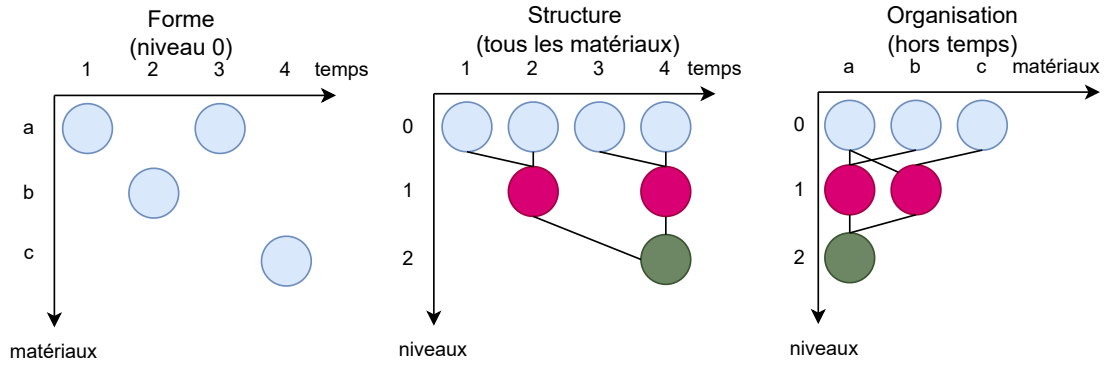


FIGURE 2.19 – Plans de la forme (de niveau 0), de la structure (tous matériaux considérés) et de l'organisation (à la fin de l'analyse).

2.3.2 Formalisation

Le plan de la forme est formalisé sur un niveau n par $r^{(n)} = x_1^{(n)} \dots x_l^{(n)}$ telle que présentée dans les sections précédentes.

Le plan de la structure et le plan de l'organisation représentent la relation entre un objet de niveau supérieur et les objets qui le constituent au niveau inférieur, indépendamment du matériau (pour le plan de la structure), ou indépendamment du temps (pour le plan de l'organisation). Ainsi, on pose pour un niveau $n > 1$ l'ensemble constitué des couples $(x_i^{(n)}, w_i^{(n-1)})$, selon les notations présentées dans la section précédente. $w_i^{(n-1)}$ est donc le facteur d'indice i et de niveau $n - 1$ correspondant aux objets de niveaux $n - 1$ qui constituent l'objet $x_i^{(n)}$ de niveau n . Pour le niveau $n = 0$, l'hyperplan défini par le plan de la structure et le plan de l'organisation est uniquement décrit par l'ensemble des objets qui constituent le niveau n soit l'ensemble des singletons $(x_i^{(n)})$.

Le plan de la structure est ainsi décrit par l'ensemble $\{(x_i^{(n)}, w_i^{(n-1)})\} \forall i \in [1, l]$ projeté indépendamment du matériau, c'est-à-dire que ce plan décrit uniquement

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

l'information de l'indice i et des indices correspondants au niveau inférieur, mais pas du label associé aux objets.

Le plan de l'organisation est décrit par l'ensemble $\{(x_i^{(n)}, w_i^{(n-1)})\} \forall i \in [1, l]$ projeté indépendamment du temps, c'est-à-dire que ce plan décrit uniquement l'information du label associé à $x_i^{(n)}$ et des labels correspondants au niveau inférieur $w_i^{(n-1)}$, mais pas de l'indice temporel associé aux objets. Plus précisément, la projection du côté de l'origine des axes correspond aux objets associés aux premières occurrences de chaque matériau.

Dans notre implémentation, les plans de la forme, de l'organisation et de la structure sont modélisés par les structures de données du diagramme formel, de la table des historiques et de la table des liens. Celles-ci sont décrites dans le chapitre suivant.

2.3.3 Applications

L'étude de ces différents plans peut aider à comparer les visualisations entre différentes œuvres, différents styles ou différents compositeurs. On peut par exemple se demander si la forme peut être récursive aux différentes échelles temporelles d'une même œuvre musicale ou si la structure est caractéristique d'un genre ou d'une époque. Pour aller plus loin, un analyste pourrait observer comment les compositeurs jouent avec les codes de forme et de structure pour rendre l'œuvre musicale plus raffinée. En ce qui concerne le plan d'organisation, on peut se demander par exemple si ce plan est similaire selon les compositeurs ou selon les personnages d'un opéra dans différentes *Aria*.

Différentes opérations peuvent alors être définies sur chacun des plans. Par exemple, on peut imaginer garder les trois plans intacts mais modifier le contenu des matériaux associés. On note que si deux plans sont identiques entre deux analyses, le troisième est nécessairement identique également.

Dans la figure 2.20 deux exemples musicaux simples présentent la même forme et la même organisation mais pas les mêmes matériaux : L'exemple musical 1 présente les matériaux *do*, *ré* et *mi* tandis que l'exemple musical 2 présente les matériaux *rebond*, *cloche* et *bruit*. Il s'agit ainsi de deux œuvres dont l'organisation, la structure et la forme sont identiques et dont l'expérience d'écoute peut être tout à fait différente de par le choix des matériaux qui les constituent.

Une autre possibilité est, à partir d'une œuvre donnée, de garder le même plan de l'organisation, c'est-à-dire la même relation hors-temps entre les matériaux,

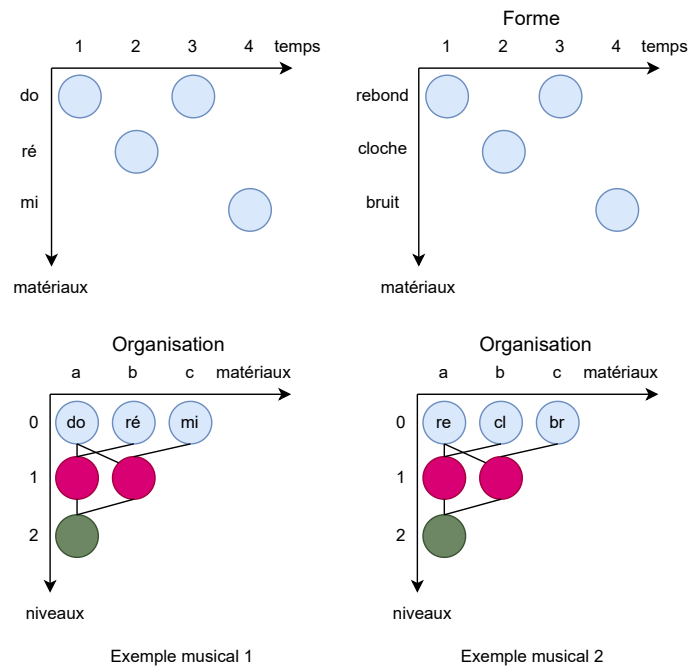


FIGURE 2.20 – Plan de la forme, de l’organisation (et de la structure) fixes et changement de matériau.

mais modifier sa forme et sa structure. La figure 2.21 représente deux exemples avec les mêmes matériaux, et la même organisation mais deux structures différentes. Dans les exemples musicaux 1 et 2, le matériau musical A de niveau 1 est constitué des matériaux a et b du niveau 0 et le matériau musical B est constitué des matériaux a et c du niveau 0. Le matériau musical A de niveau 2 est quant à lui constitué des matériaux A et B de niveau 1. Mais la répétition du matériau B de niveau 1 dans le temps modifie la forme ainsi que la structure de l’œuvre.

Une autre transformation est par exemple de garder la même structure, mais de présenter une organisation et une forme différente. La figure 2.22 présente deux analyses qui ont la même structure, seulement, le matériau B de niveau 1 n’est plus constitué des matériaux a et c de niveau 0, mais des matériaux b et c . L’objet $x_3^{(0)}$ ne correspond alors plus au label *do* dans le diagramme formel, mais au label *ré*.

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

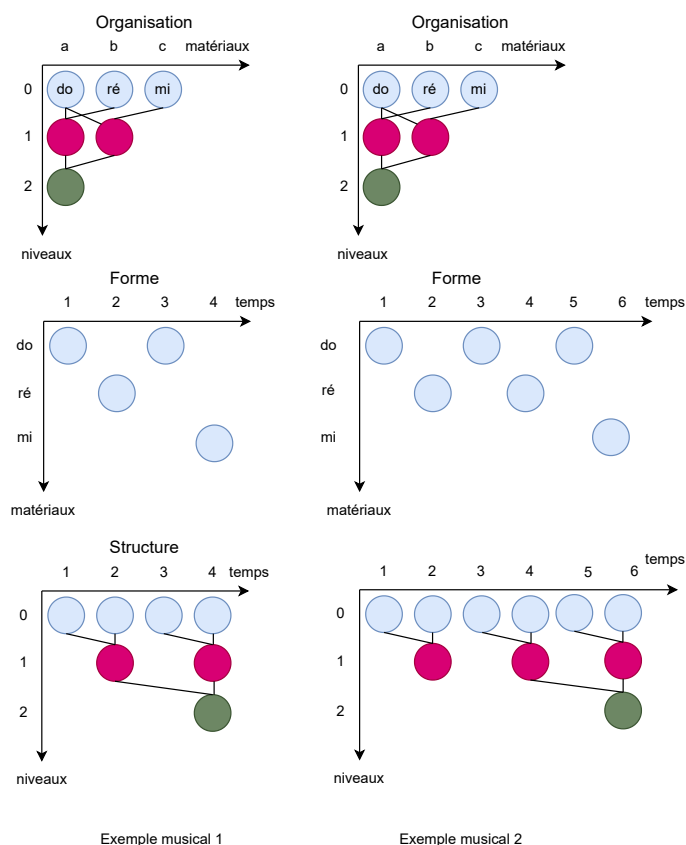


FIGURE 2.21 – Plan de l'organisation fixe et modification du plan de la forme et de la structure.

2.4 Agenda

Il est possible d'obtenir différents diagrammes formels en fonction de l'écoute, des critères de similarité, de segmentation. Ces diagrammes peuvent être complémentaires.

2.4.1 Écoutes

L'algorithme cognitif et MORFOS permettent de révéler les agencements temporels du flux sonore auquel il est soumis et de visualiser le comportement d'une écoute musicale. Cependant, si un professeur demande à une classe de 10 étudiants de musicologie, sans contraintes particulières, de proposer une analyse d'une même

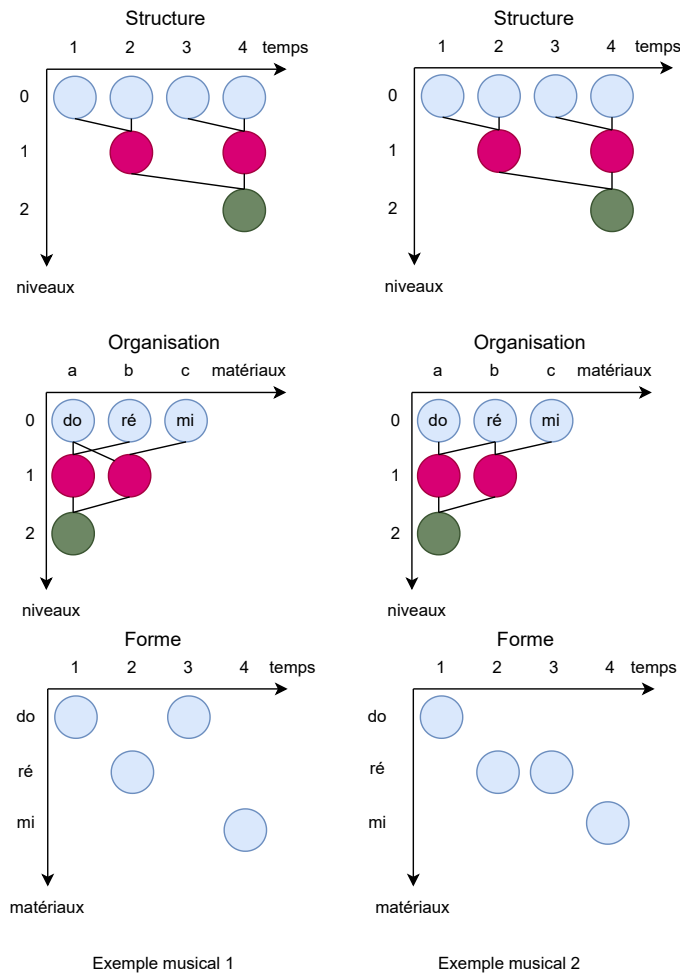


FIGURE 2.22 – Plan de la structure fixe et modification du plan de l’organisation et de la forme.

œuvre musicale, il y a de fortes chances que ceux-ci fournissent des analyses différentes, sur des dimensions différentes. De manière plus globale, si il est demandé à une population générale leur compréhension d’une œuvre, certains diront que tout leur semble similaire, tandis que d’autres seront sensibles aux plus légères variations. Dans *La musique au-delà des notes* de F. Delalande [42]², celui-ci cite : "L’écoute "actuelle" de tel auditeur dans telle circonstance est éminemment individuelle et changeante. Il semble donc exclu de rapporter l’analyse à l’écoute si l’on ne trouve pas un moyen de réduire cette diversité. Une solution, souvent implicitement adoptée, est de supposer un auditeur idéal ou un auditeur moyen ("on entend"),

2. p.67

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

ce qui est incompatible avec la diversité observée des écoutes singulières.". Selon lui, on ne peut donc pas présenter une représentation de l'écoute "idéale" face à la variété des écoutes propres aux individus selon leurs sensibilités et leur contexte. Il présente ainsi des *conduites types* chez le musicien qui peuvent varier au cours d'une unique écoute d'une même œuvre pour un individu. Il présente ainsi l'*écoute taxinomique* qui consiste à extraire une structure générale de l'œuvre musicale entière, la *figurativisation*, qui réside dans la création de différents niveaux de métaphores, et l'*écoute empathique du matériau* qui consiste à écouter et ressentir dans l'instant les tensions de l'œuvre.

Lorsque ce n'est pas fonction de l'individu qui écoute, les critères d'écoute pertinents peuvent être fournis par le compositeur en début de morceau. En effet, selon Adorno [2], il faut "*Une rééducation de l'oreille pour écouter correctement la musique de Debussy*". L'écoute est ainsi dépendante du style musical, bien que les compositeurs cherchent justement à surprendre l'auditeur en jouant avec les codes et les règles de composition. Par ailleurs, l'écoute sera induite par le nombre de matériaux fournis par le compositeur et la récurrence de ceux-ci. On peut ainsi émettre une hypothèse concernant un lien entre le mode d'écoute et la quantité d'information, illustrée par le front de découverte dans le diagramme formel.

Cela pose la question de l'évaluation des résultats obtenus par le logiciel : en effet, comment peut-on juger qu'un diagramme obtenu est pertinent s'il existe autant d'écoutes que d'individus ?

Dans notre logiciel, l'écoute est définie comme un ensemble de critères donnés au préalable de l'analyse en tant que paramètres. Ainsi, on définit des seuils de similarité et de segmentation, des critères de similarité et de segmentation, ainsi que des opérations de transformation sur les fichiers audio pour en extraire des caractéristiques spécifiques, telles que la hauteur de note ou le timbre. Cet ensemble de critères est appelé l'*agenda*. La notion d'Agenda est inspirée de l'Agenda présenté par O. Laske dans [86], qui correspond à un ensemble de contraintes auditives permettant de passer de la représentation de l'objet tel qu'il est exprimé à la représentation de l'objet tel qu'il est entendu. Une fois l'Agenda fixé dans notre logiciel, il n'existe alors plus qu'un diagramme formel multi-échelle pertinent.

À partir d'un Agenda fixé, étudions les critères présentés par C.E. Cella [30] pour définir une bonne représentation de l'écoute. Selon lui, une bonne représentation recoupe les trois critères suivants : l'*invariance*, la *stabilité* et l'*unicité*.

L'invariance à un ensemble de transformations consiste à dire que si une transformation appartenant à cet ensemble est appliqué au signal, alors la représentation obtenue est la même. L'exemple que prend C.E Cella par rapport à la musique

est que la représentation doit être invariante selon une translation dans le temps. On pourrait également souhaiter souvent que celle-ci soit invariante par rapport à la transposition. La stabilité signifie que s'il y a un petit changement au niveau du signal, cela doit se retranscrire par au maximum une petite variation au niveau de la représentation. L'unicité enfin signifie que si l'on a deux représentations différentes, alors celles-ci proviennent nécessairement de deux signaux distincts.

Ainsi, si on pose $\phi(x)$ une représentation de x , il définit mathématiquement les propriétés de la manière suivante :

- unicité : $\phi(x) \neq \phi(y) \iff x \neq y$;
 - stabilité : $\|\phi(x) - \phi(y)\|_2 \leq C\|x - y\|_2$;
 - invariance (à un groupe de transformations G) : $\forall g \in G, \phi(g, x) = \phi(x)$.
- Avec C la distance 2, soit la distance euclidienne.

Discutons du critère d'unicité pour la représentation obtenue avec MORFOS. Nous avons déjà souligné précédemment la multitude d'écoutes possibles et donc de représentations possibles d'obtenir à partir d'un signal donné, c'est pourquoi nous considérons la discussion suivante avec un Agenda fixé et donc avec des critères de pré-traitement, classification et segmentation fixés.

Montrons par la contraposée $\phi(x) \neq \phi(y) \implies x \neq y$. On cherche donc à montrer que si l'on a deux signaux x et y tel que $x = y$, alors $\phi(x) = \phi(y)$. La représentation de niveau 0 est obtenue à partir d'un pré-traitement calculé par une fonction surjective $\phi^{(0)}$. Cette fonction dépend du signal fourni en entrée ainsi que des critères de pré-traitement fixés et définis dans l'agenda. Si $x = y$, on a donc $\phi^{(0)}(x) = \phi^{(0)}(y) = r^{(0)}$.

À partir d'une représentation $r^{(n)}$ de niveau n et donc une suite d'objet donnée et pour un ensemble de critères de classification cc et de segmentation cs fixés, la représentation de niveau $n + 1$ est obtenue par applications successives d'une fonction de segmentation $\phi_s^{(n)}$ qui dépend de la représentation de niveau n et des critères de segmentation, ainsi que de la fonction de classification $\phi_c^{(n)}$ qui dépend de la représentation segmentée de niveau n obtenue et des critères de classification. Cela donne la représentation de niveau $n + 1$ à partir de la transformation de la représentation de niveau n avec la fonction $\phi^{(n)} = \phi_c^{(n)} \circ \phi_s^{(n)}$. Les fonctions étant surjectives, on a unicité de la représentation obtenue pour $r^{(n)}(x) = r^{(n)}(y)$. Les niveaux supérieurs étant induits des niveaux précédents, on a par récursivité, unicité de l'ensemble des représentations calculées pour tous les niveaux, et donc unicité par ailleurs de la représentation multi-échelle.

Il y a par ailleurs une relation d'équivalence entre le diagramme formel $Df^{(n)}(x)$ et la représentation $r^{(n)}$ de niveau n et donc unicité du diagramme formel multi-

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

échelle équivalent à la représentation multi-échelle.

Montrons maintenant que $x \neq y \implies \phi(x) \neq \phi(y)$. prenons l'exemple où x est la séquence de notes (en notation anglaise) *CECGCECG* et y est la séquence de notes *CDCFCECF*. Le critère de similarité est l'égalité stricte entre les motifs musicaux et le critère de segmentation est le retour à une note connue. Alors, les deux signaux produisent les représentations :

$$r^{(0)} = abacabac$$

$$r^{(1)} = abab$$

$$r^{(2)} = aa$$

La réciproque n'est donc pas vraie dans le cadre de MORFOS si la donnée du signal correspondant aux matériaux inscrits n'est pas considérée dans la représentation. Cela permet cependant de chercher à comprendre les points communs entre les œuvres qui produisent la même représentation, et d'utiliser celle-ci pour de nouvelles compositions. Néanmoins, si la mémoire matériau est considérée incluse dans la représentation, c'est à dire chaque segment de signal qui est équivalent à chaque matériau inscrit, les deux représentations sont alors différentes par définition du signal.

Concernant la stabilité, à partir d'un fichier audio : une légère variation par exemple dans la hauteur des notes peut avoir un impact considérable dans la représentation de Fourier obtenue et donc dans sa comparaison avec les autres fragments entendus. A partir d'une représentation symbolique, l'unicité dépend de la robustesse des règles de similarité et segmentation aux variations de la syntaxe. Par ailleurs, une variation à un niveau n impliquera une variation de la représentation au niveau $n + 1$ qui se répercutera sur tous les niveaux supérieurs. Prenons ainsi l'exemple *CECGCECG* présenté au dessus. Un deuxième do altéré, noté \tilde{C} , peut induire du signal *CE \tilde{C} GCECG* les représentations suivantes :

$$r^{(0)} = abcdabad$$

$$r^{(1)} = abc$$

Ainsi, sans présenter ici de mesure de distance précise entre les deux représentations obtenues (celle-ci mériterait d'être développée), on constate que celles-ci sont très distinctes malgré une légère variation. Ainsi, le principal défaut de la représentation des diagrammes formels à partir d'un fichier audio est que celle-ci est dépendante des critères de traitement ou de classification ou segmentation choisis et peut être très peu robuste.

Enfin, nous nous intéressons au cas de l'invariance. Si une transformation est appliquée sur l'intégralité de l'œuvre analysée de telle sorte que les relations temporelles entre les objets ne soient pas modifiées (par exemple une transposition

ou un étirement temporel) alors la représentation formelle ne sera pas impactée. Prenons l'exemple du *Geisslerlied*, transposé à la tierce en figure 2.23. Bien que les contenus même des matériaux ne soient pas les mêmes, l'ordre d'apparition et des similarités de ceux-ci restent les mêmes, permettant d'obtenir ainsi le même diagramme formel : c'est uniquement le label des matériaux qui change. De même, si la durée de tous les matériaux est doublée (figure 2.24), le diagramme formel sera le même à un facteur deux sur l'axe du temps. C'est donc la même transformation qui est appliquée sur la visualisation.

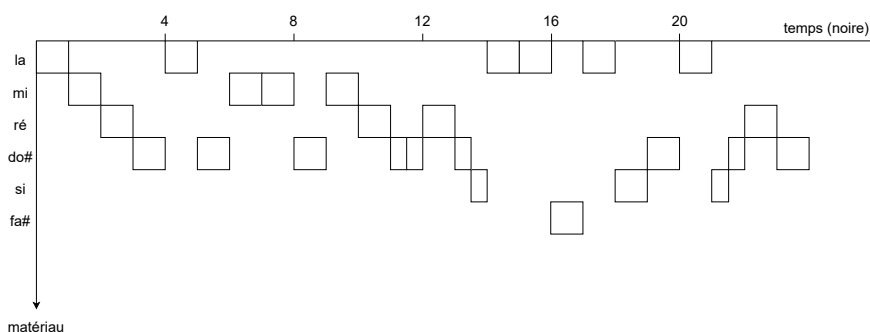


FIGURE 2.23 – Diagramme formel du *Geisslerlied* transposé à la tierce.

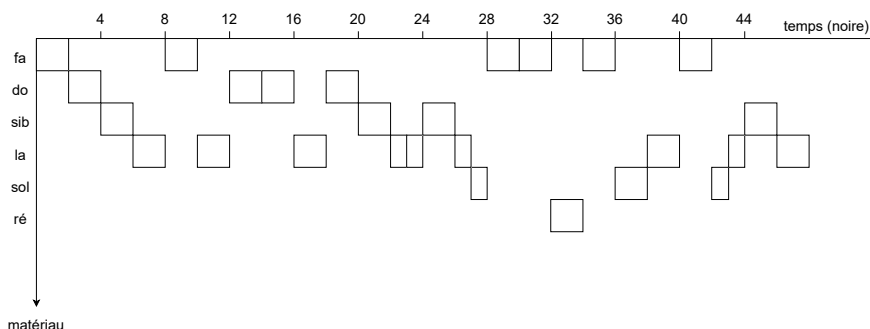


FIGURE 2.24 – Diagramme formel du *Geisslerlied* avec les durées multipliées par 2.

En revanche, à partir d'un fichier audio, l'invariance peut dépendre du traitement qui est fait de ce fichier. La Transformée de Fourier par exemple ne présente pas la même sensibilité dans les fréquences graves et aiguës, ce qui peut avoir un impact sur les critères de segmentation et de similarité nécessaires pour garder le même diagramme formel. La *Transformée à Q Constant (TQC)*, présentée par Schörkhuber et Klapuri dans [144] permet notamment de pallier ce problème en effectuant une analyse fréquentielle correspondante à l'échelle des notes de la gamme tempérée.

Ainsi ces trois critères sont fondés sur l'obtention d'une unique représentation en fonction de l'application de différentes opérations sur une représentation initiale donnée. Mais comment peut-on définir seulement quelque chose d'unique ? À partir de quand avons-nous variation, et à partir de quand avons nous distinction ?

Mais la question de la similarité se pose également au sein de l'œuvre musicale, à l'échelle de l'objet.

2.4.2 Similarité

La question de la similarité existe entre les morceaux musicaux ou au sein d'un même morceau. Nous sommes capables à l'oreille de classer des œuvres par style musical ou encore de reconnaître le morceau d'un groupe de musique, parfois sans jamais l'avoir entendu, dès les premières secondes d'écoute, grâce à la reconnaissance par exemple des timbres spécifiques à ce groupe [136].

En *Music Information Retrieval*, la question de la classification a été largement abordée pour différents critères. La reconnaissance du genre musical et des artistes est par exemple étudiée par G. Gabbolini et D. Bridge [56]. Ils introduisent une mesure de similarité calculée à partir d'une mesure « d'intérêt » des chemins entre les représentations des deux entités contenues dans un graphe. L'avantage d'une telle mesure est notamment qu'elle permet l'interprétation des résultats obtenus. La classification est également étudiée pour la reconnaissance des compositeurs : D. Herremans et al. [67] proposent cinq modèles de classifications, composés d'arbres de décisions ou de modèles statistiques qui, à partir de l'analyse de larges bases de données symboliques, distinguent les trois compositeurs Bach, Haydn et Beethoven. Pour le calcul de similarité, il existe également la détection de reprises : M. Sarfati, A. Hu et J. Donier [138] proposent un modèle de classification *many-to-many* : les auteurs ne calculent pas la similarité des œuvres deux à deux mais la similarité entre des groupes de potentielles versions d'une composition donnée par méthode de clusterisation, afin de créer un graphe des relations entre les fichiers audio correspondants. B. Martin et al. [96] quant à eux utilisent l'algorithme BLAST (*Basic Local Alignment Search Tool*) qui est un algorithme d'alignement local basé sur la détermination des régions d'alignement exact, appliqué dans ce cas précis sur les chromas des fichiers audios analysés. La reconnaissance d'accords est également étudiée par exemple par W.B. de Haas et al. [39] par l'application d'un algorithme d'alignement local (*Chord Sequence Alignment System*) appliquée à partir d'une représentation symbolique de la musique.

On note que, bien qu'il s'agisse d'un domaine de recherche actif, l'utilisation

de réseaux de neurones pose pour le moment le problème de l'explicabilité. Par ailleurs, certaines méthodes de clusterisation par exemple nécessitent d'avoir une connaissance préalable de l'intégralité de l'information à classer. C'est pourquoi nous avons décidé de mettre de côté ces techniques pour la classification (et la segmentation) avec MORFOS.

Qu'il s'agisse d'une représentation symbolique ou d'une base de données de référence, celles-ci nécessitent une labélisation préalable permettant d'induire ce qui est considéré par l'algorithme comme similaire ou distinct par défaut. Pourtant, selon la précision à laquelle on se situe, deux mêmes notes jouées à deux instants différents peuvent être considérées comme différentes, et la même œuvre jouée par deux artistes différents peut être considérée comme deux versions très similaires.

Ainsi, il s'agit de déterminer quelle est l'importance de la variation : souhaite-t-on privilégier le paradigme ou la subtilité du paradigme ? Pour le moment, MORFOS impose une considération binaire de la similarité, mais celle-ci peut être remise en question en affichant une valeur de similarité de l'objet actuel aux autres objets comme le propose la Matrice de Similarité [53].

D'un point de vue cognitif, J.P. Thibaut [156] présente deux paramètres permettant de déterminer la similarité. Le premier est la création d'un *espace de traits ou dimensions*, et le deuxième est la quantification de la similarité. L'espace de traits distingue quatre unités de descriptions :

- les *dimensions* sont un ensemble de valeurs discrètes ou continues telles que la longueur ou la largeur ;
- les *traits*, ou *attributs* sont des critères réels qui peuvent prendre une parmi deux ou quelques valeurs mutuellement exclusives, telles que le genre ou l'orientation d'une image ;
- les *parties* sont les entités qui composent celle que l'on décrit
- les *touts* correspondent à l'entité dans sa globalité, sans décomposition des entités qui le composent.

Le calcul de similarité est alors basé soit sur le partage de nombreux attributs communs entre les entités (modèles de traits), soit sur la proximité des dimensions qui les décrivent (modèles de dimensions). Par ailleurs, J.P. Thibaut souligne l'importance du contexte ainsi que les théories et connaissances développées par les sujets qui sont amenés à comparer des entités. Ainsi, deux notes de musique distinctes pourront être classées dans la même catégorie « son harmonique » si le reste de la pièce est constitué de bruits divers par exemple. Par ailleurs, si les groupements peuvent être distincts selon un contexte donné, ils peuvent également être distincts selon l'individu et par exemple, son âge. J.P. Thibaut explique que un enfant aura tendance à décrire une similarité selon des critères de *surface* là où

un adulte utilisera plutôt des critères *fonctionnels*. Plus généralement, un novice utilisera plutôt des caractéristiques superficielles, et un expert des caractéristiques profondes, c'est à dire intrinsèques à l'organisation de la classe. Différents critères de pondérations entrent ainsi en jeu, tels que l'*intensité*, la *fréquence*, la *familiarité*, le *principe de bonne forme* et le *contenu informationnel* du trait.

Par ailleurs, il distingue la similarité et catégorisation. En effet, il souligne que selon Medin et al. [100] la classification dépend non seulement de la similarité de l'item à classer avec les autres éléments de la catégorie donnée mais également de la relation entre les éléments qui constituent cette catégorie.

2.4.3 Segmentation

N. Meeùs dans [106] présente la segmentation comme une « activité primordiale de l'analyse musicale », intrinsèque à l'existence de la forme musicale. Il présente ainsi entre autres facteurs la segmentation d'une œuvre musicale comme « sa résolution en unités distribuées selon l'axe du temps et dont la durée est dès lors une caractéristique essentielle. »

Ainsi, il compare la segmentation musicale à la segmentation linguistique, similaires selon lui de par la nature essentielle semblable entre la musique et le langage. Néanmoins, si la segmentation linguistique est possible de par l'aspect linéaire du langage, ce qui est également le cas de la musique, qui est par essence temporelle, le discours, et la musique encore plus, peuvent en réalité être multi-dimensionnels et contrapuntiques. Nous laissons de côté dans le cadre de cette thèse le problème du contrepoint et de la séparation des sources car nous estimons qu'il s'agit d'un problème complexe nécessitant une étude approfondie. N. Meeùs dit par ailleurs que la segmentation permet de fractionner la musique de manière organisée en segments. Ainsi la segmentation est obtenue à partir de la *structure phraséologique*, de la *structure thématique* ou bien de *modèles formels*. La phraséologie a été théorisée par H. Riemann [129], qui présente une structure construite sur 4 niveaux de hiérarchies fondamentalement ancrés sur la métrique musicale et allant d'une durée de la mesure à la période. Il propose ensuite différentes opérations permettant des variations de cette structure stricte. Cooper et Meyer [37] quant à eux proposent un modèle métrique plus général, étendu à la musique atonale, tandis que Schoenberg propose une phraséologie basée sur les phénomènes de mémorisation des unités paradigmatiques. La structure thématique consiste en la détermination et l'exploitation d'un thème, défini par Koch comme une phrase contenant une idée sémantique et qui est l'idée initiale au développement compositionnel d'une œuvre musicale. Cette phrase a par ailleurs la caractéristique d'être manipulée au

long de l'œuvre par un travail de répétition et de transformation. Ainsi, le modèle formel découle du traitement qui est fait du thème.

Dans le modèle d'écoute présenté par S. McAdams et E. Bigand [13], la segmentation lors de l'écoute musicale est effectuée après le traitement du signal en fonction des principes de groupements extraits de la *Gestalttheorie*. Celle-ci présente des lois concernant les processus de perception de la forme. Parmi elles, la *loi de continuité* induit que les points rapprochés sont d'abord perçus comme une continuité, la *loi de proximité* induit que les éléments les plus proches sont regroupés, et la *loi de similitude* induit que les éléments les plus similaires sont regroupés.

Ainsi, l'informatique musicale s'est penchée sur la question de la segmentation d'une part pour l'estimation de la structure générale d'une œuvre musicale mais également pour la détermination du tempo et du rythme, qui peuvent être utiles pour la détermination de la métrique et de la carrure musicale.

Des études sur la détection et l'estimation du tempo utilisent des réseaux de neurones tels que le réseau convolutif CNN MGANet (*Multi-scale Grouped Attention Network*) dans [153] ou encore le *beat-tracking* avec l'utilisation de réseau de neurones temporels convolutifs [18]. H. Schreiber, F. Zalkow et M. Müller [143] utilisent également des réseaux convolutionnels pour la détection locale du tempo dans les Mazurkas de F. Chopin. M. Rohrmeier [132] propose une formalisation du rythme sous forme de grammaire générative et symbolique de la structure rythmique et des structures sous-jacentes de manière récursive, permettant d'obtenir une représentation arborescente du rythme.

F. Bimbot, V. Gillot et C. Louboutin [16], [58], [91] utilisent pour la segmentation le modèle Système & Contraste [15] qui est la transformation successive d'une *graine* a par deux fonctions distinctes f et g , puis par une troisième fonction qui est la composition de la fonction γ , induisant une surprise, et des deux fonctions précédentes f et g . Ainsi, on présente la morphologie définie par le modèle Système & Contraste par :

$$af(a)g(a)\gamma(g(f(a)))$$

La surprise du quatrième élément clos le segment musical.

B. McFee et K. M. Kinnaid [99] proposent d'améliorer les études de segmentation en étendant automatiquement les représentations à un seul niveau en représentations hiérarchisées de trois niveaux : le niveau supérieur qui augmente le nombre de labels et un inférieur qui diminue le nombre de label, ceci afin d'augmenter l'information structurelle pour la segmentation. G. Shibata et al. [61] proposent également d'étudier la segmentation par méthode de clusterisation à partir d'une

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle : formalisation de MORFOS

représentation hiérarchisée d'un modèle caché semi-markovien basé sur les critères d'homogénéité, de répétition, de nouveauté et de régularité. G. Shibata et al. [62] propose également l'utilisation hybride du modèle précédent avec un réseau de neurones LSTM (*Long-Short-Term Memory Network*).

On note que dans le cadre d'un calcul d'une transformée de Fourier, on effectue une segmentation arbitraire correspondante à la taille de la fenêtre d'analyse. Alors, le fragment correspondant au spectre calculé ne correspond pas nécessairement à une entité dont le contenu sémantique fait sens. Par ailleurs, il pourrait être intéressant de définir une notion de contraste et de segmenter en fonction du contraste musical, à l'image de la détection de contour par convolution en traitement de l'image [27] [20].

2.4.4 Multi-dimentionnalité

La distinction des matériaux dépend du choix de la projection de l'analyse sur l'œuvre musicale, celle-ci pouvant être rythmique ou motivique par exemple. Ainsi, les matériaux sont issus de l'œuvre en fonction de l'ensemble des *critères de classification* et des *critères de segmentation*, mais également en fonction des critères de pré-traitement de l'information.

En reprenant l'exemple du *Geisslerlied* en fonction de critères de pré-traitement rythmiques, le diagramme formel obtenu est présenté en figure 2.25. L'information rythmique est très monotone et ne présente que deux matériaux rythmiques, le premier étant la noire et le second la croche.

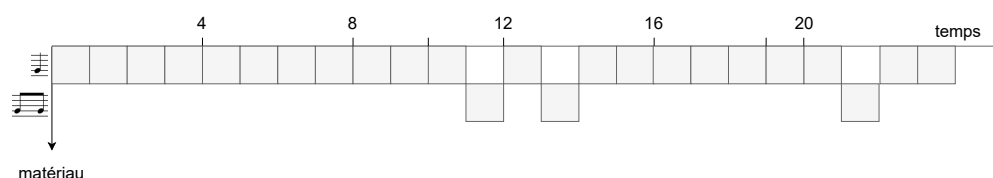


FIGURE 2.25 – Diagramme formel du *Geisslerlied* avec pour matériaux les motifs rythmiques.

Ainsi, un musicologue peut comparer les différents diagrammes formels obtenus pour une même œuvre musicale. Les informations de ceux-ci peuvent ainsi se renforcer, si les formes entre les deux diagrammes formels sont similaires (figure 2.26, en haut à gauche), se contredire, par exemple lorsque la découverte d'information sur un diagramme formel à lieu pendant une phase de stabilité d'un autre diagramme formel de cette même œuvre, ou bien lorsque les formes sont opposées

(figure 2.26, en haut à droite), ou encore relayer l'idée, par exemple en retrouvant la même forme se succéder d'un diagramme à un autre (figure 2.26, en bas).

Conclusion

Dans ce chapitre, nous présentons les notions introduites par Jean-Marc Chouvel du *Diagramme Formel*, une représentation des paradigmes musicaux au cours du temps, et de l'*Algorithmme Cognitif*, le processus qui permet d'obtenir dynamiquement des diagrammes formels.

Nous présentons ensuite notre propre formalisation de l'*Algorithmme Cognitif* avec le logiciel *Multiscale Oracle Representations For Organized Sounds* MORFOS. Ainsi, le *Diagramme Formel Multi-échelle* est défini par un mot $R = X_0.X_1...X_n$ avec n le nombre de niveaux qui constituent le diagramme formel multi-échelle. X_j est un triplet tel que $X_j = (r^{(j)}, s^{(j)}, k^{(j)})$ où $r^{(j)}$ est la représentation de niveau j telle que $r^{(j)} = x_0^{(j)}...x_n^{(j)}$ où $x_i^{(j)}$ est l'objet d'indice i dans la représentation de niveau j , et $x_i^{(j)} \in \mathcal{A}^{(j)}$ où $\mathcal{A}^{(j)}$ est l'alphabet qui contient les *matériaux* du niveau j . La représentation segmentée $s^{(j)} = w_0^{(j)}...w_k^{(j)}$ est une représentation qui regroupe les lettres définies sur $\mathcal{A}^{(j)}$ en facteurs w_i définis sur $\mathcal{A}^{*(j)}$ et $k^{(j)}$ est l'indice de la dernière segmentation dans $r^{(j)}$.

Les trois plans de la *forme*, de la *structure* et de l'*organisation*, qui correspondent à trois sous-représentations distinctes de l'information contenue dans le Diagramme Formel Multi-échelle sont introduites et feront l'objet d'exemples concrets dans la partie visualisation de cette thèse.

Enfin, l'Algorithmme MORFOS est un algorithme dynamique qui s'inspire de modèles cognitifs, c'est pourquoi nous proposons un système de paramétrisation du logiciel en fonction de critères souhaités par l'utilisateur, à travers ce que nous appelons un *Agenda*. Les notions d'écoute plurielle, de similarité, de segmentation et de mutli-dimension des représentations des œuvres analysées sont ainsi discutées.

2. L'Algorithme cognitif pour une analyse musicologique multi-échelle :
formalisation de MORFOS

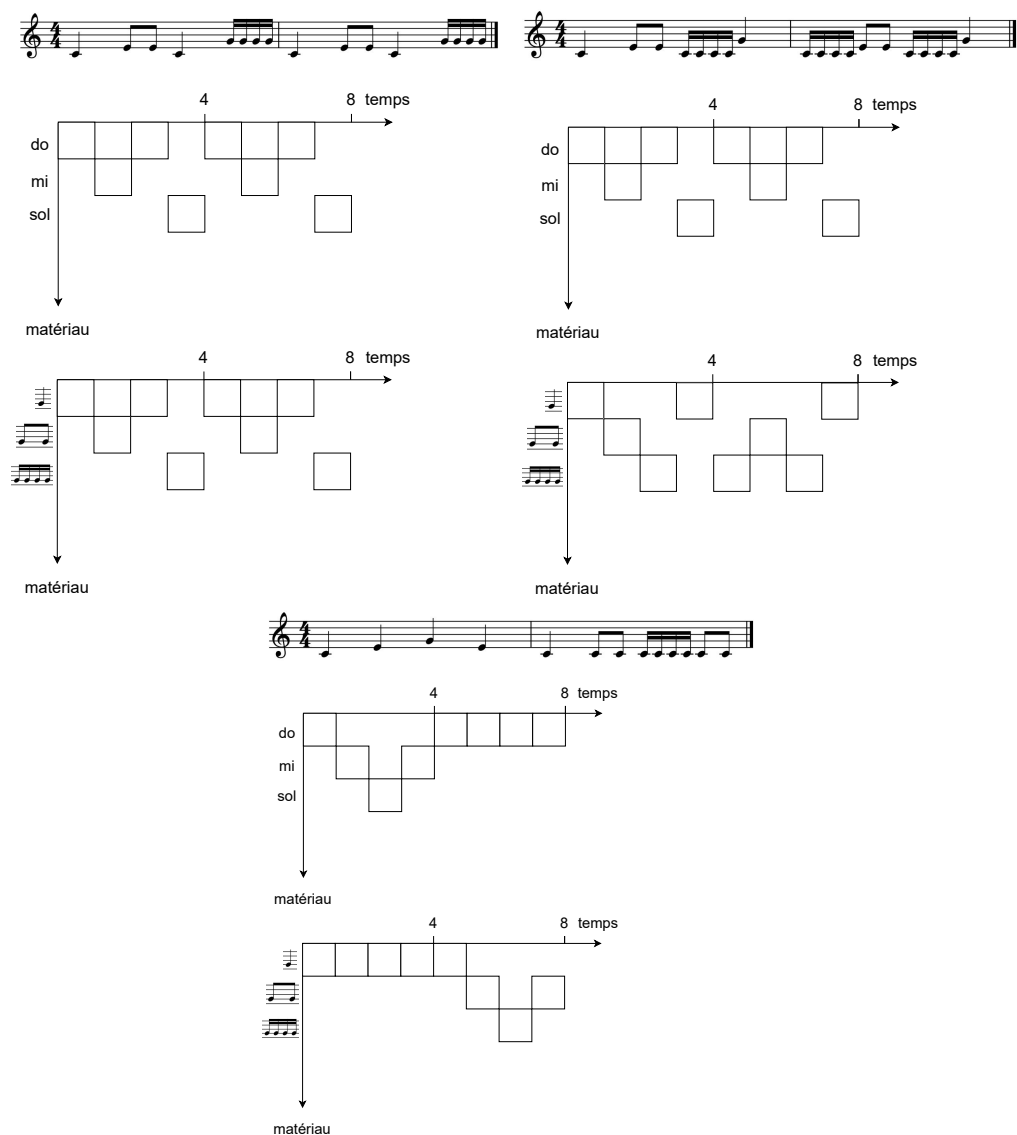


FIGURE 2.26 – Diagrammes formels ayant pour matériau la note (haut) et le rythme (bas) associés à la partition représentée au dessus de chaque diagramme.

Chapitre 3

Architecture et Implémentation de MORFOS

MORFOS est basé sur une architecture modulaire : le cœur du logiciel, constitué de l'algorithme MORFOS, de l'agenda, des structures de données, et de la visualisation découle de la modélisation présentée au chapitre 2. Des critères de classification, de segmentation et de traitement de données sont implémentés par défaut dans l'algorithme mais visent à pouvoir être agrémentés dans le futur par d'éventuels critères extérieurs au logiciel.

On présente dans ce chapitre l'architecture du logiciel ainsi que l'implémentation des différents modules. On introduit L'*oracle Multi-échelle*, une structure de données permettant de stocker les objets par niveaux et de modéliser la relation entre les objets des différents niveaux ainsi que les diagrammes formels. Cette structure de données est construite autour de l'*Oracle de Markov Variable* [160], un automate permettant à partir d'un état donné de l'automate, issu d'une série temporelle, de trouver en temps constant la plus longue chaîne de caractères similaire précédente et d'en déduire les probables états successifs. Cet automate nous permet ainsi de modéliser les phénomènes d'attention de l'Algorithme Cognitif.

La classification et la segmentation sont par ailleurs régies par un ensemble de critères, respectivement de classification et de segmentation. Ces critères sont ainsi décrits selon deux catégories *audio* et *symboliques* et correspondent à des opérations sélectionnées par l'utilisateur dans une interface de paramétrisation également décrite dans ce chapitre.

Notons que le logiciel a connu différentes versions. Si la boucle principale de l'algorithme est la même, celui-ci présente deux différences majeures.

Tout d'abord, l'architecture du code n'est pas la même : cela découle par ailleurs de l'approche objet qui a été mise en place dans la version 2.0. Ainsi, bien que la version 1 ait déjà une approche modulaire (séparation de l'agenda, des critères de segmentation, de classification et de l'algorithme), celle-ci n'est pas organisée de la même manière.

Là où la version 2.0 permet de prendre en compte à la fois l'information du signal et du label à tous les niveaux, la version 1 ne gère le signal qu'au premier niveau et n'utilise que l'information symbolique aux niveaux supérieurs, ce qui implique de nombreuses limites de segmentation. Ainsi, la boucle principale reste inchangée, cependant, une modification majeure est que dans la version 1, le code pour la gestion du niveau 0 est distinct de celui des niveaux supérieurs symboliques, là où le comportement a été factorisé dans la version 2.0. Ainsi, un algorithme de correction des objets de niveau 0 qui existait dans la version 1 (et qui reclassait un matériau de niveau 0 différent du matériau qui le précède et qui le suit en fonction de sa similarité à ces deux matériaux) n'a pas encore été intégré à la version 2.0. Nous l'aborderons à nouveau dans le chapitre 6.

Les deux dernières mises à jours des version 1 et 2 sont disponibles. Nous présentons dans ce chapitre 3 l'architecture de la version 2.0, qui se repose sur un modèle adapté dans le chapitre 2.

3.1 Architecture du logiciel

MORFOS est implémenté comme un ensemble de modules indépendants. Cela permet une meilleure adaptabilité du logiciel pour connecter des modules externes tels que les mesures de similarité ou de segmentation. Les différents modules sont les suivants :

- l'algorithme MORFOS ;
- l'implémentation des structures de données ;
- la représentation du diagramme formel ;
- l'interface des paramètres.

Ces quatre modules ne sont pas voués à être remplacés dans le logiciel à part pour de nouvelles versions améliorées de l'algorithme.

- le pré-traitement des données ;
- le calcul des règles de segmentation ;
- le calcul des règles de similarité.

Ces trois modules pourront être remplacés ou augmentés par d'éventuelles bibliothèques externes à MORFOS.

3.1.1 Pré-traitement des données

Le pré-traitement des données permet de passer d'une représentation audio ou symbolique à une représentation de niveau 0.

Dans un premier temps, le type du fichier fourni en entrée est déterminé :

- si la donnée est au format vectoriel, alors il n'y a pas de traitement à effectuer ;
- si la donnée est au format symbolique, alors un algorithme de re-labelisation est exécuté afin d'avoir une représentation symbolique ordonnée ;
- si la donnée est audio, alors une représentation au format vectoriel est extraite.

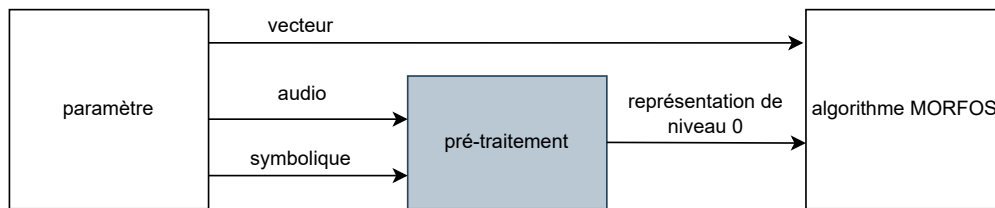


FIGURE 3.1 – Module de pré-traitement.

La donnée pré-traitée est ensuite fournie au module de l'algorithme MORFOS.

3.1.2 Module de segmentation

Ce module contient des fonctions permettant de déterminer s'il y a segmentation, en fonction de la structure de données et de l'objet acquis dans l'algorithme à l'instant t .

Il définit ainsi un ensemble de fonctions de la manière suivante :

$$segmentation_regle_n(mso, objet) : booleen$$

Ces fonctions sont appelées depuis le test de segmentation dans l'algorithme MORFOS.

3.1.3 Module de classification

Ce module contient des fonctions permettant de comparer deux vecteurs ou bien deux mots et de renvoyer une valeur de similarité entre les deux objets comparés.

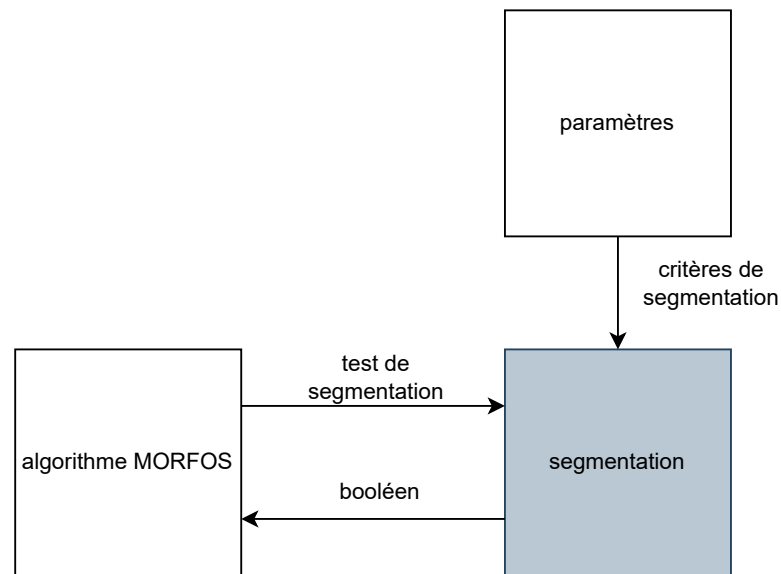


FIGURE 3.2 – Module de segmentation.

Un ensemble de fonctions de classification sont définies de la manière suivante :

$$\text{classification_regle_n}(\text{objet_a}, \text{objet_b}) : \text{flottant}$$

Les fonctions adéquates sont appelées lors de l'ajout de l'objet dans la structure de données et donc depuis le module structure de données, lui-même appelé depuis le module de l'algorithme MORFOS.

3.1.4 Implémentation des structures de données

Ce module contient la définition et l'implémentation des fonctions de mise à jour des différentes structures de données permettant de stocker les objets analysés.

- initialisation de l'Oracle multi-échelle ;
- initialisation d'un niveau de l'oracle multi-échelle ;
- fonctions de gestion de l'oracle des facteurs (développées par C.Weng [161] et adaptées) ;
- fonction d'ajout et de mise à jour dans les différentes structures de données présentées dans la section 3.1.

La suppression d'un objet n'est pas considérée car on ne peut pas "retourner en arrière".

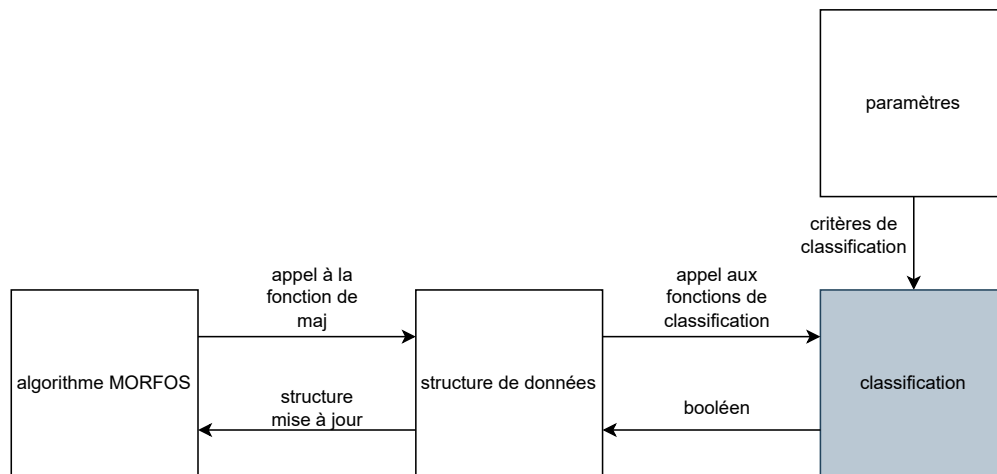


FIGURE 3.3 – Module de classification.

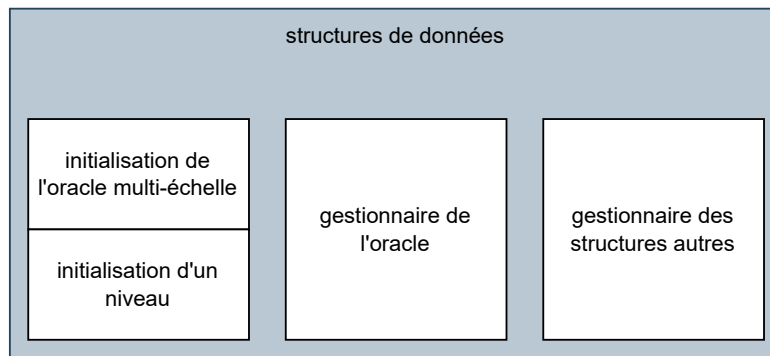


FIGURE 3.4 – Module de structure des données.

3.1.5 Visualisation du diagramme formel

Il s'agit du module qui, à partir d'une représentation du diagramme formel contenue dans la structure du MSO, propose une visualisation de celui-ci. La visualisation proposée dans ce module est produite à partir d'une représentation matricielle en 2 dimensions. Il s'agit d'une part des fonctions de transformation de la structure de données, et d'autre part des fonctions d'affichage.

Une autre visualisation est proposée et fait appel à un module externe qui est en réalité un logiciel implémenté en Electron et qui propose une visualisation 3D des diagrammes formels multi-échelles obtenus.

Ces visualisation sont détaillées dans le chapitre 5.

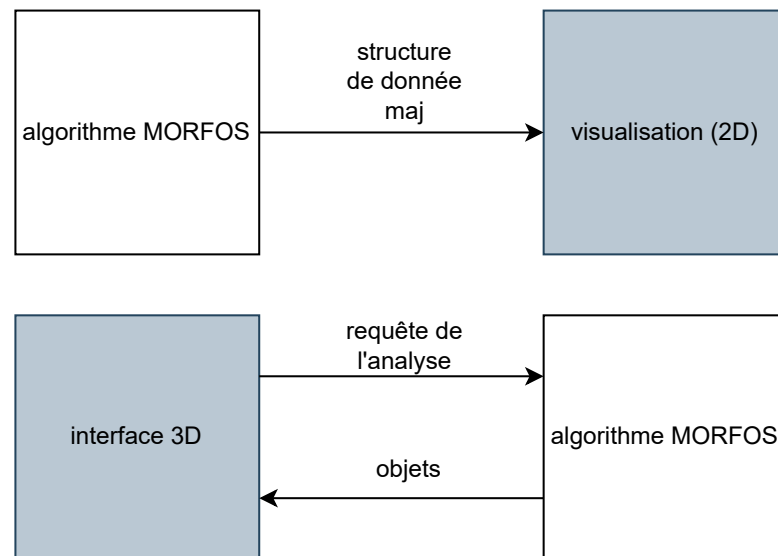


FIGURE 3.5 – Module de visualisation.

3.1.6 Interface des paramètres

les paramètres sont contenus dans un fichier JSON qui est appelé dans chacun des modules décrits ci-dessous. Ils contiennent d'une part l'agenda, à savoir :

- les données à analyser ;
- les critères de pré-traitement ;
- la liste des critères de segmentation ;
- la liste des critères de classification ;
- les éventuels hyper-paramètres de segmentation et de classification.

Les paramètres contiennent par ailleurs des paramètres statistiques, ainsi que des paramètres d'affichage.

Les données à analyser ainsi que les critères de pré-traitement sont soumis au module de pré-traitement, tandis que les critères de segmentation et de classification ainsi que les hyper-paramètres sont envoyés au module de l'algorithme MORFOS. Les paramètres statistiques sont envoyés au module de l'algorithme MORFOS et au module de structures de donnée, et les paramètres d'affichage sont envoyés au module de visualisation.

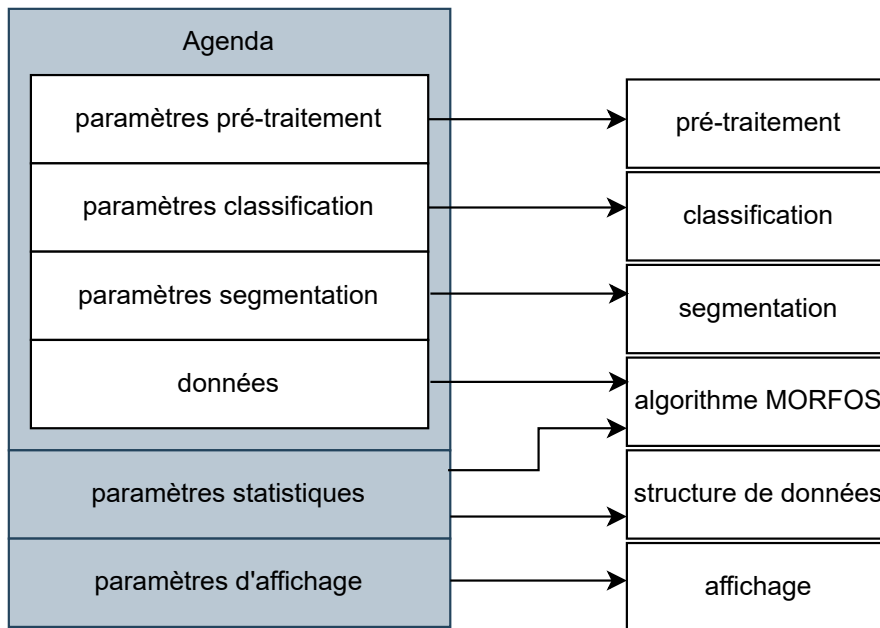


FIGURE 3.6 – Module de paramètres.

3.1.7 Algorithme MORFOS

Le module de l'algorithme MORFOS contient l'algorithme central au modèle d'analyse. Il fait appel à la représentation retournée par le module de pré-traitement, ainsi qu'aux structures de données mises à jour au fur et à mesure du calcul de l'algorithme. Ce module décrit les étapes suivantes :

1) S'il s'agit du premier appel à l'algorithme dans le niveau donné : il y a initialisation du niveau dans la structure de données : on procède alors à un appel au module de structure de données.

2) Dans un second temps, il y a acquisition de l'objet actuel. La mise à jour du diagramme formel est effectuée en fonction de l'objet acquis. Le calcul de classification est effectué à ce moment : l'opération de comparaison permettant de classer est définie dans le module de classification. Il y a ensuite à nouveau à un appel au module de structure de données pour le mettre à jour, puis au module d'affichage pour afficher le diagramme formel mis à jour.

3) Ensuite a lieu le calcul de segmentation : l'opération précise permettant de déterminer la segmentation est définie dans le module de segmentation.

4) La mise à jour des différentes structures est ensuite effectuée par appel des

fonctions définies dans le module de structure de données.

5) Enfin, un test permet de vérifier la terminaison ou non de l'algorithme.

Ainsi, l'architecture du logiciel est présentée en figure 3.7.

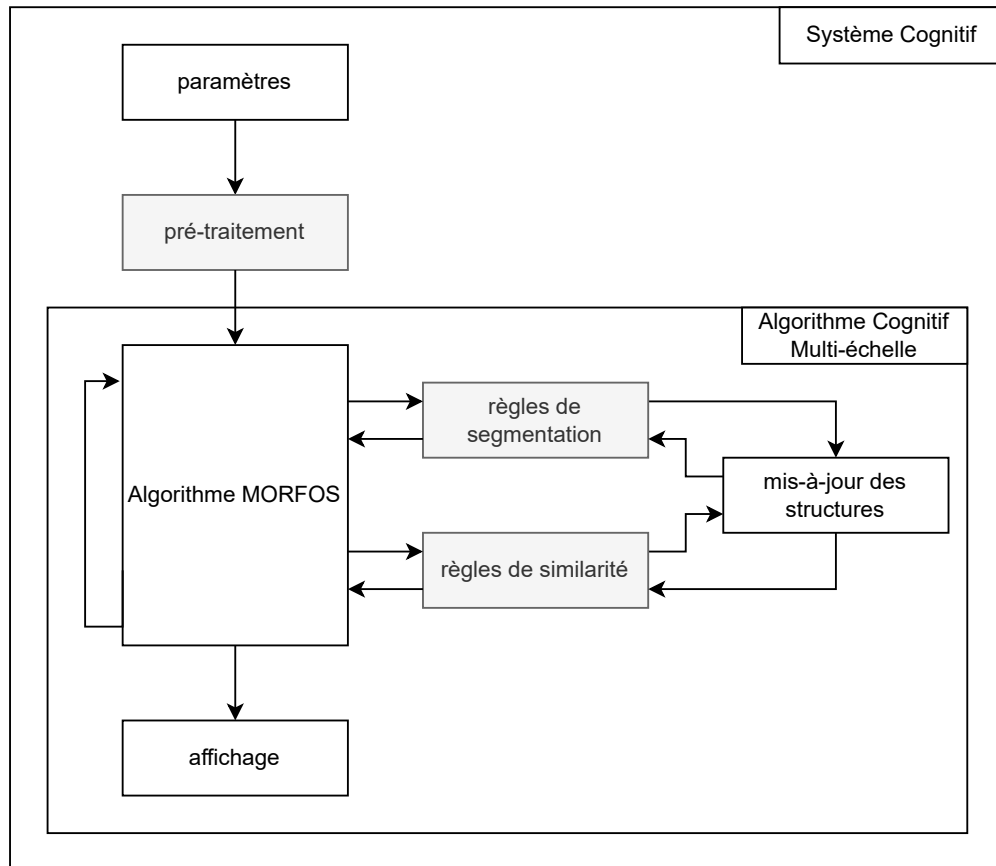


FIGURE 3.7 – Architecture de MORFOS.

3.2 Implémentation

Afin de faciliter l'implémentation algorithmique du modèle d'analyse et ne pas se concentrer sur les aspects génie logiciel, nous avons décidé d'implémenter le logiciel en Python3. Par ailleurs, de nombreuses bibliothèques pour le traitement du signal, les calculs de similarité et l'affichage ainsi que le calcul de l'oracle à une dimension existent déjà en Python.

3.2.1 Pré-traitement

Nous avons vu précédemment que l'algorithme MORFOS prend en entrée des représentations sous forme de mots dont les lettres appartiennent à un alphabet ordonné. Cependant, MORFOS peut prendre en entrée des fichiers au format audio, ou bien dans des formats symboliques divers. On a donc éventuellement une phase de pré-traitement automatisée en fonction du format du fichier fourni en entrée. Le format vectoriel, pris en charge également, ne nécessite pas de pré-traitement.

Pré-traitement d'un signal audio

Dans le cadre d'un fichier audio, les objets représentés par des lettres dans les mots correspondent soit à des spectres, soit à des descripteurs. Deux étapes sont exécutées :

- numérisation ;
- calcul du spectre ou de descripteurs.

Numérisation

Dans un premier temps, il y a numérisation du signal. Comme présenté en Chapitre 2, il est possible de rajouter après numérisation un silence en début de morceau. Cela est justifié par le fait que nous considérons le silence comme un critère fort de segmentation, qui termine une œuvre mais la précède également. C'est donc à cette étape du traitement du signal qu'un silence peut être au début du morceau. Pour cela, un nombre défini de 0 est ajouté au début de notre vecteur correspondant à notre fichier numérisé. Ce nombre est défini en paramètre par l'utilisateur comme un multiple de la taille de la fenêtre d'échantillonnage.

Calcul de spectre

Cependant, le fichier numérisé a une dimension encore très grande pour pouvoir appliquer des opérations de manière efficace dessus. Une fois le fichier audio numérisé, on souhaite obtenir une représentation significative et compacte de l'information audio.

Une de ces représentations est le spectre, qui présente l'amplitude pour chacune des fréquences qui constituent le son. Des exemples de spectres sont présentés dans le chapitre 1.

Cette décomposition en somme de sinusoïdes est possible grâce au calcul de la transformée de Fourier. La Transformée de Fourier est définie par la fonction $\mathcal{F}(f)$ avec f une fonction continue et périodique intégrable sur \mathbb{R} avec t le temps en secondes et ν la fréquence en hertz tel que :

$$\mathcal{F}(f) = \int_{-\infty}^{+\infty} f(t)e^{i2\pi\nu t} dt$$

En particulier, la Transformée de Fourier Discrète (TFD) est calculée pour un signal s des N termes $S(1), \dots, S(k), \dots, S(N)$ définis tel que :

$$S(k) = \sum_{n=0}^{N-1} s(n)e^{-2i\pi k \frac{n}{N}}$$

Étant donné la Transformée de Fourier Discrète, la Transformée de Fourier Rapide (FFT) est calculée et consiste à diminuer récursivement le nombre d'opérations de la transformée de Fourier en séparant les échantillons en deux sommes d'échantillons pairs et impairs [36], nécessitant ainsi que le nombre d'échantillons soit une puissance de deux.

Afin de limiter et d'optimiser encore le nombre d'opérations, nous proposons trois versions de la Transformée de Fourier :

- la Transformée de Fourier classique ;
- la Transformée de Fourier par bandes ;
- la Transformée de Fourier par fenêtres.

La *Transformée de Fourier* classique est calculée comme présentée précédemment et fournit une représentation de l'amplitude pour les N fréquences calculées, espacées par un intervalle fixé.

Celle-ci n'est pas très robuste aux légères variations de hauteur : en effet, deux signaux aux fréquences très similaires à l'oreille donnent une répartition différente des amplitudes des fréquences induisant deux représentations très distinctes lors du calcul de la transformée de Fourier. C'est pourquoi nous proposons une version permettant de moyenner sur un intervalle de fréquences : La *Transformée de Fourier par Bandes* (figure 3.9). L'amplitude est calculée pour chaque intervalle de fréquence. Pour cela, l'ensemble des amplitudes des fréquences de l'intervalle sont sommées, et pondérées par le nombre de fréquences incluses dans l'intervalle. Cela permet par ailleurs de réduire la taille du vecteur associé au spectre par un facteur correspondant au nombre de fréquences dans chaque intervalle. Néanmoins, l'échelle des notes n'est pas linéaire mais logarithmique. Une adaptation de la taille des bandes à la fréquence est proposée : plus les fréquences sont élevées, plus les intervalles seront petits.

Pour un intervalle $I = [K_1, K_2]$, on note $S(I)$ le spectre sur un intervalle I donné par la formule :

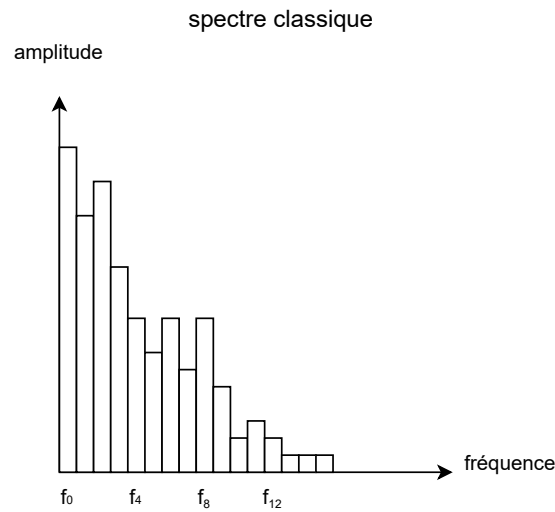


FIGURE 3.8 – FFT Classique.

$$S(I) = \frac{\sum_{k=K_1}^{K_2-1} S(k)}{K_2 - K_1}$$

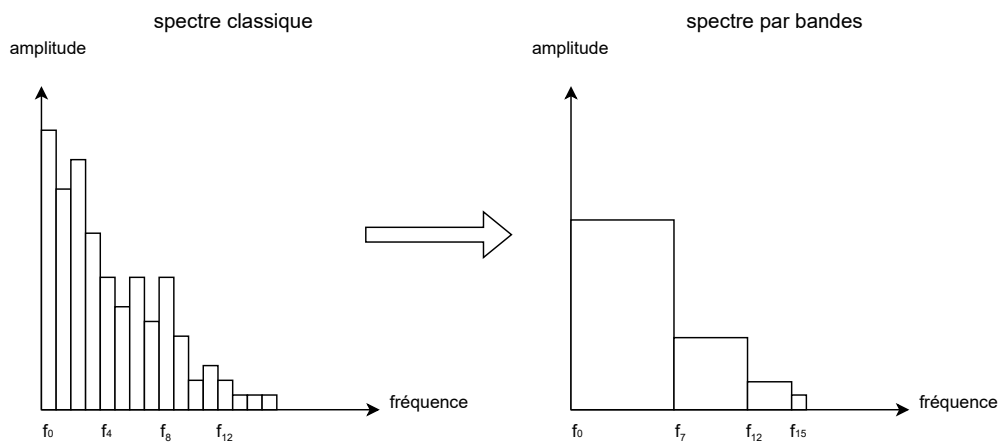


FIGURE 3.9 – FFT Bandes.

La *Transformée de Fourier par fenêtres* (figure 3.10) permet d'adapter directement l'écart entre les fréquences calculées du spectre en fonction de l'intervalle de fréquence analysé. Plus les fréquences analysées sont élevées, plus l'intervalle entre les fréquences calculées dans le spectre est petit.

Un autre calcul, particulièrement adapté au domaine de l'analyse musicale, est

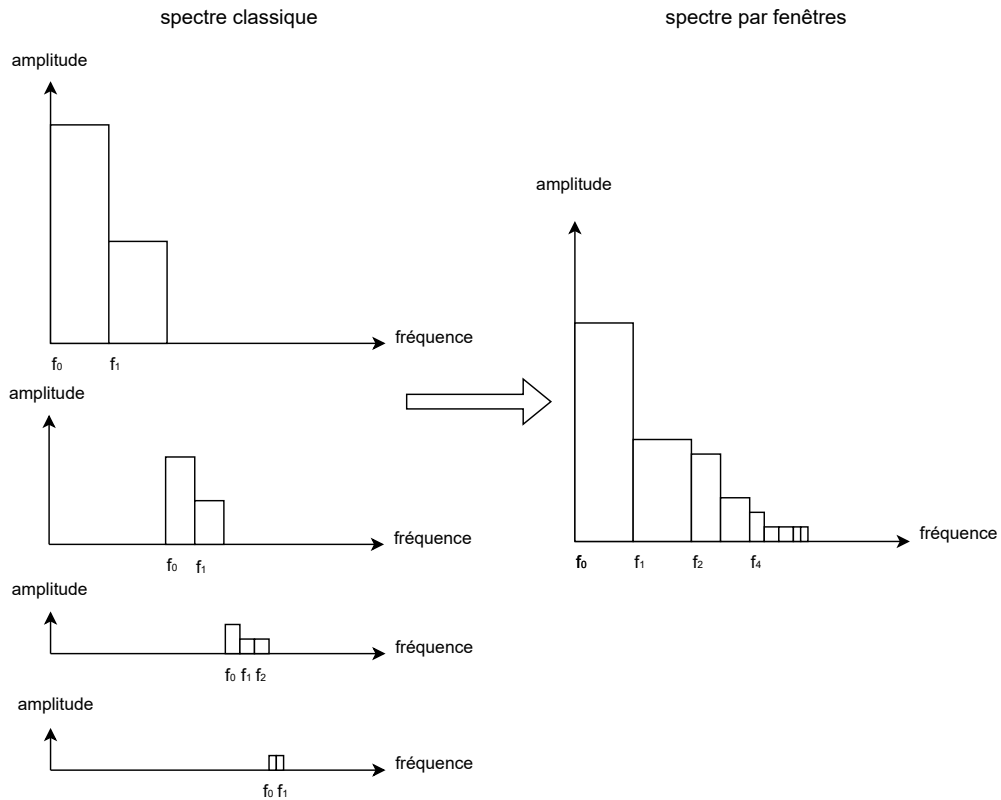


FIGURE 3.10 – FFT fenêtres.

la *Transformée à facteur de qualité Q Constant* [144]. Celle-ci à l'avantage, par rapport à la Transformée de Fourier, de garder une précision spectrale logarithmique, de la même manière que l'échelle des notes de la gamme chromatique. En revanche, cela se fait au dépens de la précision temporelle : plus les fréquences sont basses, plus la précision temporelle est dégradée.

La transformée à Q constant est définie selon son facteur de qualité Q comme le rapport entre une fréquence donnée f_k et l'intervalle de précision temporelle δf_k :

$$Q_k = \frac{f_k}{\delta f_k}$$

La Transformée à Q constant $CQT(k, n)$ d'un signal temporel discret $x(t)$ est ainsi définie par :

$$CQT(k, n) = \sum_{j=n-\lfloor \frac{N_k}{2} \rfloor}^{n+\lfloor \frac{N_k}{2} \rfloor} x(j)a_k^*(j - n + \frac{N_k}{2})$$

Avec $k = 1, \dots, K$ les k partiels calculées de la CQT, $\lfloor \cdot \rfloor$ la partie entière inférieure et $a_k^*(n)$ le conjugué complexe de $a_k(n)$ tel que $a_k(n)$ est défini par :

$$a_k(n) = \frac{1}{N_k} w\left(\frac{n}{N_k}\right) \exp\left[-i2\pi n \frac{f_k}{f_s}\right]$$

Avec f_s la fréquence d'échantillonnage et $w(t)$ une fonction de fenêtrage continue.

Le facteur Q intervient ainsi dans le calcul de la taille de la fenêtre d'analyse N_k tel que présenté dans l'article [144].

Calcul de descripteurs

Plutôt que des spectres, il est également possible de calculer divers descripteurs permettant de décrire des caractéristiques spécifiques du fichier audio.

Les premiers descripteurs que nous proposons sont les *Mel-Frequency Cepstral Coefficients* [92]. Ceux-ci décrivent l'information timbrale d'un fichier audio. Ils sont obtenus par le calcul du spectre d'amplitude du signal par une Transformée de Fourier Discrète (voir précédemment). On passe ensuite à l'échelle des Mel, qui consiste à calculer le logarithme du spectre de telle sorte à s'approcher des caractéristiques propres à l'oreille humaine :

$$mel(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right)$$

Une banque de filtres mel correspondants à un ensemble de bandes passantes triangulaires sont appliqués, avant une Transformée en Cosinus Discrète qui permet d'obtenir des coefficients aussi décorrélés que possible.

$$y(k) = \sum_{n=1}^N w(n)x(n)\cos(\cdot)$$

pour k allant de 1 à N et $w(n) = \sqrt{\frac{1}{N}}$ si $n = 1$ et $w(n) = \sqrt{\frac{2}{N}}$ sinon.

Les *chromas* [145] permettent de décrire le contenu harmonique du signal analysé et sont représentés par un vecteur de dimension 12 qui décrit l'énergie moyenne

de chaque demi-ton de la gamme chromatique. Ce vecteur est obtenu en calculant le spectre du signal en étiquetant chaque fréquence obtenue selon le demi-ton le plus proche, puis la moyenne arithmétique des énergies pour chacune des fréquences est effectuée sans distinguer les octaves. Le calcul du chroma de la classe de note \mathcal{N} à l'instant t est donc déterminé par :

$$C(\mathcal{N}, t) = \frac{\sum_{k=1}^N |S(2^k f_0(\mathcal{N}), t)|}{N}$$

avec N le nombre d'octaves considérées, f_0 la fréquence fondamentale pour la classe de note \mathcal{N} et $S(f, t)$ la TFD du signal à la fréquence f et à l'instant t .

De nombreux autres descripteurs existent et pourraient être utilisés dans le logiciel. Manuel Ghauliac, dans le cadre de sa thèse propose par exemple des descripteurs permettant de décrire l'harmonicité ou encore la rugosité d'un accord [93]. Nous aurons l'occasion d'en présenter quelques uns calculés dynamiquement à partir de deux spectres dans la suite de ce chapitre pour des mesures de similarité. Laurent Pottier quand à lui propose un programme MaxMSP permettant l'analyse d'une cinquantaine de descripteurs pour l'analyse de la musique actuelle, contemporaine ou électronique à partir du phénomène sonore [126]. Des logiciels d'extraction de descripteurs existent également tels que Marsyas¹, Yaafe², ou encore les logiciels MIR³ proposés par l'ISMIR (International Society for Music Information Retrieval). Ces descripteurs, calculés par un module externe pourraient être directement soumis à l'Algorithme MORFOS. Par ailleurs, nous pourrions également combiner les différents descripteurs, par exemple en concaténant les vecteurs, pour effectuer une analyse multi-strate de l'œuvre.

Pré-traitement d'une chaîne symbolique

Lorsque l'Algorithme MORFOS prend en entrée une représentation symbolique, celle-ci doit être ordonnée au sens défini dans le Chapitre 2. Si c'est le cas, alors elle ne nécessite aucun pré-traitement et peut être directement analysée par l'Algorithme. En revanche, il existe de nombreuses représentations où les symboles ne sont pas ordonnés mais associés à une sémantique spécifique, par exemple lorsque l'on décrit une série d'accords, ou bien encore le format MIDI.

Traitement d'une chaîne non ordonnée

1. <https://github.com/marsyas/marsyas>
2. <https://github.com/Yaafe/Yaafe>
3. <https://ismir.net/resources/software-tools/>

Soit un alphabet \mathcal{B} non nécessairement ordonné et une pré-représentation définie par le mot p sur cet alphabet tel que $p = y_1 \dots y_l$ avec y_i pour $i \in [1, l]$ des lettres définies sur \mathcal{B} . On souhaite obtenir une représentation initiale $r^{(0)} = x_1 \dots x_l$ tel que x_i pour $i \in [1, l]$ sont des lettres définies sur $\mathcal{A}^{(0)}$ un alphabet ordonné. On peut déduire le couple (x_i, y_i) qui associe à la lettre $y_i \in \mathcal{B}$ apparue en i ème position la lettre $x_i \in \mathcal{A}^{(0)}$ également en i ème position de $r^{(0)}$.

On présente l'algorithme 2 suivant, où m est le nombre de matériaux avant acquisition du fragment y_i par ailleurs égal au nombre de lettres définies dans $\mathcal{A}^{(0)}$.

Le principe de l'algorithme est pour tous les objets de la pré-représentation de comparer l'objet de la pré-représentation actuellement acquis y_i avec les objets précédemment acquis. S'il est déjà vu, le label x_k de la représentation initiale, associé à l'élément de la pré-représentation y_k similaire à y_i est assigné au label de la représentation initiale x_i . Sinon, un nouveau label de rang le plus élevé dans l'alphabet \mathcal{A} est créé. Dans tous les cas, on concatène à la représentation initiale le nouvel objet x_i .

Algorithm 2 Ordonnancement de la chaîne de caractère

```
pour chaque objet acquis  $y_i$  :  
  comparaison avec les fragments précédents ;  
  si  $\exists k \in [0, i[$  tel que  $\text{sim}(y_k, y_i) = 1$  :  
    alors  $x_i = x_k$  ;  
  sinon :  
    on définit  $x_i$  dans  $\mathcal{A}^{(0)}$  tel que  $\text{rg}(x_i) = m + 1$  ;  
     $r^{(0)} = r^{(0)}.x_i$  ;
```

fichier MIDI

Le traitement des fichiers MIDI est exactement un cas spécifique de l'algorithme précédent. La représentation p correspond à l'extraction des fréquences contenues dans le fichier MIDI à partir de la méthode `midiFile()` de la bibliothèque Python `music21`⁴.

On peut également récupérer l'onset du fichier MIDI. On définit alors $y_i = (f_i, o_i)$ avec f_i la fréquence du i ème élément et o_i la date de l'onset du i ème élément. On associe alors un objet x_i grâce à l'algorithme présenté dans la partie précédente et chaque objet x_i est lui-même défini par $x_i = (e_i, o_i)$ avec e_i le label de l'objet x_i . Cependant chaque objet est représenté par un unique indice dans le modèle du diagramme formel que nous présentons ci-dessous, l'onset est donc

4. <https://web.mit.edu/music21/>

utile principalement pour la question de la visualisation du diagramme, présentée au Chapitre 5.

3.2.2 L'oracle Multi-Echelle

Afin de stocker les objets et de modéliser les accès en mémoire, on implémente une structure de données que l'on appelle l'*Oracle Multi-échelle*. L'oracle multi-échelle est constitué de niveaux, chacun contenant les 5 structures de données suivantes :

- l'oracle ;
- l'objet en voie de constitution ;
- la table des liens ;
- le diagramme formel ;
- la mémoire matériau.

Ces structures sont représentées sur le schéma en figure 3.11

L'oracle

Les objets acquis dans le logiciel sont représentés en mémoire par un *Oracle de Markov Variable (Variable Markov Oracle)* tel que présenté par C. Allauzen, M. Crochemore, and M. Raffinot [3] et utilisé par G. Assayag, C. Wang et Shlomo Dubnov [5][160]. L'oracle est une structure de données initialement utilisée pour la compression puis pour l'improvisation de la forme $O = q_1q_2\dots q_t\dots q_l$. Il s'agit d'un automate qui trouve rapidement la plus longue chaîne de caractères similaire entendue précédemment dans le logiciel. Chaque état de l'automate correspond à l'observation à l'instant t d'une série temporelle discrète tandis que les liens représentent les successions possibles et comportent (pour les liens vers l'avant, définis ci-dessous) un label associé à l'état qu'il pointe.

Il existe deux types de liens qui connectent les états : les liens de suffixe (*suffix links*) et les liens vers l'avant (*forward links*). Les liens suffixes sont créés de l'état $t + k$ à l'état t avec $k > 0$ lorsque l'élément correspondant au lien q_{t+k} allant de l'état $t + k - 1$ à l'état $t + k$ et l'élément correspondant au lien q_t allant de l'état $t - 1$ à t sont similaires.

Les liens vers l'avant sont de deux types : les liens vers l'avant internes (*internal forward links*) correspondent aux liens allant de l'état précédent $t - 1$ à l'état t et correspondent au déroulement temporel de l'œuvre musicale. Les liens externes (*extern forward links*) sont créés de l'état t à l'état $t + k$ avec $k > 0$ lorsque le

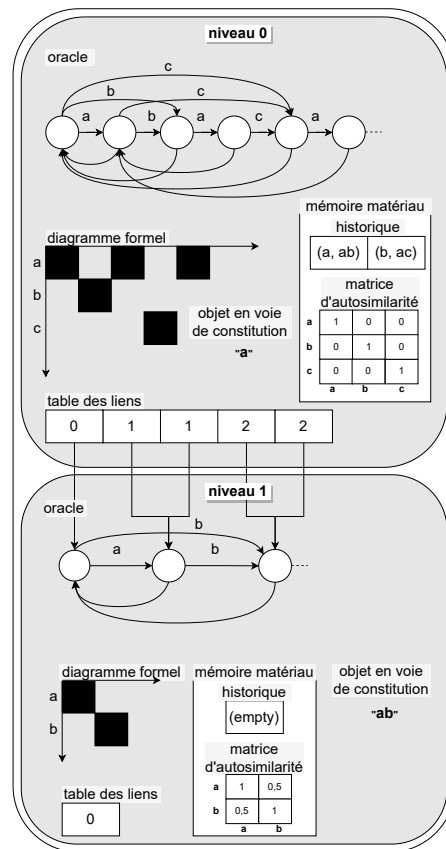


FIGURE 3.11 – Oracle Multi-échelle.

lien interne le plus récent q_{t+k} est précédé d'un lien tel que $q_{t+k-1} = q_t$ et que q_{t+k} n'a jamais suivi q_t auparavant. Chaque lien en avant est nommé d'après le label correspondant au matériau écouté à ce moment, les liens suffixes ne sont pas étiquetés et les états correspondent à chaque instant temporel t de la musique. Nous trouvons un exemple d'Oracle de Markov Variable créé à partir de la succession d'objets qui correspond à l'acquisition des 12 premiers objets de Mozart au niveau 0 et permettant de déduire la chaîne *abacabacdeab* dans la figure 3.12.

La création d'un nouvel état dans l'oracle se déroule de la manière suivante :

À partir de l'oracle $O = q_1q_2\dots q_{t-1}$ créé jusqu'à l'état $t - 1$, l'état t est ajouté, ainsi que le lien interne correspondant q_t . Ensuite, le matériau q_t est comparé à l'ensemble des liens internes q_i de l'état k , où $k < i < t$, de sorte que le suffixe de $t - 1$ pointe vers l'état k . Si un matériau similaire est trouvé, le suffixe de t pointe vers l'état i et l'ensemble des objets de classes q_i est mis à jour avec le nouvel élément q_t . Sinon, un lien externe est créé de l'état k à l'état t . L'opération est

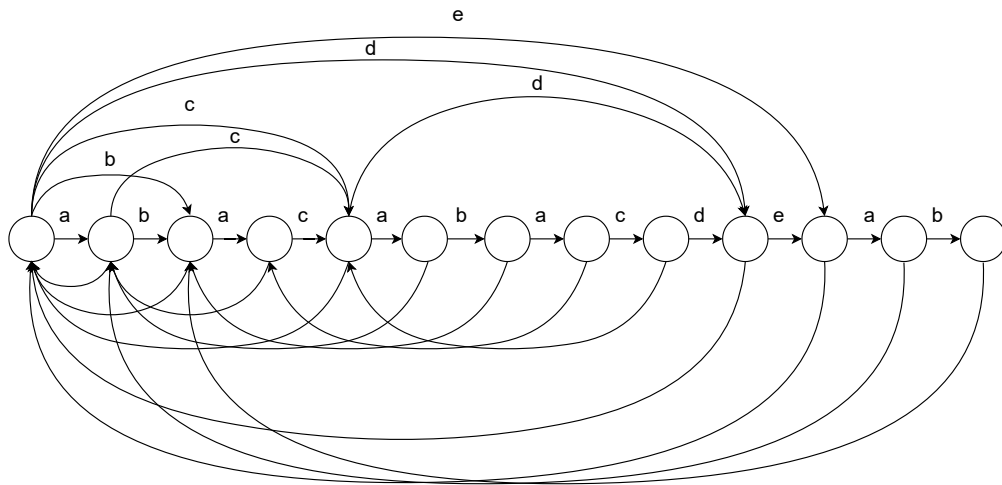


FIGURE 3.12 – Oracle de Markov Variable induisant la chaîne de caractère *abacbacdeab*.

répétée avec k_2 suffixe de k . Puis l'opération est répétée jusqu'à ce qu'un élément similaire soit trouvé dans les liens en avant de l'état k_n . Si aucun élément similaire n'est trouvé au cours de cette opération, le suffixe du nouvel état t est l'état initial 0 et une nouvelle classe est créée correspondant à q_t . Pour la mise en œuvre de l'oracle, nous utilisons le code développé par C. Wang et S. Dubnov⁵ [161].

L'avantage d'une telle organisation de la mémoire est qu'il n'est pas nécessaire de parcourir l'ensemble du mot analysé pour trouver un élément similaire déjà entendu. Chaque comparaison d'un élément à l'ensemble des objets déjà entendus est effectuée avec une complexité linéaire par rapport au nombre d'objets entendus. Avec l'Oracle de Markov Variable, l'accès au dernier élément similaire est constant une fois que le nouvel état est créé tandis que la construction d'un nouvel état dans l'oracle peut être effectuée en un temps de calcul linéaire par rapport au nombre d'états de l'oracle.

De plus, l'oracle modélise les phénomènes d'attention. En effet, grâce aux liens en avant du suffixe de l'objet actuellement entendu, nous pouvons faire une ou plusieurs hypothèses sur les objets futurs qui pourraient succéder à cet objet. Ces hypothèses seront validées ou non par le test de reconnaissance syntagmatique. La validation ou non de ces hypothèses renseigne sur le comportement cognitif lors de l'écoute musicale en consolidant la structure musicale ou, au contraire, en créant un effet de surprise.

5. <https://github.com/wangsix/vmo>

Néanmoins, l'oracle ne permet pas de segmenter, nous devons donc créer des règles pour segmenter ainsi que des structures de données pour stocker les objets et accéder aux différents niveaux. Pour cela, nous créons différents oracles $O^{(n)}$ à chaque niveau n existant, et nous les complétons avec d'autres structures de données qui stockent les informations reliant chaque niveau. C'est ce que nous appelons l'Oracle Multi-échelle (OME) ou encore *Multi-Scale Oracle (MSO)*.

L'objet en voie de constitution

L'*Objet en Voie de Constitution* est un mot qui correspond à la concaténation de tous les caractères lus depuis la dernière segmentation dans ce niveau. A chaque segmentation, cette chaîne de caractères est réinitialisée par le nouveau caractère correspondant au matériau dernièrement entendu, et les nouveaux objets sont concaténés jusqu'à une nouvelle segmentation.

Prenons l'exemple du Mozart à l'acquisition du 5eme objet au niveau 0

$$r^{(0)} = abaca$$

$$s^{(0)} = (ab)(ac)$$

$$r^{(1)} = ab$$

On a donc $R = (abaca, (ab)(ac), 5).(ab, \epsilon, 1)$.

Au niveau 0, la dernière segmentation a eu lieu juste avant l'indice 5, l'objet en voie de constitution correspond à tous les caractères lus depuis celle-ci dans la représentation de niveau 0 : on a donc $o_c^{(0)} = a$. Au niveau 1, la dernière segmentation est avant l'indice 1 donc $o_c^{(1)} = ab$.

Il s'agit ainsi du mot qui est comparé à chaque étape d'acquisition avec les matériaux déjà entendus.

La table des liens

Pour la construction d'un Oracle Multi-échelle, il faut créer un lien entre les oracles d'un niveau donné et les oracles de niveaux supérieurs : c'est pour cela que l'on a créé la *Table des Liens*.

La table des liens de niveau n est une liste qui associe à chaque état actuel l'indice de l'état correspondant dans l'oracle de niveau $n + 1$. Cette table est mise à jour à chaque segmentation. Lors de la segmentation, tous les caractères contenus dans l'objet concaténé, qui correspondent à plusieurs états de l'oracle de niveau n , correspondent à un seul caractère au niveau $n + 1$. Ainsi, on ajoute dans la table

des liens autant d'éléments qu'il y a de caractères contenus dans l'objet concaténé de niveau n . Ces indices ont la même valeur, qui est la valeur maximale de la table des liens incrémenté de 1. Cette valeur correspond à l'indice de l'état de niveau $n + 1$ associé au nouvel objet segmenté. En créant les liens temporels entre les objets des différents niveau, on note que celle-ci permet de modéliser le plan de la structure.

Avec l'exemple du Mozart à l'acquisition du 5eme objet de niveau 0 déroulé précédemment, on a :

$$Tl^{(0)} = (0, 1, 1, 2, 2)$$

Le 5eme objet, a , n'est pas encore remonté au niveau supérieur, on ne l'a donc pas encore associé à un état de niveau supérieur. De la même manière, le niveau 1 n'existant pas encore, donc $Tl^{(1)} = (0)$. La valeur du premier indice est 1 car l'indice 0 correspond à l'état initial de l'oracle et non pas une information du diagramme formel. Cet exemple est présenté en figure 3.13.

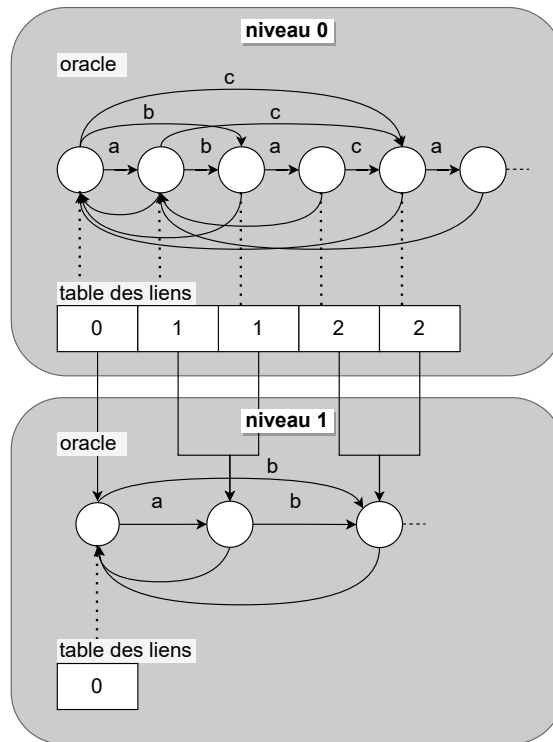


FIGURE 3.13 – Oracles et Tables des Liens sur l'exemple du *Rondo* K545 de Mozart à l'acquisition du 5eme objet de niveau 0.

Lors de l'acquisition du 7eme objet au niveau 0 , on a :

3. Architecture et Implémentation de MORFOS

$$r^{(0)} = abacaba$$

la segmentation donne

$$s^{(0)} = (ab)(ac)(ab)$$

alors, l'objet en voie de constitution est juste avant d'être recalculé $o_c^{(0)} = ab$: il est de longueur 2. Autant d'état sont ajoutés dans la table des liens que de nombre de caractères dans l'objet concaténé, et la valeur des états est la valeur maximum du tableau des liens incrémentée de 1. Cela donne :

$$Tl^{(0)} = (0, 1, 1, 2, 2, 3, 3)$$

Ensuite, R est mis à jour et l'objet en voie de constitution $R = (abaca, (ab)(ac), 7).(ab, \epsilon, 0)$, $o_c^{(0)} = a$

Cet exemple est présenté en figure 3.14.

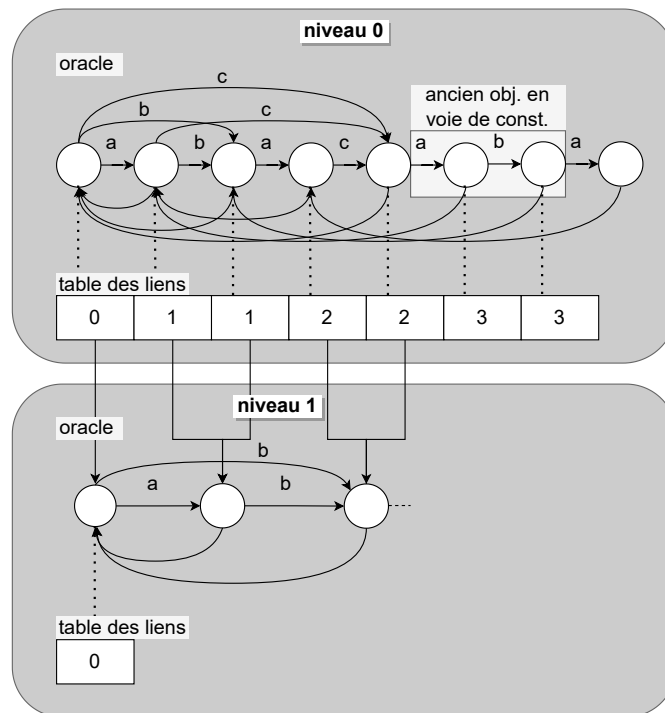


FIGURE 3.14 – Oracles et Tables des Liens sur l'exemple du *Rondo* K545 de Mozart à l'acquisition du 7eme objet de niveau 0.

Ainsi, ce tableau nous permet d'accéder à partir d'un niveau donné à tous les niveaux supérieurs et inférieurs.

Le diagramme formel

Le *Diagramme Formel* de niveau n correspond à la représentation formelle de l'état actuel de l'oracle au niveau hiérarchique n . Ce diagramme est représenté par une matrice $Df = (d_{i,j})$ de taille $m \times l$ où m est le nombre de matériaux entendus jusque là au niveau n et l le nombre d'objets entendus au niveau n depuis le début de l'analyse. Pour chaque indice, $d_{i,j} = 1$ si à l'instant j le matériau i est analysé et $d_{i,j} = 0$ sinon.

La mise à jour du diagramme formel se déroule alors en trois étapes distinctes :

- ajout d'une ligne matériau (si nécessaire) ;
- ajout d'une colonne objet ;
- représentation de l'objet.

L'ajout d'une ligne a lieu si le rang du matériau correspondant à l'objet acquis est supérieur à la taille de la matrice Df . Ici, rg est défini comme précédemment par l'opération permettant d'obtenir à partir d'une lettre définie sur un alphabet $\mathcal{A}^{(n)}$ le rang de cette lettre dans l'alphabet. L'opération de concaténation $concat(s)$ est également définie comme prenant en entrée un sens s parmi (col, lin) indiquant la concaténation d'une ligne contenant l zéros ou bien d'une colonne contenant m zéros. on a donc :

Si $rg(x_t) > m$:

$$Df_{m,l} \xrightarrow[\text{lin}]{\text{concat}} Df_{m+1,l}$$

Une colonne de la taille du nombre de matériaux actuels est ensuite concaténé :

$$Df_{m+1,l} \xrightarrow[\text{col}]{\text{concat}} Df_{m+1,l+1}$$

La représentation de l'objet dans le diagramme formel consiste ensuite à attribuer la valeur 1 à l'indice $d_{i,j}$ avec $i = rg(x_t)$ et $j = l + 1$.

Reprenons l'exemple du Mozart, cette fois-ci lors de l'acquisition du 2e objet de niveau 1, c'est-à-dire avant classification du deuxième facteur de la représentation segmentée de niveau 0. Avant classification au niveau 1, on a :

$$r^{(0)} = abaca$$

$$s^{(0)} = (ab)(ac)$$

$$r^{(1)} = a$$

$$Df^{(0)} = \begin{matrix} a \\ b \\ c \end{matrix} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

avec $a, b, c \in A^{(0)}$ et

$$Df^{(1)} = a \ (1)$$

avec $a \in A^{(1)}$

Les labels des matériaux ne sont pas stockés dans la matrice car directement corrélés à la ligne associée, et sont affichés ici sur ces premiers exemples à titre indicatif et didactique. On ne les affichera plus sur les matrices des exemples suivants.

Après classification de (ac) et donc acquisition du 2e objet au niveau 1, la représentation multi-échelle est :

$$r^{(0)} = abaca$$

$$s^{(0)} = (ab)(ac)$$

$$r^{(1)} = ab$$

1) Au niveau 1, $rg(b) = 2$ et $m = 1$ donc $rg(b) > m$: une nouvelle ligne correspondant au nouveau matériau b est donc concaténée dans le diagramme formel de niveau 1 :

$$Df^{(1)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

2) On concatène ensuite une colonne de longueur égale au nombre de matériaux, c'est-à-dire de longueur 2 :

$$Df^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

3) L'objet est ensuite représenté dans la matrice en appliquant $d_{i,j} = 1$ avec $i = 2$ la valeur du matériau et $j = 2$ car il s'agit de l'acquisition du deuxième objet au niveau 2, correspondant par ailleurs au nombre de colonnes dans la matrice Df .

$$Df^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Cela donne les diagrammes formels aux niveaux 1 et 2 présentés en figure 3.15.

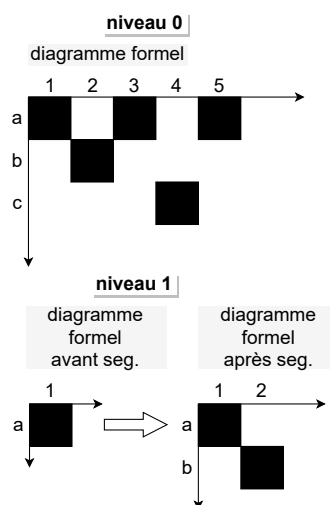


FIGURE 3.15 – Diagrammes formels sur l'exemple du *Rondo* K545 de Mozart à l'acquisition du 5eme objet de niveau 0 avant et après segmentation et classification au niveau 1.

La mémoire matériau

La *Mémoire Matériau* $Mm^{(n)}$ de niveau n correspond à deux structures de données : La *Table des Historiques*, ainsi que la *Matrice d'Autosimilarité*.

La table des historiques de niveau n fournit la correspondance entre tous les matériaux de niveau $n + 1$ obtenus jusqu'à l'instant l et les matériaux qui les constituent au niveau n . Il s'agit d'un tableau de paires (x_i, w_i) avec x_i une lettre définie sur $\mathcal{A}^{(n+1)}$ correspondante au matériau de niveau supérieur $n + 1$ et w_i facteur de $\mathcal{A}^{(n)*}$ correspondant au mot équivalent du niveau actuel n . Ainsi, la table des historiques permet de mémoriser hors temps la relation entre les matériaux des différents niveaux : il s'agit d'une modélisation du plan de l'organisation.

Sur l'exemple de Mozart après acquisition du 5eme objet et classification de l'objet segmenté (ac) tel que $R = (abaca, (ab)(ac), 5).(ab, \epsilon, 0)$, on a

$$Th^{(0)} = ((a, ab), (b, ac))$$

et

$$Th^{(1)} = ()$$

La matrice d'autosimilarité correspond à la matrice $S = (s_{i,j})_{1 \leq i,j \leq m}$ avec m le nombre de matériaux entendus, tel que $\forall i, \forall j \in [1, m], s_{i,j} = sim(w_i, w_j)$ où

$sim(w_i, w_j)$ est la fonction de qui renvoie la valeur de similarité entre les facteurs w_i et w_j de $\mathcal{A}^{(n)}$.

Sur l'exemple de Mozart à $t=5$:

Si la mesure de similarité considérée est le score de l'alignement⁶ pondéré par la longueur de l'alignement, de telle sorte que l'alignement est calculé à partir des similarité initiales au niveau 0 définies telles que $\forall x_i$ et x_j sur $\mathcal{A}^{(0)}$, $sim(x_i, x_j) = 1$ si $x_i = x_j$ et 0 sinon,

$$sim(ab,ab) = (sim(a,a)+sim(b,b))/2 = 1$$

$$sim(ab,ac) = sim(ac,ab) = (sim(a,a)+sim(b,c))/2 = 0,5$$

$$sim(ac,ac) = 1$$

Au niveau 0, la matrice d'autosimilarité est donc :

$$S^{(0)} = ((1, 0.5), (0.5, 1))$$

3.2.3 L'algorithme MORFOS

On présente dans cette partie le pseudo-algorithme de l'Algorithme MORFOS. Celui-ci est décomposé en trois parties : *L'Algorithme Principal* est la boucle principale de l'algorithme, permettant de mettre à jour l'ensemble des structures de données. On présente également *l'Algorithme de Segmentation* qui permet de déterminer si il y a segmentation ou non, ainsi que *l'Algorithme de Classification* qui permet de classer les objets de manière optimisée.

L'algorithme principal

L'algorithme principal consiste à appeler la fonction *algorithme_cognitif(niveau n , représentation r , marqueur m)*. On définit une représentation partielle rp comme un sous-mot de taille 1 à l d'une représentation $r = x_1...x_l$. La fonction prend en argument un niveau n généralement initialisé à 0, une représentation partielle rp initialisée à $r^{(0)}$, ainsi qu'un marqueur de fin d'algorithme m initialisé à 0. L'algorithme 3 présente la fonction principale.

Avec $Tl^{(n)}$, $Df^{(n)}$, $oc^{(n)}$ et $Mm^{(n)}$ les structures de données de la table des liens, du diagramme formel, de l'objet en voie de constitution et de la mémoire matériau de niveau n tels que définis ci-dessus. La fonction *init_oracle* initialise l'Oracle de Markov Variable $O^{(n)}$ en fonction d'un ensemble de paramètres par défaut *params* et *parse(rp)* la fonction qui prend en argument une représentation partielle rp et

6. Nous définissons plus précisément l'alignement dans la section 3.3.

Algorithm 3 `algorithme_cognitif(niveau $n = 0$, représentation partielle $rp = r^{(0)}$, marqueur $m = 0$)`

Si le niveau n n'existe pas et $m == 0$
 $init_mso_n()$
 Tant que l'on analyse rp :
 $c \leftarrow parse(rp)$
 $Df^{(n)}(c, t) \leftarrow 1$
 $afficher(Df^{(n)})$
 Si ($segmentation(n, c, cs)$ et $m == 0$) ou ($m == 1$ et $|o_c^{(n)}| > 0$)
 $niveau_sup(n + 1, m)$
 $o_c^{(n)} \leftarrow concatener(o_c^{(n)}, c)$
 Si $n == 0$ et EOS
 $m \leftarrow 1$
 Si $m == 1$
 $niveau_sup(n + 1, m)$
 $o_c^{(n)} \leftarrow concatener(o_c^{(n)}, c)$

Algorithm 4 `init_mso_n()`

$O^{(n)} \leftarrow init_oracle(params)$
 $Tl^{(n)} \leftarrow [0]$
 $Mm^{(n)} \leftarrow (tab(empty), matrix(empty))$
 $Df^{(n)} \leftarrow matrix(empty)$
 $o_c^{(n)} \leftarrow string(empty)$

renvoie le premier objet de cette représentation. On présente l'algorithme 5 qui correspond à la fonction de passage au niveau supérieur. Celle-ci prend les mêmes arguments que la fonction principale.

La fonction principale *algorithme_cognitif* est une fonction récursive divisée en trois parties.

La première partie correspond à l'initialisation des différentes structures de données au niveau actuel si elles n'existent pas. L'oracle $O^{(n)}$ de niveau n est initialisé avec la fonction d'initialisation fournie dans le *Module Variable Markov Oracle* développé par C. Wang et S. Dubnov [161]. La table des liens $Tl^{(n)}$ de niveau n est initialisée avec un nœud de valeur 0 qui correspond à l'index de l'état initial de l'oracle. La mémoire matériau $Mm^{(n)}$ de niveau n est initialisée avec une table vide et une matrice vide tandis que le diagramme formel de niveau n $df^{(n)}$ est initialisé par une matrice vide. L'objet en voie de constitution $o_c^{(n)}$ de

Algorithm 5 niveau_sup(niveau n, objet c, marqueur m)

```

s ← classification(n, cc)
tn+1 ← max(Tl(n)) + 1
Pour i in range(|oc(n)|)
    Tl(n) ← concatener(Tl(n), tn+1)
oc(n) ← string(empty)
algorithme_cognitif(n + 1, s, m)

```

niveau n est par ailleurs initialisé avec une chaîne de caractères vide. La fonction démarre ensuite une boucle qui analyse la représentation partielle rp fournie en argument de la fonction. Elle ajoute dans les structures de données le nouvel objet acquis à ce niveau. Tout d'abord, on récupère un objet nommé c correspondant au premier objet lu dans la représentation rp et obtenu grâce à la fonction *parse*. Ensuite, le diagramme formel $Df^{(n)}$ est mis à jour, ce qui signifie que le matériau correspondant à l'objet c est ajouté par une ligne supplémentaire au diagramme formel s'il n'existe pas encore, et la valeur de la matrice correspondante au matériau c et au temps t est fixée à 1. Le diagramme formel mis à jour est ensuite affiché.

Dans une deuxième partie, un test de segmentation est effectué en fonction d'un certain nombre de critères de segmentation cs , du niveau n (et plus précisément des structures de données qui le constituent) et le nouvel objet acquis c . Ce test est effectué par une fonction *segmentation*(n, c, cs) qui renvoie 1 si il y a segmentation à l'instant t et 0 sinon. Cette fonction est détaillée plus précisément dans la partie suivante.

Si le test de segmentation est validé à l'instant t , il y a alors opération de classification. Celle-ci prend également en arguments les structures de données du niveau n ainsi qu'un ensemble de critères de classification cc . Elle renvoie le matériau correspondant à l'objet en voie de constitution $o_c^{(n)}$. Cette fonction est également détaillée dans la partie suivante. Après classification, la table des liens de niveau n est mise à jour avec l'index correspondant du nœud de niveau supérieur. L'objet en voie de constitution est ensuite réinitialisé à la chaîne de caractères vide et la fonction *algorithme_cognitif*($n + 1, s, m$) est appelée au niveau supérieur avec comme représentation s au niveau supérieur, le nouvel objet s correspondant à un objet instance du matériau associé à l'objet en voie de constitution. Aux niveaux supérieurs, la chaîne de caractères correspond donc à un seul objet, et la fonction retourne au niveau inférieur lorsqu'il n'y a pas de segmentation. Ainsi, si la fonction est appelée au premier niveau, la représentation $r^{(0)}$ est lue tant qu'il n'y a pas de segmentation. Enfin, qu'il y ait segmentation ou non, $o_c^{(n)}$ est mis à jour en concaténant le caractère analysé c .

Une question est de savoir quand terminer l'algorithme aux niveaux supérieurs sans créer un nombre infini de niveaux. Pour cela, nous utilisons un *marqueur*. Cela nous amène à la troisième partie : lorsque le dernier caractère de la représentation initiale $r^{(0)}$ est lu, le marqueur passe de zéro à un et reste à un pour chaque appel à la fonction aux niveaux supérieurs. La segmentation à chaque niveau est alors systématiquement appliquée. La fonction *niveau_sup* est donc appelée à nouveau pour les derniers caractères de chaque niveau, sans créer de nouvel état pour chaque niveau. Par ailleurs, si le niveau supérieur n'existe pas, il n'est pas créé et l'algorithme retourne au niveau inférieur. Comme toute l'information concernant le signal d'entrée est fournie au logiciel à cet instant, la lecture et la segmentation des caractères restants dans les niveaux intermédiaires sont achevées quelques étapes plus tard.

L'algorithme de segmentation

La fonction de segmentation est définie par *segmentation(niveau n, objet o, critères cs) : boolean* qui renvoie 1 si il y a segmentation et 0 sinon.

Le niveau n permet d'accéder à l'ensemble des structures de données du niveau actuel n et en particulier à l'objet en voie de constitution, qui contient les objets de niveau n qui constituent l'objet de niveau supérieur $n + 1$, à l'oracle de niveau n afin de comparer avec les objets similaires précédemment entendu ainsi qu'à la table des liens pour avoir l'information de segmentation entre les objets de l'oracle. L'objet c quand a lui donne l'information du dernier objet acquis.

Penchons-nous sur les critères de segmentation : Les critères de segmentation correspondent à un ensemble de règles sélectionnées au préalable de l'analyse et permettant de déterminer s'il y a segmentation ou non. Un objectif du logiciel étant sa modularité, nous souhaitons proposer aux utilisateurs développeurs de pouvoir ajouter leurs propres règles de segmentation.

Implémentation des règles de segmentation

Les règles de segmentation sont ainsi constituées de deux paramètres : le *type*, et l'*implémentation*.

Le type de la règle de segmentation peut être de trois sortes : *obligatoire*, *interdite* ou encore *bivalente*.

Les règles de segmentation obligatoire sont des règles qui induisent nécessairement la segmentation à l'instant précédent l'acquisition de l'objet actuel. Elles renvoient donc la valeur 1 si il y a segmentation entre l'objet acquis x_{i+1} et le dernier objet x_i .

Les règles de segmentation interdite sont des règles qui interdisent la segmentation entre l'objet actuellement acquis et le précédent. Elles renvoient donc la valeur 1 si il n'y a pas de segmentation entre l'objet acquis x_{i+1} et le dernier objet x_i .

Les règles de segmentation bivalentes induisent des contraintes d'interdiction de segmentation et des contraintes d'obligation de segmentation. Celles-ci sont implémentées indépendamment en deux règles, l'une de segmentation obligatoire et l'autre de segmentation interdite.

L'implémentation est la fonction permettant de définir la contrainte de segmentation. L'implémentation de la fonction est laissée libre au développeur, mais la définition de la fonction admet deux contraintes : l'entête de la fonction doit prendre strictement en argument l'objet x_{i+1} ainsi que le niveau n , et le retour de fonction doit être un booléen défini selon les critères ci-dessus. Si certaines règles contiennent des paramètres externes, ceux-ci sont ainsi définis comme des variables globales. L'ensemble des règles de segmentation implémentées par défaut dans l'algorithme MORFOS sont présentées en partie 3.3.2.

Une question importante est celle de la *compatibilité* entre les règles. La compatibilité entre les règles est définie comme le champ permettant de déterminer si deux règles peuvent être utilisées en simultanément. Aucune preuve logique n'a été mise en place pour démontrer l'incompatibilité entre deux règles. Une piste pourrait être de définir un certain nombre de propriétés fondamentales et de déterminer si deux règles sont contraires selon une de ces propriétés. Actuellement, la formule logique renvoyée lors du test de segmentation renvoie toujours une valeur : il s'agit donc pour nous de minimiser les résultats qui induiraient une absence de segmentation (résultat *faux* à toutes les itérations) à cause de deux règles contraires. L'idée est donc de dire lors de l'implémentation d'une règle donnée les règles pré-existantes qui ne peuvent pas être utilisées en même temps, et le cas échéant, d'indiquer quelle règle doit être utilisée. Ainsi, la compatibilité peut être définie par une matrice antisymétrique C de taille m^2 avec m le nombre de règles implémentées. Si le champ $c_{i,j}$ correspondant à la compatibilité entre la règle j et la règle i contient la valeur 0, alors les règles sont compatibles. Si le champ contient la valeur 1, alors les règles sont incompatibles et la règle i est prioritaire. A l'inverse, si le $c_{i,j} = -1$, alors les règles sont incompatibles et la règle j est prioritaire.

Ainsi, l'ajout d'une règle se passe de la manière suivante :

- 1) implémentation de la règle selon les contraintes pour la définition de la fonction
- 2) spécification du type de règle
- 3) détermination de la priorité : création d'un vecteur de priorité avec les valeurs telles que décrites précédemment de compatibilité de la règle créée par rapport à l'ensemble des règles existantes. La matrice de compatibilité est ensuite mise à

jour lors de la validation de l'ajout de la règle.

La règle est enfin ajoutée dans la *liste des règles* appelée par le gestionnaire de règle. On présente le pseudo-algorithme de la création d'une règle en 6 avec f la fonction implémentée par l'utilisateur, t le type de la règle correspondante, et v le vecteur de compatibilité avec les règles existantes.

Algorithm 6 `creer_regle`(fonction f , type t , vecteur v)

```

 $r \leftarrow \text{init\_regle}()$ 
 $\text{maj\_regle}(r, f, t)$ 
 $\text{maj\_mat\_compatibilite}(v)$ 
 $\text{ajout\_liste}(r)$ 

```

Gestionnaire de règles

La fonction de segmentation fait ainsi appel à un gestionnaire de règle. Lors de l'initialisation, le gestionnaire de règle stocke la valeur des règles sélectionnées en fonction des critères choisis par l'utilisateur. Le gestionnaire de règle vérifie ensuite en fonction des règles sélectionnées la compatibilité entre celles-ci dans la matrice de compatibilité. Alors, il retire des règles sélectionnées celles qui sont non prioritaires parmi les règles incompatibles. En effet, la désactivation s'effectue avant l'analyse et non pas à chaque fois qu'il y a ambiguïté afin de garder une cohérence sur les choix de segmentation tout au long de l'analyse. Ainsi, avec C la matrice de compatibilité et cs les critères de segmentation correspondant au sous-ensemble des règles activées pour l'analyse, l'initialisation d'une nouvelle règle de segmentation est présentée avec l'algorithme 7.

Algorithm 7 `init_segmentation`(agenda a)

```

 $cs \leftarrow \text{get\_regles\_select}(a)$ 
 $cs \leftarrow \text{maj\_regles\_select}(cs, C)$ 

```

Ensuite à chaque appel de la fonction de segmentation et en fonction des règles qui ont été activées, le gestionnaire de règles va calculer les booléens associés au résultat de l'exécution de l'implémentation de chacune des règles de segmentation dans le contexte donné. Un ensemble de booléens $\{r_i\}_{1 \leq i \leq m_1}$ est alors obtenu avec m_1 le nombre de règles sélectionnées pour l'analyse (chaque règle bivalente étant divisé en deux règles).

Afin de déterminer si il y a finalement segmentation, la formule logique suivante est calculée :

$$b = \left(\bigvee_{r_i \in \mathcal{O} \cap \mathcal{S}} r_i \right) \wedge \left(\bigwedge_{r_j \in \mathcal{I} \cap \mathcal{S}} \neg r_j \right)$$

Où \mathcal{O} est l'ensemble des règles obligatoires, \mathcal{I} l'ensemble des règles interdites et \mathcal{S} l'ensemble des règles sélectionnées et b le booléen résultant de la formule logique. Le pseudo-algorithme 8 présente la fonction de segmentation, avec $bool_regle_i(n, r, r_i)$ la fonction qui renvoie un booléen selon que l'on segmente ou non la représentation r dans le niveau n en fonction de la règle r_i et $bool(r)$ la fonction qui évalue le booléen b en fonction de l'ensemble des booléens r .

Algorithm 8 `segmentation(niveau n, representation r, critères cs) : boolean`

```
r ← ensemble(empty)
pour i in cs :
    ri ← bool_regle_i(n, r, ri)
    r ← ajouter(ri)
b ← bool(r)
retourner b
```

Une fois le booléen b calculé, les divers indices de parcours sont éventuellement mis à jour en fonction de l'évaluation des règles de segmentation (si on a par exemple un retour en arrière dans la représentation).

L'algorithme de classification

L'algorithme de classification est défini par la fonction *classification(niveau n, critères cs) : matériau*. Celle-ci consiste à ajouter dans l'oracle de niveau supérieur l'objet constitué de l'objet en voie de constitution de niveau actuel. Pour ajouter cet objet dans l'oracle, il est comparé de manière optimisée selon la construction de l'oracle présentée dans la partie précédente.

Pour cela, l'algorithme 9 est utilisé. Pour chercher le matériau correspondant à l'objet nouvellement acquis, l'objet en voie de constitution de niveau n est comparé avec le facteur de niveau n associé aux objets précédents de niveau $n + 1$, accédés par construction de l'oracle de niveau $n + 1$, et accessibles depuis la table des historiques dans la mémoire matériau de niveau n . Le calcul de similarité *similarite*($w_i, o_c^{(n)}, cc$) est appliqué sur le facteur correspondant et l'objet en voie de constitution de niveau n . Par ailleurs, si le matériau c obtenu est un nouveau matériau, la matrice d'autosimilarité de la mémoire matériau de niveau n est mise à jour avec toutes les valeurs de similarité calculées précédemment et stockées lors de l'appel à la fonction *similarite*. Par construction de l'oracle, tous les matériaux existants ont bien été comparés si l'on en crée un nouveau.

Enfin, la fonction *similarite*(*facteur f₁*, *facteur f₂*, *critères de classification*)

Algorithm 9 classification(niveau n, critères cs) : objet

pour x_i accédé dans $oracle^{(n+1)}$ en remontant les liens suffixes depuis le dernier état :

$$w_i \leftarrow Th^{(n)}(x_i)$$

$$s \leftarrow similarite(w_i, o_c^{(n)}, cc)$$
si $s = 1$:
$$c \leftarrow x_i$$
fin pour
sinon :
$$c \leftarrow creer_materiau()$$

$$maj(S^{(n)})$$

retourner c

cc) : *booleen* calcule la similarité entre deux facteurs f_1 et f_2 en fonction de critères de classification cc qui sont les règles de classification, correspondant à des mesures de similarité, sélectionnées par l'utilisateur. Si au moins deux formules sont choisies, on procède à la moyenne arithmétique des similarités $sim_k(f_1, f_2)$. Ensuite, l'algorithme vérifie si la valeur obtenue dépasse un seuil de similarité ss défini également en paramètre. Cela revient à prendre la partie entière supérieure à la différence entre la valeur de similarité et le seuil.

$$similarite(f_1, f_2, cc) = \lceil \frac{\sum_{sim_k \in cc} sim_k(f_1, f_2)}{Card(cc)} - ss \rceil$$

3.2.4 Lien signal/symbolique

L'objet est représenté jusque ici par son label. En réalité, il contient à tous les niveaux un label, et un ensemble de descripteurs correspondants : les *descripteurs concaténés* et les *descripteurs moyens*. Pour obtenir les descripteurs moyens de niveau $n + 1$ à partir du niveau n , l'algorithme calcule la moyenne entre les descripteurs qui constituent l'objet concaténé de niveau n associé à l'objet de niveau $n + 1$. Pour obtenir l'objet concaténé de niveau $n + 1$, l'ensemble des descripteurs de l'objet concaténé de niveau n est concaténé.

Si on pose $x_j^{(n+1)}$ l'objet de niveau $n + 1$ associé à l'objet concaténé $o_c^{(n)} = x_1^{(n)} \dots x_l^{(n)}$ de niveau n tel que pour tout i , $x_i^{(n)} = \{e_i^{(n)}, d_i^{(n)}\}$, avec $d_i^{(n)} = \{dc_i^{(n)}, dm_i^{(n)}\}$ où $e_i^{(n)}$ est le matériau associé, et $d_i^{(n)}$ l'ensemble des descripteurs constitué des descripteurs concaténés $dc_i^{(n)}$ et des descripteurs moyens $dm_i^{(n)}$. On définit les des-

cripteurs $dc_i^{(n)}$ et $dm_i^{(n)}$ par deux tableaux de vecteurs de taille nc , où nc est le nombre de critères tels que les mfcc ou les chromas calculés sur le signal fourni en entrée du logiciel.

On pose $dc_i^{(n)} = [vc_{i,1}^{(n)}, \dots, vc_{i,nc}^{(n)}]$, $dm_j^{(n+1)} = [vm_{j,1}^{(n+1)}, \dots, vm_{j,nc}^{(n+1)}]$ et "." l'opération de concaténation entre deux vecteurs tels que pour $v_1 = [a, b, c]$ et $v_2 = [e, f, g]$ on obtient $v = [a, b, c, d, e, f]$.

$$vc_{j,k}^{(n+1)} = vc_{1,k}^{(n)} \cdot vc_{2,k}^{(n)} \dots vc_{l,k}^{(n)}$$

$$vm_{j,k}^{(n+1)} = \frac{\sum_{i=1}^l vm_{i,k}^{(n)}}{l}$$

Si la donnée fournie en entrée est symbolique, les descripteurs sont initialisés à une valeur par défaut et ne sont pas considérés. Si la donnée fournie en entrée est audio, alors les descripteurs sont labélisés dès le niveau 0 ou le niveau 1 en fonction des critères de segmentation et de classification présentés précédemment.

Ainsi, pour chaque objet de tous les niveaux il existe un label symbolique et un ou plusieurs descripteurs audio associés.

Pour le calcul de similarité, dans le cas où l'objet contient l'information audio et symbolique, alors l'utilisateur doit choisir au moins une mesure de similarité audio et une mesure de similarité symbolique. La similarité finale est la moyenne entre les deux mesures.

$$similarite(f_1, f_2, cc) = \frac{similarite(f_1, f_2, cc(audio)) + similarite(f_1, f_2, cc(symbole))}{2}$$

Pour le calcul de segmentation dans le cas où l'objet contient l'information audio et symbolique, nous avons choisi jusqu'à présent de n'utiliser que la segmentation symbolique, mais on pourrait considérer le booléen :

$$b = b_{sig} \wedge b_{symp}$$

Avec b_{sig} le booléen obtenu après le calcul du test de segmentation selon les règles appliquées au signal et b_{symp} le booléen obtenu après le calcul du test de segmentation selon les règles appliquées au symbolique.

3.3 Règles

Pour effectuer la classification et la segmentation, on prend en compte un certain nombre de critères qui sont définis par des règles. Ces règles sont sélectionnées par l'utilisateur dans l'Agenda, et correspondent aux choix des opérations pré-implémentées dans les modules de classification et de segmentation, qui sont utilisées pour l'analyse.

3.3.1 Règles de Classification

La classification fonctionne de la manière suivante : MORFOS compare l'objet actuellement analysé avec les objets entendus précédemment en parcourant l'oracle par les liens suffixes. La similarité est calculée selon des mesures distinctes selon que l'algorithme compare des vecteurs ou bien des symboles.

Classification signal/vecteur

A partir du signal, deux traitements sont proposés dans MORFOS, et présentés dans la partie 3.2.1 : le calcul de spectre, ou le calcul de descripteurs. Chaque objet est alors représenté par un vecteur. On présente dans cette partie différentes mesures de similarité implémentées pour comparer deux vecteurs.

La Concordance Différentielle

Manuel Gaulhiac, dont la thèse [93] consiste à étudier les descripteurs harmoniques, écrit que les descripteurs dynamiques « décrivent la transition entre deux verticalités. [...] ils traitent l'aspect horizontal de l'harmonie ». Ainsi, il est possible de calculer la similarité entre deux fragments par l'obtention d'un descripteur dynamique qui traduit une forte corrélation entre ces deux fragments.

La *concordance différentielle* [93] découle de la notion de concordance harmonique. La concordance harmonique est évoquée pour la première fois dans *Esquisses pour une pensée musicale, Les métamorphoses d'Orphées* [32]. Il s'agit d'un outil d'analyse de l'harmonie : il correspond à l'extraction du *terme de couplage* (mesure de l'orthogonalité) dû au calcul de l'énergie associée à la superposition de deux sons harmoniques, à partir de leur représentation fréquentielle, c'est à dire le Spectre obtenu par Transformée de Fourier. La concordance représente ainsi une forme d'énergie *d'interaction* ou *commune* des harmonies jouées en simultanée.

Ainsi, Manuel Gaulhiac définit la concordance harmonique [93] $C(x_i^{n_1}, x_i^{n_2})$

entre deux notes simultanés d'indice n_1 et n_2 à l'instant i , $x_i^{n_1}$ et $x_i^{n_2}$ décrivant des spectres comme le produit scalaire de ces deux spectres.

Pour l'ensemble de la section décrivant les règles de similarité et de segmentation au niveau du signal, on définit les variables $x_i^{n_p}$ partielle n_p à l'instant i comme des vecteurs de dimension d tels que $x_i^{n_p} = (x_{i,1}^{n_p}, \dots, x_{i,d}^{n_p})$ et plus généralement l'objet x_i à l'instant i par $x_i = (x_{i,1}, \dots, x_{i,d})$

On a donc

$$C(x_i^{n_1}, x_i^{n_2}) = \langle x_i^{n_1}, x_i^{n_2} \rangle = \sum_{1 \leq k \leq d} x_{i,k}^{n_1} x_{i,k}^{n_2}$$

avec k l'indice des partiels contenus dans le spectre.

Alors, si un accord contient N notes, M. Gaulhiac définit la concordance à l'instant i $C(x_i)$ comme la somme des concordances des couples de notes :

$$C(x_i) = \sum_{n_1 < n_2 \leq N} C(x_i^{n_1}, x_i^{n_2}) = \sum_{n_1 < n_2 \leq N} \sum_k x_{i,k}^{n_1} x_{i,k}^{n_2}$$

La concordance différentielle correspond à l'état transitionnel de la concordance harmonique. En effet, la concordance est calculée sur chaque spectre distinct de chacune des notes d'un même accord et donc jouées en simultané. La concordance différentielle est quant à elle définie comme une mesure de la similarité spectrale entre deux fragments successifs. Elle permet ainsi de déterminer selon Jean-Marc Chouvel [32] le transfert d'énergie entre les deux spectres.

On pose donc :

$$C_{diff}(x_i, x_{i+1}) = \langle x_i, x_{i+1} \rangle = \sum_{1 \leq k \leq d} x_{i,k} x_{i+1,k}$$

Dans le cadre de MORFOS, on souhaite cependant comparer deux accords distincts et non nécessairement successifs. Il nous faut donc mesurer la concordance différentielle entre deux fragments disjoints :

$$C_{diff}(x_i, x_j) = \langle x_i, x_j \rangle = \sum_{1 \leq k \leq d} x_{i,k} x_{j,k}$$

Elle est défini alors de manière plus générale comme l'énergie commune entre deux spectres distincts.

Un certain nombre de distances sont par ailleurs proposées.

La Distance Euclidienne

La distance euclidienne permet initialement de calculer la distance entre deux points dans un espace à n dimensions (figure 3.16).

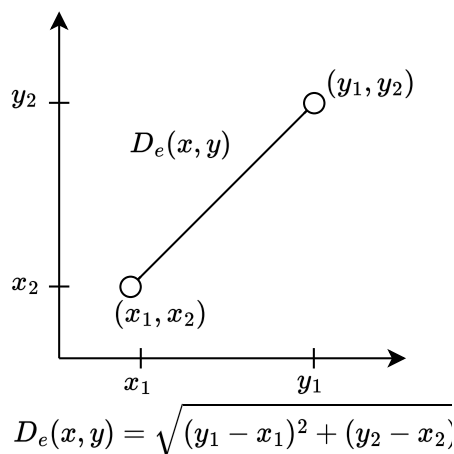


FIGURE 3.16 – Distance euclidienne.

Elle est également largement utilisée pour les méthodes d'apprentissage automatique tel que la mesure des k plus proches voisins (*k-nearest neighbors*) [52] afin de déterminer les k points les plus proches d'un point donné, ou encore pour l'analyse de données pour regrouper des points similaires dans des ensembles de données, par exemple avec l'algorithme *k-means* [90]. En informatique musicale, la distance euclidienne est utilisée par exemple pour la recherche de motifs rythmiques et du tempo par J. Foote et al. en calculant la similarité sur des représentations spectrales, représentées ensuite sous forme de matrices de similarité [54]. Elle est également utilisée pour la classification d'accords par exemple par Pickens and Crawford, qui décrivent les accords comme un vecteur de similarité avec chacune des 24 triades majeures et mineures des demi-tons de la gamme chromatique [121].

La distance euclidienne entre deux objets x_i et x_j est donc

$$D_e(x_i, x_j) = \sqrt{\sum_{1 \leq k \leq m} (x_{i,k} - x_{j,k})^2}$$

Pour obtenir une similarité euclidienne S_e , on choisit de calculer la fonction inverse de la distance euclidienne incrémentée de 1 :

$$S_e(x_i, x_j) = \frac{1}{1 + D_e(x_i, x_j)}$$

La Similarité Cosinus

La similarité cosinus permet de calculer la similarité entre deux vecteurs en calculant le cosinus de l'angle entre les deux vecteurs (figure 3.17). La valeur de 1 correspond à des vecteurs colinéaires et donc à une similarité plus grande entre les vecteurs tandis qu'une valeur de 0 correspond à des vecteurs orthogonaux. Il s'agit du produit scalaire et est donc quasiment identique à la concordance différentielle, à la différence que la similarité cosinus est normalisée par le produit des normes des deux vecteurs comparés.

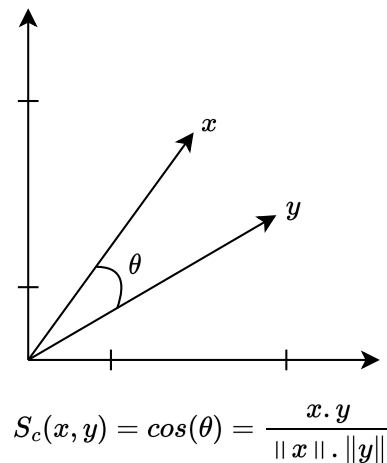


FIGURE 3.17 – Similarité cosinus.

Foote et Cooper [53] proposent également d'utiliser la similarité cosinus à partir de vecteurs correspondants aux Mel-Frequency Cepstral-Coefficient ou encore de spectres obtenus par Transformée de Fourier pour le calcul de la matrice d'autosimilarité.

La similarité cosinus entre deux objets x_i et x_j est définie de la manière suivante :

$$S_c(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \cdot \|x_j\|}$$
$$S_c(x_i, x_j) = \frac{\sum_k x_{i,k} x_{j,k}}{\sqrt{\sum_k x_{i,k}^2} \cdot \sqrt{\sum_k x_{j,k}^2}}$$

Le Dynamic Time Warping

Pour comparer deux descripteurs conservant l'aspect temporel de l'objet étudié,

l'algorithme *Dynamic Time Warping (DTW)* [7] permet de comparer deux séries temporelles et est utilisé par exemple pour la reconnaissance vocale [120] ou la synchronisation audio [108].

L'algorithme consiste à trouver un alignement, c'est à dire les correspondances point par point optimales entre les deux séquences. Pour cela, une matrice des distances M est calculée telle que pour tout $m_{i,j}$ élément de la matrice, sa valeur correspond à la distance entre l'élément d'indice i de la séquence 1 et l'élément j de la séquence 2, calculée par exemple avec la distance euclidienne. Le chemin dans la matrice allant de l'indice $0,0$ à l'indice n,m est ensuite calculé avec n la longueur de la séquence 1 et m la longueur de la séquence 2 tel que la somme des indices parcourus dans la matrice est minimale. L'algorithme est exhaustivement décrit dans [135].

Classification symbolique

Au niveau symbolique, la similarité peut être mesurée de deux manières différentes.

La première mesure vérifie la stricte égalité entre deux chaînes de caractères. Elle est défini telle que $\forall x_i, x_j \in \mathcal{A}^{(n)}, sim(x_i, x_j) = 1$ si $x_i = x_j$ et 0 sinon.

Prenons l'exemple du Mozart après acquisition du 5ème objet et segmentation au niveau 0 et avant classification :

$$\begin{aligned} r^{(0)} &= abaca \\ s^{(0)} &= (ab)(ac) \\ r^{(1)} &= a \end{aligned}$$

Les facteurs ab et ac étant différents, $sim(ab, ac) = 0$. Il n'y a donc aucun matériau existant similaire au mot acquis, un nouveau matériau b est donc créé.

Néanmoins, l'égalité stricte entre deux objets considérés comme similaires dans une pièce musicale est rare. En effet, deux motifs considérés comme similaires peuvent, en réalité, présenter une variation de note, de rythme, de tempo, de nuance, voire de transposition, qui peut être transcrite sous la forme d'un caractère différent à un niveau inférieur et donc avoir un impact sur la reconnaissance du motif à des niveaux plus élevés. Par exemple, les motifs do-do-sol-sol-mi-mi-la-la peuvent être considérés comme similaires à do-sol-mi-la pour le musicologue alors que la stricte comparaison des chaînes associées aabbccaa et abca (avec a associé

au do, b au sol et c au mi) donne à coup sûr un résultat négatif. Le même problème se posera avec l'expansion ou la réduction temporelle du même motif, ou de la *nota cambiata* qui pourrait apparaître dans certains motifs.

C'est pourquoi on propose une deuxième mesure de la similarité à partir d'une représentation symbolique qui utilise les techniques d'alignement présentées dans la partie précédente. Nous utilisons l'algorithme de Needleman-Wunsch [114]. L'algorithme de Needleman-Wunsch est un algorithme de programmation dynamique utilisé initialement pour l'alignement de séquences biologiques, notamment pour comparer et aligner des séquences d'ADN, d'ARN ou de protéines. Cet algorithme trouve l'alignement optimal entre deux séquences en maximisant un score basé sur le calcul de similarités entre les caractères qui constituent les séquences. L'algorithme de Needleman-Wunsch est le suivant :

- **Initialisation de la matrice de scores :** Une matrice M de dimensions $(l_1 + 1) \times (l_2 + 1)$, où l_1 est la longueur de la première séquence et l_2 est la longueur de la deuxième séquence est créée. L'élément $m_{i,j}$ correspond une fois rempli au score maximal de l'alignement des i premiers caractères de la première séquence avec les j premiers caractères de la deuxième séquence. Il faut dans un premier temps initialiser la première ligne et la première colonne de la matrice en fonction des pénalités associées aux *décalages* avec $m_{1,j} = m_{i,1} = gap * (j - 1)$ où *gap* est la valeur du décalage. Le décalage correspond à un score de pénalité lié à l'insertion ou la suppression d'un caractère d'une séquence par rapport à l'autre.
- **Remplissage de la matrice de scores :** La matrice M est parcourue en associant les valeurs adéquates de $m_{i,j}$ en fonction des cases voisines. Le score $m_{i,j}$ dépend des trois directions possibles : diagonale (correspondant à l'alignement des caractères, la valeur correspond donc à la somme de la similarité actuelle avec la valeur de similarité entre les deux caractères), gauche (représentant l'ajout d'un décalage dans la première séquence, la pénalité associée au décalage est ajoutée) et haut (représentant l'ajout d'un décalage dans la deuxième séquence, la pénalité associée au décalage est également ajoutée). Le score maximal est choisi parmi ces trois options.
- **Traçage du chemin optimal :** Une fois que la matrice de scores est remplie, il faut partir de l'indice m_{l_1+1,l_2+1} en suivant les valeurs maximales de chaque case pour remonter à l'indice $m_{1,1}$ pour déterminer l'alignement optimal.
- **Construction de l'alignement :** Pendant le parcours inverse, la direction indique la construction de l'alignement : un déplacement en diagonale signifie que les caractères correspondants des deux séquences sont alignés, tandis qu'un déplacement vers la gauche ou vers le haut indique l'insertion

d'un décalage dans l'une des séquences.

- **Calcul du score d'alignement** : Le score total de l'alignement est égal à la valeur m_{l_1+1, l_2+1} dans le coin supérieur gauche de la matrice.

L'algorithme de Needleman-Wunsch garantit de trouver l'alignement optimal entre deux séquences avec une complexité temporelle quadratique. On l'utilise dans le cadre de notre algorithme sur des séquences de moins d'une dizaine de caractères en général.

L'alignement global optimal obtenu avec Needleman-Wunsch inclut l'ensemble des séquences, d'une extrémité à l'autre. Cela est utile pour comparer des séquences entières, même si certaines parties sont très différentes. L'alignement local optimal obtenu avec Smith-Waterman se concentre sur une partie spécifique des séquences où l'alignement est le plus significatif en termes de similarité, ce qui permet de retrouver des régions similaires ou des motifs courts dans des séquences plus longues. Dans notre cas, c'est la distance entre deux mots bout à bout, déjà isolés par l'opération de segmentation qui est calculée : l'alignement de Needleman-Wunsch est donc le plus adéquat.

3.3.2 Règles de Segmentation

Pour rappel, les règles de segmentation sont définies en trois sous-groupes : les *règles de segmentation obligatoire*, les *règles de segmentation interdite*, et les *règles de segmentation bivalentes*. Les règles de segmentation sont également définies selon le type d'objet en entrée, qu'il soit audio ou symbolique. On présente dans cette partie les différentes règles de segmentation implémentées par défaut dans MORFOS.

Segmentations obligatoires pour le signal

Nous proposons pour les règles de segmentation pour le signal uniquement des règles de segmentation obligatoire. Ainsi, six règles permettent de contraindre la segmentation entre deux fragments vectoriels. Ces règles mesurent la dissimilarité entre deux vecteurs successifs.

La première est la *dissimilarité d'amplitude* :

La dissimilarité d'amplitude entre deux objets x_i et x_j $D_d(x_i, x_j)$ est une mesure de la différence de dynamique ou dynamique moyenne au sens de l'amplitude entre deux spectres. Nous la définissons par :

$$D_a(x_i, x_j) = |v(x_i) - v(x_j)|$$

Pour la normaliser entre 0 et 1, il est nécessaire de connaître au préalable le volume maximal v_{max} sur l'ensemble du fichier audio. On a alors la dissimilarité d'amplitude normalisée D_{an} :

$$D_{an}(x_i, x_j) = |v(x_i) - v(x_j)| / (2v_{max})$$

Plus spécifiquement, on définit la dissimilarité d'amplitude normalisée sur x_i , $D_{an}(x_i)$ comme la valeur absolue de la différence de volume entre l'objet x_i et l'objet qui le précède x_{i-1} , normalisée par le double du volume maximum du fichier audio :

$$D_{an}(x_i, x_j) = |v(x_i) - v(x_{i-1})| / (2v_{max})$$

Cette dissimilarité découle de l'hypothèse qu'un changement brusque de volume induit une segmentation. C'est le cas par exemple avec l'apparition d'un silence plus ou moins long et ayant donc une pondération plus ou moins importante dans un segment de durée fixée. Ainsi, un silence d'un quart de seconde aura un impact important à l'échelle de la note et sera plus négligeable sur la dynamique moyenne à l'échelle de la phrase musicale. En revanche, une variation de volume peut parfois à l'inverse souligner un accent important au milieu d'une phrase.

Dissimilarité d'amplitude dynamique :

La dissimilarité d'amplitude dynamique $D_{ad}(x_i, x_j)$ entre deux objets x_i et x_j consiste à mesurer l'évolution du volume entre deux objets. Nous la définissons par la valeur absolue de la différence de la dissimilarité d'amplitude calculée sur x_j et celle calculée sur x_i :

$$D_{ad}(x_i, x_j) = |D_{an}(x_i) - D_{an}(x_j)|$$

Cette mesure nous informe sur la différence de variation de volume dans le temps et pourrait ainsi apporter une information sur le mode de jeu.

Dissimilarité fréquentielle :

La dissimilarité fréquentielle $D_{fr}(x_i, x_j)$ entre deux objets x_i et x_j correspond au calcul de la différence de fréquences entre deux spectres par une mesure de la distance euclidienne sur x_i et x_j .

Cela induit la supposition qu'un fort changement en fréquence implique un changement de section. Cela peut être le cas par exemple entre deux notes, ou bien entre deux parties modulées.

L'obtention de la dissimilarité fréquentielle nécessite un pré-traitement par le calcul de la FFT ou bien de la CQT mais la distance euclidienne peut être étendu à n'importe quel descripteur calculé préalablement.

Changement harmonique :

M. Gauhliac présente dans [93] un autre calcul permettant de quantifier les différences spectrales entre deux objets x_i et x_j appelé le changement harmonique $C_h(x_i, x_j)$. Il est défini par la norme 1 du *spectre différentiel en amplitude* x , dont chaque partiel est défini comme la valeur absolue de la différence d'amplitude entre les même partiels de chaque spectre :

$$C_h(x_i, x_{i+1}) = \|x_{x_i \rightarrow x_{i+1}}^{diff}\|_1 = \sum_k |x_{i,k} - x_{i+1,k}|$$

Avec $x_{x_i \rightarrow x_{i+1},k}^{diff} = |x_{i,k} - x_{i+1,k}|$

Transformée de Fourier Différentielle

J-M. Chouvel [32] propose quant à lui la transformée de fourier différentielle $TFD(x_i, x_{i+1})$ entre deux objets successifs. Il la définit comme une quantification de l'énergie entrante et sortante liée à l'amplitude des partiels des spectres x_i et x_{i+1} . La différence avec le changement harmonique est que la valeur absolue n'est pas calculée, le signe permet alors de nous informer sur la perte ou le gain d'énergie spectrale.

$$TFD(x_i, x_{i+1}) = \sum_k (x_{i,k} - x_{i+1,k})$$

Divergence de Kullback-Leibler :

La divergence de Kullback-Leibler [81], consiste à calculer une divergence entre deux distributions de probabilité. A la différence des distances présentées précédemment, celle-ci n'est pas symétrique. Il s'agit donc de la divergence de Kullback-Leibler de x_i objet actuellement écouté par rapport à l'objet précédent x_j . Cette divergence notée $D_{kl}(x_i||x_j)$ est donnée par la formule :

$$D_{kl}(x_i||x_j) = \sum_{1 \leq k \leq m} x_{i,k} \log \frac{x_{i,k}}{x_{j,k}}$$

avec $\forall k \in [1, m], x_{j,k} \neq 0$

Une possibilité de règle de segmentation bivalente pour le signal pourrait être une contrainte sur le nombre de temps des objets grâce à l'évaluation du tempo, ou du moins dans un premier temps la mesure de la durée en secondes des objets d'un niveau donné.

Segmentations symboliques obligatoires

On présente ici un certain nombre de règles grammaticales pour les représentations symboliques. Bien que Lerdhal et Jackendoff [88] se soient attaqués à la question dans leur ouvrage *Autour de la théorie générative de la musique tonale* pour un contexte tonal, nous souhaitons proposer un ensemble de règles qui se veulent plus générales et donc les moins dépendantes possibles du style musical.

Règle 1 : hypothèse d'un objet de niveau supérieur

Cette règle consiste à segmenter lorsque le début d'un objet de niveau supérieur, c'est-à-dire le début d'un facteur de la représentation segmentée de niveau actuel, est retrouvé.

Soit $r^{(n)} = x_1 \dots x_i$ une représentation de niveau n et $s^{(n)} = w_1 \dots w_j$ la représentation segmentée de niveau n associée. Après une opération d'acquisition ou de classification d'un objet x_{i+1} , une segmentation par la règle 1 est effectuée si et seulement si :

$$\begin{aligned} \exists k \in [0, i] \text{ tel que } sim(x_{i+1}, x_k) = 1 \text{ et} \\ \exists p \in [1, j] \text{ tq } x_k = w_p[0] \end{aligned}$$

La notation $w_p[0]$ correspond au premier élément du facteur w_p .

Soit la règle de classification $sim(w_1, w_2)$ entre deux objets ou deux facteurs w_1 et w_2 qui renvoie 1 si les facteurs sont similaires et 0 sinon. On considère sur tous les exemples à venir dans cette partie la règle de classification de l'égalité stricte. On note + l'opération d'acquisition au niveau 0 ou de classification aux niveaux l avec $l > 0$ et on utilisera indépendamment le terme d'*acquisition* quelque soit le niveau. On note par ailleurs par une parenthèse fermante) l'opération de segmentation et par une parenthèse ouvrante (le début d'un facteur, partiel ou non.

Prenons alors la chaîne de caractère $r^{(n)} = ab$ ainsi que la représentation segmentée $s^{(n)} = (ab)$. Si l'algorithme acquiert un nouvel objet $x_{i+1} = x_3 = a$, on a pour $k = 1$, $sim(x_1, x_3) = sim(a, a) = 1$ et pour $p = 1$, $w_1 = x_1.b$. Le symbole a est donc le suffixe d'un facteur de $s^{(n)}$ et on segmente donc avant l'objet x_3 .

$$(ab + a \implies (ab)(a$$

Considérons maintenant l'exemple du *Rondo* de Mozart étudié précédemment :

$r^{(0)} : abacabacdeabfgabachijklmhinopqabacrsrsttu$

$s^{(0)} : (ab)(ac)(ab)(acde)(abfg)(ab)(achijklmhinopq)(ab)(acrsrsttu)$

$r^{(1)} : ABACDDAEAF$

$$\begin{aligned}
s^{(1)} &: (AB)(ACD)(AE)(AF) \\
r^{(2)} &: ABCD \\
s^{(2)} &: (ABCD) \\
r^{(3)} &: A
\end{aligned}$$

La correspondance entre le niveau 3 et le niveau 0 donne $A = (abac)$, $B = (abacdeabfg)$, $C = (abachijklmhinopq)$ et $D = (abacrsrsttu)$.

Ainsi, nous constatons que cette règle seule peut donner des objets de taille disproportionnée à des niveaux élevés (de 4 à 16 éléments). Elle n'est pas nécessairement cohérente à elle seule dans une perspective où les objets de même niveau devraient avoir une durée similaire (selon la règle 5 de *Grouping preference* dans [Lerdhal Jackendoff]).

Règle 2 : reconnaissance d'un objet de niveau supérieur

Dès que l'objet en voie de constitution du niveau actuel est un objet déjà vu au niveau supérieur, la chaîne de caractère est segmentée.

Soit $r^{(n)} = x_1 \dots x_i$ une représentation de niveau n , $o_c^{(n)} = x_k \dots x_i$ l'objet en voie de constitution de niveau l avec k l'indice de dernière segmentation tel que $k < i$ et $s^{(n)} = w_1 \dots w_j$ la représentation segmentée de niveau n . Il y a segmentation après l'opération d'acquisition ou de classification de x_{i+1} si :

$$\exists p \in [1, j] \text{ tel que } \text{sim}(w_p, o_c^{(n)}) = 1$$

Avec par exemple la représentation $r^{(n)} = ababc$ ainsi que la représentation segmentée $s^{(n)} = (ab)$ et donc l'objet en voie de constitution $o_c^{(n)} = ab$. Il existe $k = 1$ tel que $w_k = ab = o_c^{(n)}$. Il y a segmentation avant l'objet nouvellement acquis quelqu'il soit, ici le caractère c , ce qui donne :

$$(ab)(ab + c \implies (ab)(ab)(c$$

Cette règle ne peut être sélectionnée seule car elle nécessite une deuxième règle permettant de faire remonter dans un premier temps des objets au niveau supérieur. Avec l'exemple de Mozart auquel on applique la règle 1 de l'hypothèse d'un objet de niveau supérieur, on obtient :

$$\begin{aligned}
r^{(0)} &: abacabacdeabfgabachijklmhinopqabacrsrsttu \\
s^{(0)} &: (ab)(ac)(ab)(ac)(de)(ab)(fg)(ab)(ac)(hijklm)(hinopq)(ab)(ac)(rs)(rs)(t)(t)(u) \\
r^{(1)} &: ABABCADABEFABGGHHI \\
s^{(1)} &: (AB)(AB)(C)(AD)(AB)(EF)(AB)(G)(G)(H)(H)(I) \\
r^{(2)} &: AABCADAEEFFG \\
s^{(2)} &: (A)(A)(BC)(AD)(AEEFFG)
\end{aligned}$$

$r^{(3)} : \mathbb{A}\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{D}$
 $s^{(3)} : (\mathbb{A})(\mathbb{A})(\mathbb{B}\mathbb{C}\mathbb{D})$
 $r^{(4)} : \mathbb{A}\mathbb{A}\mathbb{B}$
 $s^{(4)} : (\mathbb{A})(\mathbb{A})(\mathbb{B})$
 $r^{(5)} : \mathbb{A}\mathbb{A}\mathbb{B}$

On remarque que cette règle permet d'autres segmentations par le biais de la règle d'hypothèse d'un objet de niveau supérieur. En effet, de nouveaux suffixes apparaissent tels que le h et le r et permettent de sous-segmenter les facteurs ($hijklmhinopq$) et ($rsrsttu$) précédemment obtenus. En revanche, on constate que la combinaison de ces deux règles force l'isolation de certains matériaux lors de la répétition de ceux-ci, comme les $(A)(A)$ qui sont alors remontés sur tous les niveaux supérieurs. En effet, la règle 1 d'hypothèse permet la segmentation du premier A qui est immédiatement répété, tandis que la règle 2 de reconnaissance permet la segmentation du 2e A quelque soit ce qui le succède. Ces deux A , segmentés, donneront alors toujours deux A successifs et donc segmentés aux niveaux supérieurs. Ceci n'est pas souhaitable, car alors il n'y a plus de regroupements possibles au niveaux supérieurs, et cela conforte l'incohérence de la taille des objets de plus en plus importante au fur-et-à-mesure que le niveau est élevé. Ainsi, l'algorithme de segmentation se terminerait, non pas parce que nous ne pouvons plus segmenter, mais parce que les mêmes segmentations sont obtenues aux niveaux supérieurs.

Règle 3 : reconnaissance d'un objet de niveau supérieur précédent

Pour $r^{(n)} = x_1 \dots x_i$ et $s^{(n)} = w_1 \dots w_j$, il y a segmentation lorsque il existe $p \in [1, j]$ tel que l'objet en voie de constitution de niveau actuel $o_c^{(n)}$ est similaire à un préfixe p_1 de taille minimum 2 d'un facteur w_p de la représentation segmentée $s^{(n)}$. Il y a segmentation alors du facteur w_p en deux sous facteurs distincts p_1 et p_2 tels que $w_p = p_1.p_2$ et l'algorithme modifie les objets de niveaux supérieurs en conséquence.

Il y a donc segmentation si :

$$\exists k_1, k_2 \in [1, i - |o_c^{(n)}|] \text{ avec } k_1 < k_2 \text{ et } \exists p \in [1, j] \text{ tel que}$$

$$\text{sim}(x_{k_1} \dots x_{k_2}, o_c^{(n)}) = 1 \text{ et } x_{k_1} \dots x_{k_2} \cdot [w_p]_{k_2+1, |w_p|} = w_p$$

Alors, l'algorithme en entier est ré-exécuté, en forçant la segmentation juste avant l'acquisition de l'objet x_{k_2+1} et on déroule également à nouveau les opérations jusqu'à l'acquisition de x_{i+1} . La reconnaissance du facteur $w = x_{k_1} \dots x_{k_2}$ de $s^{(n)}$ tel que $\text{sim}(w, o_c^{(n)}) = 1$ permet ainsi la segmentation avant x_{i+1} également. Cette règle peut avoir un impact rétro-actif également sur les niveaux supérieurs obtenus.

Une possibilité aurait pu être de segmenter uniquement juste avant x_{i+1} et

non pas au niveau de x_{k_2+1} et de considérer que l'écoute étant en temps réel, la modification est effective uniquement pour l'objet écouté actuellement. Nous avons fait le choix de considérer que l'apparition de cette nouvelle information modifie rétrospectivement le modèle de la forme musicale construite lors de l'écoute, selon un phénomène pouvant se rapporter au modèle de révision orwellienne dans la conscience [45].

Ainsi, prenons par exemple la chaîne de caractère $r^{(n)} = abcdab$ ainsi que la représentation segmentée $s^{(n)} = (abcd)$ et donc l'objet en voie de constitution $o_c^{(n)} = ab$. On constate que $k_1 = 1$ et $k_2 = 2$ tel que $sim(x_{k_1}x_{k_2}, o_c^{(n)}) = sim(ab, ab) = 1$. Il y a segmentation donc après l'acquisition du nouvel objet x_{i+1} avant cet objet quelqu'il soit. En l'occurrence, après l'acquisition de $x_{i+1} = e$,

$$(abcd)(ab + e \implies (ab)(cd)(ab)(e$$

Par ailleurs, la chaîne *abacabacdeabfgabachijklmhinopqabr sr sttu* structurée avec les trois règles présentées donne les résultats suivants :

$$\begin{aligned} r^{(0)} &: abacabacdeabfgabachijklmhinopqabr sr sttu \\ s^{(0)} &: (ab)(ac)(ab)(ac)(de)(ab)(fg)(ab)(ac)(hi)(jklm)(hi)(nopq)(ab)(ac)(rs)(rs)(t)(t)(u) \\ r^{(1)} &: ABABCADABEFEGABHHIIJ \\ s^{(1)} &: (AB)(AB)(C)(AD)(AB)(EF)(EG)(AB)(H)(H)(I)(I)(J) \\ r^{(2)} &: AABCADDEAF FGGH \\ s^{(2)} &: (A)(A)(BC)(A)(DE)(A)(F)(F)(G)(G)(H) \\ r^{(3)} &: AABACADDEEF \\ s^{(3)} &: (A)(A)(B)(A)(C)(A)(D)(D)(E)(E)(F) \\ r^{(4)} &: AABACADDEEF \end{aligned}$$

La différence entre la règle 2 de *reconnaissance d'un objet de niveau supérieur* et la règle 3 de *reconnaissance d'un objet de niveau supérieur précédent* dans cet exemple est la structure de *(hi)* avec la règle 3 alors que la règle 2 donne les deux objets *(hijklm)* et *(hinopq)*. Cependant, comme dans l'exemple précédent, les objets immédiatement répétés sont immédiatement structurés comme des matériaux de niveau supérieur avec cette règle. Cela induit l'isolement de plus encore de matériaux qui ne peuvent plus être inclus dans un matériau plus large. Cela implique également que tous les niveaux supérieurs contiendront exactement les mêmes matériaux de cette longueur donnée.

Ainsi, il est nécessaire de définir des règles qui empêchent la segmentation selon un contexte donné.

Segmentations symboliques interdites

Règle 4 : objet isolé

Cette règle est destinée à éviter de retrouver des objets isolés qui remonteraient aux niveaux supérieurs : un objet de niveau supérieur isolé, c'est-à-dire correspondant à un mot de niveau actuel de longueur 1, n'est pas segmenté (Grouping Preference Rule 1 dans [88]). Cela signifie que la segmentation est interdite après l'acquisition de l'objet x_{i+1} si :

$$|o_c^{(n)}| = 1$$

Par exemple, prenons la chaîne de caractère $r^{(n)} = aba$ ainsi que la représentation segmentée $s^{(n)} = (ab)$ et donc l'objet en voie de constitution $o_c^{(n)} = a$. Alors $|o_c^{(n)}| = 1$ donc pour i la longueur de $r^{(n)}$, quelque soit l'objet x_{i+1} acquis, il n'y a pas de segmentation. Pour $x_{i+1} = a$:

$$(ab)(a + a \implies (ab)(aa$$

La chaîne *abacabacdeabfgabachijklmhinopqabacrsrsttu* segmentée avec les quatre règles $\{1, 2, 3, 4\}$ donne les résultats suivants :

$r^{(0)}$: *abacabacdeabfgabachijklmhinopqabacrsrsttu*

$s^{(0)}$: *(ab)(ac)(ab)(ac)(de)(ab)(fg)(ab)(ac)(hi)(jklm)(hi)(nopq)(ab)(ac)(rs)(rs)(ttu)*

$r^{(1)}$: *ABABCADABEFEGABHHI*

$s^{(1)}$: *(AB)(AB)(CAD)(AB)(EF)(EG)(AB)(HHI)*

$r^{(2)}$: *AABACDAE*

$s^{(2)}$: *(AAB)(ACD)(AE)*

$r^{(3)}$: *ABC*

$s^{(3)}$: *(ABC)*

$r^{(4)}$: *Ⓐ*

Il s'agit de la segmentation des exemples présentés en figure 2.3 dans le Chapitre 2. Les objets ont des longueurs à peu près équivalentes pour chacun des niveaux. Cependant, cette règle peut ne pas être pertinente. En effet, les fragments isolés sont intégrés dans le facteur suivant alors qu'il peut être plus pertinent de les intégrer dans le facteur précédent ou encore de garder cet objet isolé, par exemple, s'il est fortement isolé par un silence.

Règle 5 : objet de niveau supérieur en voie de construction

Si l'objet en voie de constitution de niveau actuel est un préfixe d'un facteur de la représentation segmentée de niveau actuel (règle de reconnaissance d'un objet de niveau supérieur précédent), mais que l'objet acquis est également identique à l'objet qui suit le préfixe alors on ne segmente pas. En d'autres termes, la segmentation

est interdite après l'acquisition de l'objet x_{i+1} si :

$$\exists k_1, k_2 \in [1, i - |o_c^{(l)}|] \text{ avec } k_1 < k_2 \text{ et } \exists p \in [1, j] \text{ tel que}$$

$$\text{sim}(x_{k_1} \dots x_{k_2}, o_c^{(l)}) = 1 \text{ et } x_{k_1} \dots x_{k_2} \cdot [w_p]_{k_2+1, |w_p|} = w_p$$

et

$$x_{k_2+1} = x_{i+1}$$

Prenons un exemple avec la chaîne de caractère $r^{(n)} = abcdab$. On suppose que $s^{(n)} = (abcd)$ et donc $o_c^{(n)} = ab$. Alors, $k_1 = 1$ et $k_2 = 2$ tel que $\text{sim}(x_{k_1} x_{k_2}, o_c^{(n)}) = \text{sim}(ab, ab) = 1$. Cependant, si l'objet acquis est $x_{i+1} = c$ avec i la longueur actuelle du mot, c'est-à-dire 6, alors $x_{k_2+1} = x_3 = c$ et $x_{i+1} = x_7 = c$ donc $x_{k_2+1} = x_{i+1}$: il n'y a pas de segmentation avant le septième objet.

$$(abcd)(ab + c \implies (abcd)(abc$$

Segmentation bivalente symbolique

Règle 6 : contrainte sur la taille absolue

Soit deux limites fixées f_i et f_s . Si la longueur de l'objet en voie de constitution est inférieure à la limite inférieure f_i alors la segmentation est interdite, tandis que si la longueur de l'objet en voie de constitution est égale à la limite supérieure f_s alors la segmentation est obligatoire :

$$\text{Segmentation interdite si } |o_c^{(n)}| < f_i$$

$$\text{Segmentation obligatoire si } |o_c^{(n)}| = f_s$$

Prenons un exemple avec $r^{(n)} = ab$, $o_c^{(n)} = ab$ et $f_i = 4$. Il y a $|o_c^{(n)}| = 2$ donc $|o_c^{(n)}| < f_i$: quelque soit l'objet acquis par la suite, la segmentation est interdite. Si l'objet $x_3 = a$ est acquis alors :

$$(ab + a \implies (aba$$

Prenons par ailleurs l'exemple $r^{(n)} = abcdeabc$, $o_c^{(n)} = abcdeabc$ et $f_s = 8$. On a $|o_c^{(n)}| = 8 = f_s$ donc quelque soit l'objet acquis par la suite, il y a segmentation. Si l'objet $x_9 = d$ est acquis, alors :

$$(abcdeabc + d \implies (abcdeabc)(d$$

Néanmoins, cette règle dépend de valeurs arbitraires définies en amont de l'analyse. On pourrait choisir de définir la longueur des mots en fonction des segmentations qui ont précédemment eu lieu au cours de l'analyse. C'est ce qui nous mène à la définition de la règle suivante.

Règle 7 : contrainte sur la taille relative

On définit $A(n)$ comme la longueur moyenne des facteurs de la représentation segmentée de niveau n .

$$A(n) = \frac{\sum_{k=1}^j |w_k|}{j} \text{ avec } s^{(n)} = w_1 \dots w_j$$

Si la longueur de l'objet en voie de constitution est inférieur à $A(n)/2$ alors la segmentation est interdite, tandis que si la longueur de l'objet en voie de constitution est égale à $A(n) + A(n)/2$ alors la segmentation est obligatoire :

Segmentation interdite si $|o_c^{(n)}| < A(n)/2$

Segmentation obligatoire si $|o_c^{(n)}| = A(n) + A(n)/2$

Alors, la première segmentation est nécessairement induite par une règle externe.

Prenons un exemple avec $r^{(n)} = abcda$, $s^{(n)} = (abcd)$ et $o_c^{(n)} = a$. Il y a toujours $A(n) = 4$ et $|o_c^{(n)}| = 1$ donc $|o_c^{(n)}| < A(n)/2$: quelque soit l'objet acquis par la suite, la segmentation est interdite. Si l'objet $x_6 = a$ est acquis :

$$(abcd)(a + a \implies (abcd)(aa$$

Prenons par ailleurs l'exemple $r^{(n)} = abcdabcefg$, $s^{(n)} = (abcd)$ et $o_c^{(n)} = abcefg$. Alors $A(n) = |abcd| = 4$ et $|o_c^{(n)}| = 6 = A(n) + A(n)/2$ donc quelque soit l'objet acquis par la suite, il y a segmentation. Si l'objet $x_11 = h$ est acquis, alors :

$$(abcd)(abcefg + h \implies (abcd)(abcefg)(h$$

Règle 8 : règle de groupement par similarité

Si l'objet en voie de constitution contient au moins un objet et que le dernier objet qui le constitue est similaire à l'objet acquis alors il n'y a pas segmentation entre les deux objets. Si l'objet en voie de constitution contient au moins deux objets, que les deux derniers objets qui le constituent sont similaires et que le dernier objet est différent de l'objet acquis alors il y a segmentation. On définit mathématiquement cette règle par :

Segmentation interdite si $\text{sim}(x_i, x_{i+1}) = 1$

Segmentation obligatoire si $|o_c^{(n)}| \geq 2$, $\text{sim}(x_{i-1}, x_i) = 1$ et $\text{sim}(x_i, x_{i+1}) = 0$

Prenons pour exemple $r^{(n)} = a$ et $o_c^{(n)} = a$. Si pour $i = 1$ l'objet $x_{i+1} = a$ est acquis, alors $\text{sim}(x_i, x_{i+1}) = \text{sim}(a, a) = 1$ donc la segmentation est interdite entre les deux objets :

$$(a + a \implies (aa$$

Prenons également l'exemple $r^{(n)} = aa$ et $o_c^{(n)} = aa$. Si pour $i = 2$ l'objet $x_{i+1} = b$ est acquis, on a alors $\text{sim}(x_{i-1}, x_i) = \text{sim}(a, a) = 1$ et $\text{sim}(x_i, x_{i+1}) = \text{sim}(a, b) = 0$. Il y a alors segmentation entre x_i et x_{i+1} :

$$(aa + b \implies (aa)(b$$

3.4 Paramétrisation

Avant toute analyse, nous demandons à l'utilisateur de fournir un ensemble de paramètres permettant d'orienter le type d'analyse fournie par le logiciel. Passons en revue les paramètres demandés à l'utilisateur lors de l'analyse d'une œuvre musicale. Ceux-ci se découpent en trois groupes de paramètres. Le premier groupe est constitué des paramètres concernant l'Agenda : il contient les données à analyser, les choix de calcul pour le pré-traitement de ces données, les critères de segmentation ainsi que les critères de classification. Le deuxième groupe sont les paramètres associés au calcul de statistiques, ainsi que des paramètres liés à l'affichage et à la représentation. Ces paramètres seront détaillés plus amplement dans les chapitres 4 et 5.

Parmi les paramètres de l'Agenda, nous distinguons également deux types de paramètres. Les premiers correspondent à des paramètres que l'utilisateur doit fournir obligatoirement : il s'agit des critères généraux de l'analyse, permettant de passer d'un traitement objectif à un traitement subjectif de l'écoute. Le deuxième groupe de paramètres correspond à des hyper-paramètres qui sont des affinages des paramètres précédents : des valeurs sont proposées par défaut pour chacun des paramètres choisis par l'utilisateur mais celui-ci peut régler spécifiquement une partie d'entre eux.

3.4.1 Données à analyser

Tout d'abord, le premier paramètre fondamental à l'analyse est le fichier d'entrée. Celui-ci peut être au format mp3, wav, csv, txt, npy ou MIDI. Le format est analysé automatiquement afin de proposer l'algorithme et les critères d'analyse adaptés aux représentations de données.

3.4.2 Critères de pré-traitement

Si le fichier fourni en entrée est au format vectoriel ou symbolique, il n'y a alors pas de paramètres à fournir pour le pré-traitement. Dans le cas d'un fichier audio, l'utilisateur choisit quelle est l'information qu'il souhaite que la représentation vectorielle contienne. Les traitements suivants sont proposés et détaillés dans la section 3.2 :

- spectrale par calcul de la fft ;
- spectrale par calcul de la cqt ;
- mfcc ;
- chromas.

Hyper-paramètres du pré-traitement

Selon le type d'analyse choisi pour le signal, il y a un certain nombre de sous-paramètres adéquats à définir.

Tout d'abord, il y a la *fréquence d'échantillonnage* : Si la fréquence d'échantillonnage choisie est trop faible, le signal peut être confondu avec un signal d'une autre fréquence. C'est le phénomène de *repliement de spectre (aliasing)*. Pour pallier cela, le Théorème de Nyquist-Shannon [147] : assure de pouvoir reconstituer exactement un signal périodique numérisé il faut échantillonner ce signal à une fréquence au moins deux fois supérieure à la fréquence du signal analogique. L'audition humaine étant entre 20 Hertz et 20000 Hertz, il faut donc échantillonner à une fréquence minimale de 44100 Hertz. Cependant, nous avons choisi de considérer l'analyse des fréquences au dessus de 10000 Hertz comme négligeables pour pouvoir procéder à des analyses à partir de 22050 Hertz. La fréquence d'échantillonnage choisie par défaut est 44100 Hertz.

Il faut également choisir une taille pour la fenêtre d'analyse. Celle-ci détermine alors la durée correspondante à l'échantillon analysé. On choisit une taille multiple d'une puissance de 2 : par défaut, la taille de la fenêtre d'analyse est de 2048 échantillons. On a $T = n_e / f_e = 2048 / 44100 = 0.046$ secondes.

Par ailleurs, si la FFT est choisie comme méthode d'analyse, il faut préciser

alors l'implémentation parmi les 3 proposées que sont la *fft classique*, la *fft par bandes* et la *fft par fenêtre*. Par défaut, la *fft par fenêtre* est sélectionnée. En fonction, il faut alors également préciser la précision entre chaque note.

Si le calcul des MFCC est choisi, il faut alors fixer le nombre de partielles à calculer ainsi que la normalisation ou non des partielles. Par défaut, on propose 50 partielles et une MFCC non normalisée.

Enfin pour le calcul de la CQT, le nombre total de notes doit être spécifié, ainsi que le nombre de notes par octave ainsi que la note minimale pour l'analyse. Les valeurs choisies par défaut sont respectivement de 48 notes dont 12 par octaves et le Do 3 en note minimale.

3.4.3 Critères de similarité

Les critères de similarité et de segmentation sont également choisis lors de la définition de l'Agenda. Ils se décomposent en deux ensembles de paramètres : le premier correspond aux règles. L'utilisateur choisit donc un sous ensemble de règles parmi les règles proposées.

Pour les règles de similarité, il choisit une ou plusieurs opérations qui permettent de calculer la similarité entre deux objets, au niveau du signal si nécessaire et au niveau symbolique.

Les règles de similarité renvoient un chiffre réel entre 0 et 1 : c'est pourquoi nous demandons également à l'utilisateur de choisir des seuils, un pour la segmentation, et un pour la similarité. La valeur de ces seuils est entre 0 et 1 et correspond à la capacité de l'utilisateur à distinguer deux objets. Plus le seuil de similarité est proche de 0 et plus les objets seront classés dans une unique catégorie, plus ce seuil est proche de 1 et plus les objets auront tendance à être classés dans des catégories distinctes.

Une possibilité peut être de calculer automatiquement un seuil optimal de similarité tel que le *Information Rate* présenté par S. Dubnov, G. Assayag et A. Cont dans [48].

3.4.4 Critères de segmentation

Pour les règles de segmentation, l'utilisateur choisit distinctement pour le signal et pour l'audio un sous-groupe de règles à appliquer parmi un ensemble de règles implémentées. Un ordre de priorité existe par défaut au cas où des règles exclusives

seraient choisies en même temps.

Pour la segmentation symbolique, les règles de segmentation retournent un booléen. Pour la segmentation sur un vecteur, la fonction renvoie un réel entre 0 et 1. Pour définir s'il y a segmentation ou non, on définit également un seuil de segmentation. Plus le seuil sera élevé, plus la valeur de segmentation devra être élevée pour procéder à une segmentation, et donc moins on aura de segmentation globalement sur l'extrait.

Conclusion

MORFOS est constitué de quatre modules fondamentaux présentés dans ce chapitre : l'*algorithme MORFOS*, les *structures de données*, la *visualisation du diagramme formel multi-échelle* ainsi que l'*interface des paramètres*. Ces modules sont implémentés en fonction de la formalisation de MORFOS. Trois autres modules sont indépendants de l'algorithme et proposent des opérations afin de traiter les données en entrée de l'algorithme. Il s'agit des modules de *pré-traitement des données*, des *règles de segmentation* et des *règles de similarité*.

Ces trois derniers modules présentent l'implémentation de différentes opérations choisies par l'utilisateur à travers le module d'interface des paramètres. Ainsi, le pré-traitement des données peut être effectué afin d'obtenir un spectre d'amplitude, soit par calcul de la *Transformée de Fourier*, dont trois optimisations différentes sont proposées, soit par le calcul de la *Transformée à Q Constant*. Le logiciel permet également de calculer divers descripteurs, tels que les *Mel Frequency Cepstral Coefficients* ou les *chromas*.

Les règles de classifications implémentées sont par ailleurs pour le signal la *Concordance différentielle*, la *Distance Euclidienne* ainsi que la *Similarité Cosinus*. On propose une implémentation du *Dynamic Time Warping* pour prendre en compte l'aspect temporel des données mais cette mesure n'a pas été testée. Au niveau symbolique, une mesure de *stricte égalité* entre de séquences de caractères, ou encore l'algorithme d'*alignement* de Needleman-Wunsch sont implémentés.

Les règles de segmentation sont divisées en règles pour les représentations signal et pour les représentations symboliques, mais elles sont également divisées en trois types de règles : les *règles de segmentation obligatoire*, qui forcent la segmentation, les *règles de segmentation interdite*, qui interdisent la segmentation et les *règles de segmentation bivalentes*, qui possèdent des propriétés forçant la segmentation dans un contexte donné et l'interdisant dans un autre contexte. Les règles

proposées pour les représentations signal sont toutes des règles de segmentation obligatoire : la *Dissimilarité d'Amplitude*, la *Dissimilarité d'Amplitude Dynamique*, la *Dissimilarité Fréquentielle*, le *Changement Harmonique*, la *Transformée de Fourier Différentielle* et la *Divergence de Kullback-Leibler*. Au niveau symbolique, nous présentons 8 règles grammaticales. Trois règles de segmentation obligatoire basées sur la reconnaissance d'un objet de niveau actuel ou supérieur, deux règles de segmentation interdite basées sur la construction en cours d'un objet de niveau supérieur ou sur l'isolement d'un matériau et trois règles de segmentation bivalente fondées sur des contraintes de taille de l'objet en voie de constitution ou de regroupement de matériaux similaires.

La boucle principale de l'algorithme MORFOS est également présentée : celle-ci est construite sur un principe de récursivité. A chaque acquisition d'un nouveau matériau, les structures de données sont mises à jour et un test de segmentation est effectué. S'il y a segmentation, alors le test de classification de l'objet de niveau supérieur est effectué lors de l'acquisition de l'objet au niveau supérieur où l'algorithme est appelé à nouveau. Une modélisation d'un gestionnaire des règles de segmentation, construit sur des formules logiques est également présenté.

Chapitre 4

Un Modèle d'Écoute : coûts et cognition

L'Algorithme Cognitif est un modèle de l'écoute et non pas une reproduction fidèle du système cognitif lié à l'écoute musicale. On s'intéresse aux phénomènes cognitifs liés à l'écoute musicale sur trois catégories : la tâche de classification et segmentation, les phénomènes d'attention et la charge cognitive.

La segmentation musicale est le découpage de la pièce en plusieurs segments. Comme vu précédemment, cette segmentation peut être à différentes échelles et est définie ici par l'indication de la fin d'une section : par défaut la fin d'une section indique le début d'une nouvelle. La classification correspond au regroupement des objets selon des caractéristiques communes. Dans notre cas le regroupement est dépendant de l'échelle d'analyse à laquelle on se situe et, la musique étant un phénomène complexe, de la projection sur laquelle on filtre l'objet que l'on souhaite classer. On définit les tâches de segmentation et de classification comme une analyse musicale de premier ordre. Cette analyse nécessite des coûts de calcul qui induisent une certaine complexité, qui dépendent de la structure créée lors de la classification et de la segmentation.

Nous référons les phénomènes d'attention au sens des mécanismes associés au phénomène de la conscience lors de l'écoute musicale. L'attention est définie¹ comme *"la modalité de la conscience : "être conscient" c'est être attentif. L'attention est ce qui constitue les objets de la conscience, même si toute conscience n'est pas attentive – toute attention étant évidemment consciente."* Le temps associé à l'attention se situe entre la *réention*, la mémoire qui retient l'objet qui

1. Ars Industrialis <https://arsindustrialis.org/vocabulaire-attention-retention-protention>

a été acquis, et la *protention*, qui est l'attente de l'objet à venir [70]. Pour comprendre ces phénomènes, on procède à une analyse de second ordre qui nécessite d'étudier les comportements associés notamment aux phénomènes de classification et de segmentation.

La charge cognitive est définie par J. Sweller (1994) [155] comme la quantité d'effort nécessaire pour effectuer une tâche donnée : par exemple, la tâche de segmentation et de classification, qui peut être explicite, mais également les tâches de protension et de rétention, lié au schéma cognitif de l'individu. On cherchera ainsi à comprendre un peu mieux les phénomènes liés à la charge cognitive lors de la segmentation pendant l'écoute active d'une œuvre musicale.

Par ailleurs, nous faisons les liens suivants entre les phénomènes cognitifs et ceux du logiciel : les processus de segmentation et de classification donnent lieu à une représentation, qui est pour MORFOS la représentation hiérarchisée ainsi que le diagramme formel multi-échelle qui en découle. Les phénomènes d'attention peuvent être comparés au comportement algorithmique de MORFOS, tandis que la charge cognitive pourrait être associée à la charge computationnelle, c'est-à-dire la complexité de MORFOS. Cette mise en perspective est résumée en figure 4.1.

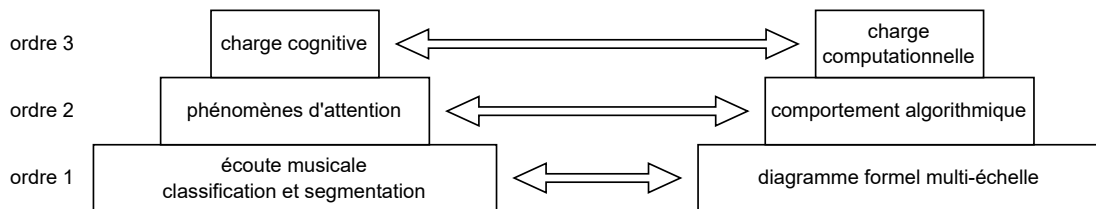


FIGURE 4.1 – équivalences entre les phénomènes cognitifs et MORFOS.

La figure 4.2 présente les phénomènes cognitifs liés à l'écoute musicale dans le cadre de l'algorithme cognitif

D'autres facteurs, tels que les émotions, le contexte du moment d'écoute ou la connaissance externe entrent en jeu et ne sont pas pris en compte dans cette thèse.

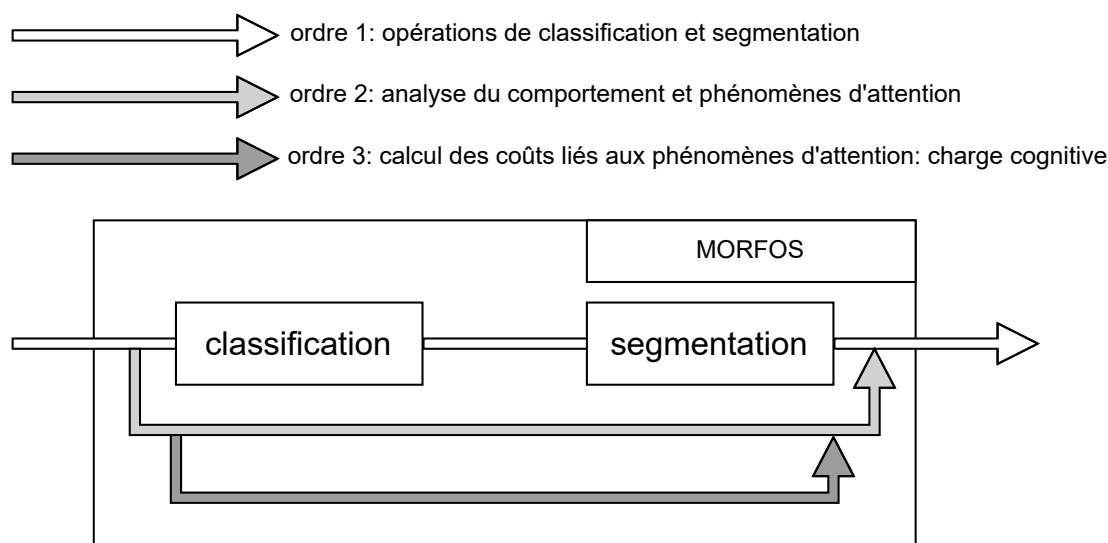


FIGURE 4.2 – Phénomènes cognitifs liés à l’écoute musicale : la classification/segmentation, les phénomènes d’attention et la charge cognitive.

4.1 Bibliographie sur la modélisation des phénomènes cognitifs

Bien que ces notions soient intrinsèquement liées, nous divisons ce chapitre en trois sections découpées selon les ordres présentés ci-dessus que sont la structure obtenue par les opérations de classification et segmentation, les phénomènes d’attention et la charge cognitive. On présente une brève revue des modèles existants.

4.1.1 Structure et Complexité

La complexité peut être utilisée pour expliquer la structure sous-jacente au phénomène de l’écoute musicale. Il existe deux types de calculs pour expliquer un objet musical en fonction de son contexte. La première est l’entropie statistique au sens de la Théorie de l’Information de Shannon [146]. Il s’agit d’une distribution statistique : on définit une fonction de probabilité de l’objet à venir en connaissant son contexte. Seulement, la distribution statistique choisie dépend de différents facteurs, notamment culturels, tels que le genre musical. Étant donné que l’on souhaite définir un logiciel le plus général possible, une telle mesure n’est pas adéquate pour nos calculs. L’autre mesure est la complexité telle que définie par

Kolmogorov [80] : combien d'opérations faut-il pour obtenir un objet selon son contexte? Alors, il faut fournir une bibliothèque de transformation et des coûts. C. Guichaoua [63] et V. Gillot [59] s'inspirent de ces concepts dans leurs thèses respectives afin d'étudier l'inférence des critères de complexité pour la définition de la structure musicale sur des représentations hiérarchisées. Nous nous pencherons dans une première section sur la question de la qualité de la structure en fonction de la complexité mesurée selon les principes définis par la complexité de Kolmogorov.

Une deuxième question relative à celle-ci et qui n'est pas approfondie dans ce manuscrit, est la question de la compression. Différents algorithmes permettent de diminuer l'espace pris par l'information à encoder : soit avec perte d'information comme l'algorithme JPEG [119], soit sans perte d'information, tel que l'algorithme de codage d'Huffman [68], ou encore les différentes versions de Lempel-Ziv [167], [166]. Nous avons pu ébaucher l'idée dans le chapitre précédent que l'algorithme MORFOS, permet par la factorisation des objets musicaux en matériaux, de représenter une forme compressée de l'œuvre musicale. Par ailleurs, la structure hiérarchisée apporte une deuxième passe de compression possible, où le retour à une partie de niveau supérieur ne nécessite pas d'être décrite à nouveau dans ses niveaux inférieurs. On présente des algorithmes pour représenter une version compressée des diagrammes formels multi-échelles en annexe B. Une étude plus approfondie pourrait être effectuée afin de déterminer d'une part si l'algorithme MORFOS offre une compression concurrençant les algorithmes présentés précédemment, et avec quel taux de perte. Par ailleurs, il s'agirait de regarder à quel point une œuvre dite ici "bien structurée", c'est-à-dire qui présenterait une complexité de Kolmogorov moindre, permet ou pas une meilleure compression de l'information.

4.1.2 Les phénomènes d'attention

Ainsi, on souhaite définir les phénomènes structurants par des calculs de complexité, et à les expliquer par les modèles cognitifs d'attention et de structuration lors de l'écoute musicale. Inspiré de l'application des lois de la *Gesstalt* par L. Meyer [103], E. Narmour présente dans [109] [110] le modèle Implication-Réalisation. Ce modèle vise à présenter les mécanismes d'attention à partir de ses aspects cognitifs lors de l'écoute musicale. Deux affirmations concernant ce modèle sont, d'une part, la réalisation ou non des deux hypothèses suivantes : (1) $A + A \rightarrow A$ et (2) $A + B \rightarrow C$ où A, B et C sont des groupements musicaux. Ainsi, l'hypothèse (1) signifie que la répétition du même motif musical implique une répétition successive de ce motif tandis que l'hypothèse (2) signifie que la succession de deux motifs musicaux distincts implique un troisième motif distinct. La deuxième affirmation définit que les schémas précédemment décrits impliquent une clôture ou non de

la phrase musicale. Le modèle Système & Constraste [15] présenté précédemment (section 2) s'inscrit dans la continuité des recherches de E. Narmour en décrivant l'organisation interne et les relations entre les propriétés d'éléments musicaux regroupés en carrés de quatre éléments. J.M. Chouvel, présente quant à lui dans [31] une analyse du comportement de l'Algorithme Cognitif, donnant lieu à quatre phases secondaires, correspondant aux quatre possibilités des résultats des tests de classification et de segmentation dans le circuit cognitif. C'est cette modélisation que nous allons expliciter dans une section suivante et mettre en perspective avec les résultats du logiciel.

4.1.3 La charge cognitive

Ainsi, ces différentes mesures sont une piste à comparer avec la charge cognitive lors de l'écoute musicale, dans l'objectif de pouvoir mieux l'expliquer. Selon J. Sweller [155], la charge cognitive est divisée en trois catégories : la charge intrinsèque, la charge extrinsèque et la charge essentielle. La charge intrinsèque correspond à la combinatoire associée au nombre d'interactions entre les éléments d'une tâche donnée, en fonction de la connaissance de l'individu [154]. La charge extrinsèque est externe à la tâche même et dépend de la manière dont l'information est présentée à l'utilisateur. La suppression de la redondance permet par exemple de diminuer la charge extrinsèque. Tandis que la charge essentielle est mise en œuvre afin d'intégrer l'information à long terme sous forme de schémas mentaux.

Pour les processus liés à l'écoute musicale, la tâche même de la segmentation musicale associée constitue la charge cognitive intrinsèque, dont on émet l'hypothèse qu'elle dépend de la quantité d'information dans l'œuvre d'une part, et de l'expérience musicale de l'individu d'autre part. La charge cognitive extrinsèque est quand à elle liée à l'interface proposée aux individus pour effectuer la tâche, dans le cadre d'une expérience et notre objectif en tant que chercheurs est de proposer une interface adaptée et des instructions claires permettant aux utilisateurs de diminuer cette charge. Enfin, la charge cognitive essentielle va dépendre de la manière dont les utilisateurs procèdent pour effectuer la tâche demandée : une phase d'entraînement en début d'expérience devrait également permettre de diminuer cela, mais elle peut être propre à chacun des individus en fonction des schémas mis en place.

Nous avons ainsi conçu et réalisé une expérience cognitive que nous décrivons dans une troisième section afin de mesurer la charge cognitive liée à la segmentation lors de l'écoute musicale, dans l'idée de la comparer à l'avenir aux coûts computationnels de MORFOS.

4.2 Complexité

On cherche à définir et à qualifier la structure des diagrammes formels multi-échelles. Pour cela, nous définissons une mesure de la complexité en fonction de critères maximisant ou minimisant la valeur de la complexité. On explicitera celle-ci ensuite selon la définition de la complexité de Kolmogorov en fonction du nombre d'opérations nécessaires pour expliquer l'acquisition d'un nouvel objet en fonction d'un contexte donné. Comme nous constatons que le nombre d'opérations effectuées est proche du calcul de similarité par alignement tel que proposé au chapitre 3, nous étendons le calcul de la complexité à un calcul de similarités de manière générale. Enfin, nous chercherons ainsi à comparer les segmentations proposées par MORFOS avec d'autres segmentations arbitraires ou aléatoires. Nous espérons alors que la complexité pour une segmentation optimisée soit minimisée.

Nous proposons un ensemble de considérations permettant de déterminer quels sont les facteurs intervenant dans le calcul de la complexité.

- Nous procédons à une analyse agnostique de l'œuvre, nous n'avons donc pas de connaissances spécifiques sur la structure musicale vis-à-vis par exemple de son genre, de son époque ou de son instrumentation. Les règles de similarité et de segmentation proposées se veulent les plus générales possibles.
- La représentation de la structure que nous étudions est hiérarchisée selon le modèle de l'Algorithme Cognitif présenté par Jean-Marc Chouvel : ainsi, les segmentations des niveaux supérieurs dépendent de celles des niveaux inférieurs.
- Selon le point de vue de l'écoute, le déroulement temporel de l'œuvre est primordial : on calcule donc la complexité d'un objet uniquement à partir du contexte des éléments qui le précèdent.
- L'hypothèse posée actuellement est de garder en mémoire tous les matériaux entendus depuis le début de l'écoute du morceau : on peut cependant poser comme contexte d'un objet de niveau n l'ensemble des objets qui le précèdent depuis la dernière segmentation à ce même niveau.
- Les phénomènes d'attention sont également prépondérants (voir partie suivante). Ces phénomènes sont pris en compte dans l'architecture de l'algorithme et plus spécifiquement sur la structure de données de l'oracle multi-échelle.
- Plus un objet présente des similarités avec un ou plusieurs objets dans la section musicale à laquelle il appartient, plus cet objet sera considéré comme structurant.
- Plus un objet a déjà été segmenté, plus cet objet sera considéré comme structurant.

- On se place ici dans une conception linéaire de la musique en tant que séquence d'objets musicaux.

Ainsi, nous souhaitons caractériser la qualité d'une structure hiérarchique selon un ensemble de règles choisies en calculant une approximation de la quantité d'information nécessaire pour décrire ou construire les objets constituant la forme obtenue en fonction de l'ensemble des règles choisies. Un objet de forte complexité relativement à son contexte est ainsi un objet caractérisé comme étant fortement structurant.

Pour cela, on choisit donc de se baser sur la complexité de Kolmogorov [80].

4.2.1 Définition de la complexité

On propose de définir dans cette section la complexité d'une représentation de niveau n (définition 1), la complexité du diagramme formel multi-échelle (définition 2) et la complexité du diagramme de niveau j par rapport au diagramme de niveau $j - 1$ (définition 3).

Notations Nous utilisons pour cette section les notations suivantes, dont certaines sont introduites dans 2.2.2 et qui seront définies plus précisément au cours de la section :

- $r^{(n)} = x_1^{(n)} \dots x_t^{(n)} \dots x_l^{(n)}$ est la représentation de niveau n et de taille l .
- $r^{(j)}(t)$ est la représentation de niveau j à l'instant t telle que $r^{(j)}(t) = x_1^{(n)} \dots x_t^{(n)}$.
- $s^{(j)}(t)$ est la représentation segmentée de niveau j à l'instant t .
- $X_j(t) = (r^{(j)}(t), s^{(j)}(t), k^{(j)}(t))$ est le diagramme formel de niveau j à l'instant t avec $k^{(j)}(t)$ le marqueur de l'indice de la dernière segmentation au niveau j à l'instant t .
- $R(t) = X_0(t).X_1(t)\dots X_n(t)$ est le diagramme formel multi-échelle à l'instant t , où $n + 1$ est la taille du mot correspondant au nombre de niveaux.
- $C(R(t))$ est la complexité totale du diagramme formel multi-échelle à l'instant t , définie en fonction de trois coûts : $C_o(X_j(t))$ le *coût d'organisation interne*, $C_c(X_j(t))$ le *coût de classification* et $C_s(X_j(t))$ le *coût de segmentation*.
- $x_t^{(j)}$ est l'objet acquis à l'instant t et au niveau j .
- w^j est un facteur (au sens défini au chapitre 2) de niveau j .
- $H_s(X_j(t))$ est le *contexte* de $x_t^{(n)}$ dans le diagramme formel de niveau j à l'instant t . On le qualifiera également de *mémoire de travail* et il sera composé ici des éléments acquis depuis la dernière segmentation au niveau

- j .
- $M(j, t)$ est la mémoire matériau de niveau j à l'instant t .
 - $O(X_j(t))$ est le nombre d'états parcourus lors de l'ajout du dernier objet $x_t^{(j)}$ dans $X_j(t)$.

Definition 1 Soit $r^{(n)} = x_1^{(n)} \dots x_t^{(n)} \dots x_l^{(n)}$ une représentation de niveau n . On définit la *complexité d'une représentation de niveau n* selon la définition de Kolmogorov comme la somme des complexités des motifs successifs connaissant leur contexte.

$$C(r^{(n)}) = C(x_1^{(n)} \dots x_t^{(n)} \dots x_l^{(n)})$$

$$C(r^{(n)}) = C(x_1^{(n)}) + C(x_2^{(n)} | x_1^{(n)}) + C(x_3^{(n)} | x_1^{(n)} x_2^{(n)}) + \dots + C(x_t^{(n)} | x_{t-1}^{(n)} \dots x_1^{(n)}) + \dots + C(x_l^{(n)} | x_{l-1}^{(n)} \dots x_1^{(n)})$$

Avec l la longueur de la représentation. Si on limite le calcul de la complexité à un contexte donné qui est celui depuis la dernière segmentation que l'on définit comme la *mémoire de travail*, on obtient :

$$C(r^{(n)}) \approx \sum_{t=1}^l C(x_t^{(n)} | H_s(X_n(t))) \quad (4.1)$$

Où $H_s(X_n(t))$ est défini comme le *contexte* de $x_t^{(n)}$ qui correspond au préfixe de $x_t^{(n)}$ obtenu depuis la dernière segmentation lors de sa lecture dans $r^{(n)}$, obtenu grâce à $X_n(t)$ le diagramme formel de niveau n à l'instant t présenté dans la définition 2.

Definition 2 On suppose que chaque niveau dépend du niveau directement inférieur, et on note comme au chapitre 2 $R = X_0.X_1\dots X_n$ avec $n + 1$ la taille du mot correspondant au nombre de niveaux et $X_j = (r^{(j)}, s^{(j)}, k^{(j)})$ où $r^{(j)}$ est la représentation de niveau j , $s^{(j)}$ est la représentation segmentée de niveau j et $k^{(j)}$ le marqueur de l'indice de la dernière segmentation au niveau j . X_j est une représentation dynamique qui évolue au court du temps et de l'acquisition des objets de niveau j , on note donc $X_j(t) = (r^{(j)}(t), s^{(j)}(t), k^{(j)}(t))$ le diagramme de niveau j à l'instant t . La *complexité totale du diagramme formel hiérarchisé* à l'instant t est définie par :

$$C(R(t)) = C(X_0(t).X_1(t)\dots X_n(t)) \approx C(X_0(t)) + C(X_1(t) | X_0(t)) + \dots + C(X_n(t) | X_{n-1}(t)) \quad (4.2)$$

Dans le modèle que nous proposons, le niveau j dépend directement du niveau $j - 1$ et non pas des niveaux précédents car chaque concaténation et segmentation d'un ensemble d'objets de niveau $j - 1$ induit nécessairement la création d'un objet

de niveau j . Cela peut être une limite dans le cas de certaines œuvres musicales composées avec des ambiguïtés sur les échelles temporelles des niveaux.

Definition 3 La complexité du diagramme de niveau j à l'instant t , noté $X_j(t)$ par rapport au diagramme de niveau $j - 1$ à l'instant t , noté $X_{j-1}(t)$ est définie en fonction de trois coûts : $C_o(X_j(t))$ le coût d'organisation interne ; $C_c(X_j(t))$ le coût de classification et $C_s(X_j(t))$ le coût de segmentation de la façon suivante :

$$C(X_j|X_{j-1}) \approx \sum_{t=1}^{l_j} (C_o(X_j(t)) + C_s(X_j(t)) + C_c(X_j(t))) \quad (4.3)$$

Où l_j est le nombre d'objets dans le diagramme de niveau j .

Les dépendances de chacun de ces trois coûts sont explicitées dans les sections suivantes.

Coût d'organisation

Il exprime une complexité plus faible si les éléments qui constituent un même mot sont proches entre eux. On le définit donc par le coût de l'objet actuel $x_t^{(j)}$ connaissant le contexte $H_s(X_j(t))$, soit $C(x_t^{(j)}|H_s(X_j(t)))$.

$$C_o(X_j(t)) = C(x_t^{(j)}|H_s(X_j(t))) \quad (4.4)$$

Il est défini par la somme des coûts de $x_t^{(j)}$ par rapport aux objets $x_k^{(j)}$ contenus dans $H_s(X_j(t))$.

$$C(x_t^{(j)}|H_s(X_j(t))) = \frac{1}{|H_s(X_j(t))|} \sum_{x_k^{(j)} \in H_s(X_j(t))} C(x_t^{(j)}|x_k^{(j)}) \quad (4.5)$$

Coût de segmentation

Ce coût répond à la contrainte que plus un objet présente des similarités avec un ou plusieurs objets dans la section musicale à laquelle il appartient, plus cet objet sera considéré comme structurant. Il consiste donc à minimiser la complexité si les objets du contexte $H_s(X_j(t))$ sont fortement structurants, c'est-à-dire s'ils

sont proches d'un matériau existant dans la mémoire matériau $M(j, t)$ de niveau j dans l'état à l'instant t . Il s'agit donc de la complexité du contexte $H_s(X_j(t))$ par rapport à la mémoire matériau $M(j, t)$ soit $C(H_s(X_j(t))|M(j, t))$.

On a donc

$$C_s(X_j(t)) = C(H_s(X_j(t))|M(j, t)) \quad (4.6)$$

Cette complexité est mesurée selon les deux critères suivants :

- critère de reconnaissance du mot au niveau supérieur : correspond à la *règle 2* du chapitre 3 ;
- critère de retour à un objet de niveau actuel défini par la *règle 1* du chapitre 3.

Critère de reconnaissance du mot au niveau supérieur

Il s'agit de comparer le contexte de l'objet acquis $H_s(X_j(t))$ avec les facteurs $w^{(j)}$ déjà entendus au niveau actuel afin de déterminer si l'objet en voie de constitution est similaire à un objet de niveau supérieur déjà perçu. Le *coût de reconnaissance du mot au niveau supérieur* est défini par la somme, pour les facteurs $w^{(j)}$ contenus dans la mémoire matériau $M(j, t)$ des coûts du contexte $H_s(X_j(t))$ de $x_t^{(j)}$ par rapport aux facteurs $w^{(j)}$. Ce coût sera noté par $c(H_s(X_j(t))|w^{(j)})$.

Critère de retour à un objet de niveau actuel

Il s'agit de comparer le premier objet du contexte de l'objet acquis $H_s(X_j(t))[1]$ avec le premier objet des facteurs déjà entendus $w^{(j)}[1]$ au niveau actuel pour déterminer si il y a potentielle segmentation par le retour d'un objet de niveau actuel. Le *coût de retour à un objet de niveau actuel* est donc défini par la somme sur les facteurs $w^{(j)}$ contenus dans la mémoire matériau $M(j, t)$, du coût du premier objet du contexte de $x_t^{(j)}$, soit $H_s(X_j(t))[1]$ par rapport au premier objet du facteur, soit $w^{(j)}[1]$. On le note $c(H_s(X_j(t))[1]|w^{(j)}[1])$.

Ainsi, le calcul du coût de segmentation donne

$$C(H_s(X_j(t))|M(j, t)) = \sum_{w^{(j)} \in M(j, t)} c(H_s(X_j(t))|w^{(j)}) + \sum_{w^{(j)} \in M(j, t)} c(H_s(X_j(t))[1]|w^{(j)}[1]) \quad (4.7)$$

Coût de classification

Le coût de classification vise à minimiser la complexité liée à la description de l'objet $x_t^{(j)}$ en fonction du facteur de niveau $j - 1$ qui le constitue $w_t^{(j-1)}$. Il est donc décrit par le coût $c(x_t^{(j)}|w_t^{(j-1)})$.

$$C_c(X_j(t)) = C(x_t^{(j)}|w_t^{(j-1)}) \quad (4.8)$$

Complexité totale

La complexité totale du diagramme formel de niveau j en fonction du diagramme formel de niveau $j - 1$ étant obtenue selon l'équation (4.3) par la somme des coûts d'organisation (équation (4.4)), coût de segmentation (équation (4.6)) et coût de classification (équation (4.8)), on obtient :

$$C(X_j|X_{j-1}) \approx \sum_{t=1}^{l_j} [C(x_t^{(j)}|H_s(X_j(t)) + C(H_s(X_j(t))|M(j,t)) + C(x_t^{(j)}|w_t^{(j-1)})] \quad (4.9)$$

Afin de pouvoir comparer les complexités entre différents diagrammes formels multi-échelles, on divise la complexité totale par le nombre d'états inclus dans le diagramme formel multi-échelle. En effet, le diagramme formel de niveau 0 par exemple, s'il n'y a aucune segmentation, présente un coût de segmentation systématiquement nul. La segmentation aura donc tendance à faire augmenter la valeur de la complexité. La question est de savoir si la segmentation apporte un intérêt par rapport à une absence totale de segmentation, et si oui, quelle segmentation est plus pertinente qu'une autre, on cherche donc à calculer une complexité indépendamment du nombre d'objets créés en mémoire. Le coût d'organisation est également indépendant du nombre d'objets auxquels on compare l'objet actuel. En effet, on cherche à moyenniser la similarité de l'objet en cours d'acquisition avec l'ensemble des objets précédemment acquis, indépendamment de la longueur de l'objet en voie de constitution (on choisit qu'un objet plus long n'ait pas un coût d'organisation interne plus élevé). En revanche, on tiendra compte de la taille de la mémoire et de l'historique pour les coûts de classification et de segmentation : en effet, un historique plus grand nécessitera éventuellement d'effectuer un plus grand nombre de comparaisons pour classer l'objet ou pour déterminer si l'objet à segmenter est similaire à un objet de niveau supérieur. Si l'organisation de la mémoire est prise en compte pour le calcul du coût de classification avec l'oracle, ce n'est pas encore le cas dans l'algorithme pour la segmentation, ou l'objet est comparé avec tous ceux de l'historique.

On définit ainsi la complexité du diagramme formel hiérarchisé en fonction du nombre d'états ne de R par $C(R, ne)$ par :

$$C(R, ne) = \frac{C(R)}{ne} \quad (4.10)$$

Pour chaque coût, nous proposons de les exprimer de deux manières distinctes : en fonction d'une valeur de similarité, ou encore en fonction du nombre d'opérations nécessaires pour obtenir une telle information.

4.2.2 Complexité par calcul des opérations

Nous détaillons plus précisément la valeur de ces coûts. Une première approche est de compter le nombre d'opérations nécessaires pour décrire un objet selon son contexte. Pour cela, on s'intéresse à la notion d'alignement, et les opérations d'occurrence, substitution, suppression et insertion permettant de décrire une séquence à partir d'une autre.

Definition 4 Soit (u, v) un couple de mots sur l'alphabet $\mathcal{A}^{(j)}$. On définit un *alignement* $\phi(u, v)$ de u et de v par l'opérateur ϕ permettant d'obtenir le quadruplet de mots (u', v', f, g) tel que $u' = u \sqcup _{}^k = x_1 \dots x_{|u'|}$ et $v' = v \sqcup _{}^k = y_1 \dots y_{|v'|}$ avec \sqcup l'opérateur shuffle permettant d'introduire de manière ordonnée à des indices aléatoires k éléments $_{}^k$. Le caractère $_{}^k$ représente ce qui est appelé un décalage ou *gap*. f et g sont deux mots définis par $f = f_1 \dots f_{|u'|}$ et $g = g_1 \dots g_{|v'|}$ tels que $\forall i, f_i \in Op$ et $g_i \in Op$ où Op est l'ensemble des opérations d'*occurrence*, d'*insertion*, de *suppression* et de *substitution* que l'on définit ci-dessous. Chaque opération d'indice i de f (respectivement g) est appliquée à chaque lettre indicée i de u' (respectivement v') de telle sorte à obtenir un mot commun z avec $f(u') = f_1(x_1) \dots f_{|u'|}(x_{|u'|}) = z$ et $g(v') = g_1(y_1) \dots g_{|v'|}(y_{|v'|}) = z$.

On a ainsi

$$\begin{aligned} u &\rightarrow u' \xrightarrow{f} z \\ v &\rightarrow v' \xrightarrow{g} z \end{aligned}$$

Definition 5 On définit Op l'ensemble des quatre opérations permettant de définir un alignement :

- l'occurrence $occ(x) = x$;
- la suppression $del(x) = \emptyset$;
- l'insertion $ins(_, a) = a$ et $ins(x, a) = \emptyset$ pour $x \neq _{}^k$;
- la substitution $sub(x, a) = a$.

Où x et a sont des lettres de $\mathcal{A}^{(j-1)}$ et $_{}^k$ est le caractère représentant un décalage.

Definition 6 Le coût d'un alignement $\phi(u, v) = (u', v', f, g)$ avec $f = f_1 \dots f_{|u'|}$ et $g = g_1 \dots g_{|v'|}$, $C(\phi(u, v))$ est défini comme la somme des coûts associés aux opérateurs f_i et g_i normalisés respectivement par la taille de u' et la taille de v' .

$$C(\phi(u, v)) = \frac{1}{2} \left(\frac{1}{|u'|} \sum_{i=1}^{|u'|} C(f_i(x_i)) + \frac{1}{|v'|} \sum_{i=1}^{|v'|} C(g_i(y_i)) \right) \quad (4.11)$$

On calcule ainsi un alignement $\phi_{min}(u, v)$ nous permettant d'obtenir un coût minimal des alignements possibles entre u et v . On considérera la complexité en fonction de ce coût minimal.

On cherche alors à formaliser les coûts de ces quatre opérations. Nous constatons que le calcul du coût de $\phi(u, v)$ dépend de la définition récursive du mot u (respectivement v) qui constitue le matériau de niveau supérieur. Ainsi, au niveau 0, on a u et v des mots de taille 1 tel que $\forall(u, v)$, si $u \neq v$ alors $C(\phi(u, v)) = 1$ et $C(\phi(u, v)) = 0$ sinon.

Occurrence

Une occurrence $occ(x_i)$ ne donne lieu à aucune modification de la séquence u' . Cette opération a donc toujours un coût nul. Donc

$$C(occ(x_i)) = 0$$

On a bien au niveau 0 pour $u = v$, $\phi(u, v) = (u, v, occ(u), occ(v))$ donc

$$C(\phi(u, v)) = \frac{1}{2} \left(\frac{1}{|u|} C(occ(u)) + \frac{1}{|v|} C(occ(v)) \right) = 0$$

Insertion

On souhaite qu'une insertion n'ait pas nécessairement le même coût en fonction du l'historique de l'élément a à insérer à la place de $x_i = _$ dans u' . Par exemple, si l'élément à insérer est identique à celui qui le précède, le coût sera moindre par rapport à un élément tout à fait distinct. On considère ici un historique de taille 1, c'est à-dire que l'on considère uniquement l'élément x_{i-1} qui précède directement x_i . Le coût d'insertion correspond donc au nombre d'opérations nécessaires pour passer de l'élément précédent x_{i-1} à a ajouté au coût constant de l'opération d'écriture d'un caractère.

Si $i = 0$, il s'agit du coût d'écriture d'un élément $c_{ecr}(x) = 1$, soit

$$C(ins(_, a)) = C_{ecr}(a) = 1$$

Si $i > 0$:

$$C(\text{ins}(_, a)) = \frac{(c(\phi(x_{i-1}, a)) + C_{\text{ecr}}(a))}{2}$$

Au niveau 0, on a donc :

si $x_{i-1} = a$:

$$C(\text{ins}(_, a)) = \frac{(C(\phi(x_{i-1}, a)) + C_{\text{ecr}}(a))}{2} = \frac{(0 + 1)}{2} = \frac{1}{2}$$

sinon :

$$C(\text{ins}(_, a)) = \frac{(C(\phi(x_{i-1}, a)) + C_{\text{ecr}})}{2} = \frac{(1 + 1)}{2} = 1$$

Si l'élément précédant l'élément x_i , x_{i-1} n'existe pas on a :

$$C(\text{ins}(_, a)) = (C_{\text{ecr}} + C_{\text{ecr}})/2 = 1$$

Suppression

On souhaite également qu'une suppression n'ait pas nécessairement le même coût en fonction de l'historique de l'élément x_i à supprimer dans u' . Par exemple, si l'élément à supprimer est identique à celui qui le précède, le coût sera moindre par rapport à un élément tout à fait distinct. On considère ici également un historique de taille 1. Le coût d'insertion correspond donc au nombre d'opérations nécessaires pour passer de l'élément précédent x_{i-1} à x_i ajouté au coût constant de l'opération d'effacement d'un caractère.

Si $i = 0$, il s'agit du coût d'effacement d'un élément $C_{\text{eff}}(x_i) = 1$,

$$C(\text{del}(x_i)) = 1$$

Si $i > 0$:

$$C(\text{del}(x_i)) = (C(\phi(x_{i-1}, x_i)) + C_{\text{eff}})/2$$

Au niveau 0, on a donc :

si $x_{i-1} = x_i$:

$$C(\text{del}(x_i)) = \frac{(C(\phi(x_{i-1}, x_i)) + C_{\text{eff}})}{2} = \frac{(0 + 1)}{2} = \frac{1}{2}$$

sinon :

$$C(\text{del}(x_i)) = \frac{(C(\phi(x_{i-1}, x_i)) + C_{eff})}{2} = \frac{(1 + 1)}{2} = 1$$

Si l'élément précédant l'élément x_i , x_{i-1} n'existe pas on a :

$$C(\text{del}(x_i)) = (C_{eff} + C_{eff})/2 = 1$$

Substitution

Une substitution $sub(x_i, a)$ dépend du coût de l'alignement du facteur $f_w(x_i)$ de x_i de niveau inférieur et du facteur $f_w(a)$ de a de niveau inférieur stocké dans la mémoire matériau, avec $f_w(x)$ la fonction permettant d'accéder au facteur de x . Ce coût est ainsi obtenu récursivement à partir des coûts calculés au niveau 0.

On a

$$C(sub(x_i, a)) = C(\phi(f_w(x_i), f_w(a)))$$

avec au niveau 0, $C(\phi(f_w(x_i), f_w(a))) = C(\phi(x_i, a)) = 1$ si $x_i \neq a$ et 0 sinon.

Nous présentons maintenant l'évaluation des trois coûts d'organisation, de segmentation et de classification en fonction des quatre opérations d'occurrence, d'insertion, de suppression et de substitution décrites ici.

Coût d'organisation

Si on calcule les coûts en fonction du nombre d'opérations effectuées, il s'agit de compter le nombre d'opérations minimal permettant d'écrire $x_t^{(j)}$ à partir des $x_k^{(j)} \in H_s(x_t^{(j)})$. Cela revient à trouver l'alignement minimal entre les deux objets $w_k^{(j-1)}$ et $w_t^{(j-1)}$ si $j > 0$ et $x_t^{(j)}$ et $x_k^{(j)}$ sinon.

On a donc :

$$C_o(X_j(t)) = C(x_t^{(j)} | H_s(X_j(t))) = \sum_{x_k^{(j)} \in H_s(X_j(t))} C(\phi(x_t^{(j)}, x_k^{(j)})) \quad (4.12)$$

Coût de segmentation

Le coût de segmentation est détaillé ici en fonction des deux critères définis précédemment.

Critère de reconnaissance du mot au niveau supérieur

Si on calcule le coût en fonction du nombre d'opérations effectuées, il s'agit de la somme sur l'ensemble des éléments de la mémoire matériau de niveau j , $M(j, t)$, du nombre d'opérations effectuées pour transformer le mot de la mémoire matériau en $x_t^{(j)}$.

On a donc :

$$C(H_s(X_j(t))|w^{(j)}) = C(\phi(H_s(X_j(t)), w^{(j)})) \quad (4.13)$$

Critère de retour à un objet de niveau actuel

En fonction du nombre d'opérations, on calcule la somme sur l'ensemble des éléments de la mémoire matériau de niveau j , $M(j, t)$, du nombre d'opérations effectuées pour transformer le premier caractère du mot de la mémoire matériau en $x_t^{(j)}[1]$.

$$C(H_s(X_j(t))[1]|w^{(j)}[1]) = C(\phi(H_s(X_j(t))[1], w^{(j)}[1])) \quad (4.14)$$

Coût de classification

Si on compte le nombre d'opérations effectuées pour le coût de classification, celui-ci se divise en trois opérations :

- le coût de parcours de l'oracle $C_{oracle}(x_t^{(j)})$;
- le coût de création de l'objet $C_{objet}(x_t^{(j)})$;
- le coût de mise à jour de l'historique $C_{hist}(x_t^{(j)})$.

On propose de définir le coût de parcours de l'oracle de la manière suivante :

$$C_{oracle}(x_t^{(j)}) \approx |M(j, t)| + \delta + \beta(x_t^{(j)}|P(x_t^{(j)})) \quad (4.15)$$

avec δ le nombre de liens suffixes créés dans l'oracle pour l'ajout de l'état correspondant à $x_t^{(j)}$ qui est égal à 1 quelle que soit la valeur du nouvel état car pour tout nouvel état, un unique lien suffixe est créé. Par ailleurs $\beta(x_t^{(j)}|P(x_t^{(j)}))$ est le nombre de liens extérieurs créés dans l'oracle pour l'ajout de ce même état. Il dépend de $x_t^{(j)}$ et de $P(x_t^{(j)})$ le préfixe de $x_t^{(j)}$ que l'on définit comme le plus long mot déjà vu précédant $x_t^{(j)}$. On présente en annexe C le nombre de liens créés en fonction de la qualité du nouvel objet ajouté dans l'oracle et de son préfixe.

On a par ailleurs :

$$C_{hist}(x_t^{(j)}) < \alpha + 2|M(j, t)|$$

$$C_{objet}(x_t^{(j)}) \approx \gamma$$

Où α est une constante associée au calcul du label et à l'ajout dans le tableau des historiques et γ est une constante d'ajout dans les différentes structures de données qui constituent l'objet. La longueur de la mémoire est ajoutée deux fois lors de la création d'un nouveau matériau pour le parcours de la matrice d'auto-similarité des matériaux lors de sa mise à jour : la valeur de similarité avec chaque matériau est ajoutée aux lignes correspondantes au matériau comparé et au nouveau matériau.

4.2.3 Complexité par calcul de similarité

Une deuxième approche est de décrire la complexité en fonction de la similarité entre deux objets étudiés. En effet, au lieu de compter le nombre d'opérations nécessaires pour décrire un objet x_t en fonction de son contexte, on utilise l'alignement optimal qui est calculé lors du calcul de similarité entre deux objets. La valeur de similarité correspond aux opérations successives d'occurrence (une occurrence donne une valeur de 1), d'insertion et de suppression (toutes deux représentées par une pénalité correspondant à un *décalage*) et une substitution, dont la valeur entre deux labels correspond à la valeur de similarité calculée au niveau inférieur et obtenue récursivement depuis le niveau 0. Ainsi, nous proposons une version simplifiée des calculs précédents prenant en compte les valeurs de similarité calculées. On propose alors d'étendre ce calcul à toutes les similarités proposées dans l'Agenda.

Coût d'organisation

Si on calcule les coûts en fonction du calcul de similarité, alors $C(x_t^{(j)}|x_k^{(j)})$ correspond à un moins la similarité de l'objet actuel $x_t^{(j)}$ avec $x_k^{(j)} \in H_s(X_j(t))$. On obtient ainsi :

$$C(x_t^{(j)}|H_s(X_j(t))) = \sum_{x_k^{(j)} \in H_s(X_j(t))} (1 - sim(x_t^{(j)}, x_k^{(j)})) \quad (4.16)$$

Avec $sim(x_t^{(j)}, x_k^{(j)})$ la fonction qui renvoie la valeur de similarité entre $x_t^{(j)}$ et $x_k^{(j)}$ selon les opérations de similarité présentées au chapitre 3, et précisément

l'opération choisie par l'utilisateur dans l'agenda. Le résultat est compris entre 0 et 1. Cette fonction est donc une généralisation des fonctions de similarité choisies dans l'Agenda et utilisées pour la classification, dont l'alignement.

Coût de segmentation

Le coût de segmentation est à nouveau défini par les deux critères de reconnaissance du mot de niveau supérieur et de retour à un objet de niveau actuel.

Critère de reconnaissance du mot au niveau supérieur

En fonction de la similarité, on souhaite minimiser le coût s'il existe un mot similaire dans la mémoire matériau $M(j, t)$. Le coût est défini en fonction de la similarité entre $H_s(X_j(t))$ le *contexte* de $x_t^{(n)}$ et $w^{(j)}$ le facteur de niveau j dans $M(j, t)$.

$$c(H_s(X_j(t))|w^{(j)}) = (1 - sim(H_s(X_j(t)), w^{(j)})) \quad (4.17)$$

Critère de retour à un objet de niveau actuel

Le calcul du coût de retour à un objet de niveau actuel en fonction de la similarité est le même que pour le critère de retour d'un mot de niveau supérieur, mais uniquement sur le premier caractère de chaque mot :

$$c(H_s(X_j(t))[1]|w^{(j)}[1]) = (1 - sim(H_s(X_j(t))[1], w^{(j)}[1])) \quad (4.18)$$

Coût de classification

Le coût de la classification est un moins la similarité de $w_t^{(j-1)}$ avec tous les facteurs associés aux labels auxquels $w_t^{(j-1)}$ est comparé dans le parcours de l'oracle. Plus un matériau similaire est trouvé rapidement, moins le coût sera élevé. Cela donne :

$$C(x_t^{(j)}|w_t^{(j-1)}) = \sum_{w^{(j-1)} \in O(X_j(t))} (1 - sim(w_t^{(j-1)}, w^{(j-1)})) \quad (4.19)$$

où $O(X_j(t))$ est l'ensemble des états parcourus dans l'oracle à l'ajout de $x_t^{(j)}$

On note que le temps d'accès à un élément similaire n'est pas pris en compte dans le calcul de la similarité, alors qu'il est optimisé avec l'implémentation de

l'Oracle de Markov. Une piste d'amélioration est de ne compter pour le coût de segmentation, uniquement le coût des comparaisons effectivement faites, et non pas nécessairement la comparaison avec l'intégralité des objets de la mémoire matériau.

4.2.4 Analyses

L'utilité du calcul d'une telle complexité est multiple : d'une part la calculer dynamiquement, au fur-et-à-mesure de l'analyse, pourrait être un outil de segmentation. En effet, on chercherait à choisir la segmentation permettant de minimiser au fur et à mesure le coût calculé à l'ajout de chacun des nouveaux états dans l'algorithme. Ici, nous déterminons un protocole afin d'étudier si MORFOS permet de présenter une structure plus organisée que d'autres segmentations. Si MORFOS permet une segmentation plus pertinente qu'une autre, on s'attend à avoir une complexité cumulée plus faible que pour d'autres segmentations. On propose donc de comparer les complexités $C(R, ne)$ par calcul de similarité sur le *Rondo K545* de W.A. Mozart selon différentes segmentations :

- aucune segmentation ;
- tous les objets sont segmentés ;
- les facteurs de taille deux sont segmentés
- une segmentation aléatoire ;
- un ensemble A de règles de segmentation pertinentes ;
- un ensemble B de règles de segmentation pertinentes.

Les mesures de similarités elles, sont gardées identiques sur tous les exemples.

Aucune segmentation

Le premier cas étudié est le cas où il n'y a aucune segmentation. On présente en figure 4.3 le diagramme formel multi-échelle obtenu. Il est constitué ici d'un unique diagramme au niveau 0, contenant les 41 états fournis en entrée de l'algorithme.

La complexité calculée lors de l'acquisition de chaque objet en fonction du temps est présentée en figure 4.4.

Chaque courbe correspond à la complexité obtenue pour un niveau de segmentation donnée : ici, il n'y a qu'un niveau de segmentation. On constate une complexité croissante pour chaque acquisition d'un matériau jamais vu précédemment. A l'inverse, l'acquisition d'un matériau correspondant à une hypothèse directe par rapport à l'élément précédemment entendu induit une complexité très faible. Cela est induit par le parcours dans l'oracle pour l'ajout du matériau, qui impacte le

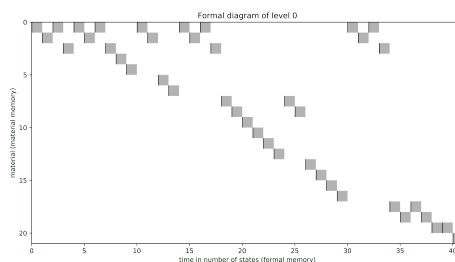


FIGURE 4.3 – Diagramme Formel Multi-échelle du *Rondo K545* de W.A. Mozart dans le cas sans segmentation.

coût de classification. On propose d’analyser les coûts en fonction des hypothèses dans la section suivante.

Segmentation tous les 2 objets

On présente en figure 4.5 le diagramme formel multi-échelle obtenu. Tous les facteurs de taille 2 sont segmentés, à l’exception du dernier facteur qui est de taille 3 lorsque le nombre d’objet dans le niveau est impair. La complexité calculée lors de l’acquisition de chaque objet est également présentée en figure 4.5.

Cinq courbes sont obtenues pour les cinq niveaux calculés. Si nous nous concentrons sur la courbe du niveau 0, nous constatons une tendance générale similaire à la courbe du diagramme formel de niveau 0 sans segmentation. Cependant, nous observons en plus de cela une tendance croissante, liée au coût de segmentation supplémentaire, qui était nul dans le premier exemple, tandis que le nombre de matériaux dans l’historique est croissant au cours de l’acquisition de l’œuvre ici. A partir du niveau 2, nous observons l’acquisition de nouveaux matériaux au fur-et-à-mesure, d’où l’obtention de courbes de complexité strictement croissantes mais aux coûts d’autant moins élevés qu’il n’y a d’états et de matériaux dans la représentation.

Segmentation aléatoire

On présente en figure 4.6 deux exemples de segmentation aléatoire et les complexités calculées lors de l’acquisition de chaque objet correspondantes.

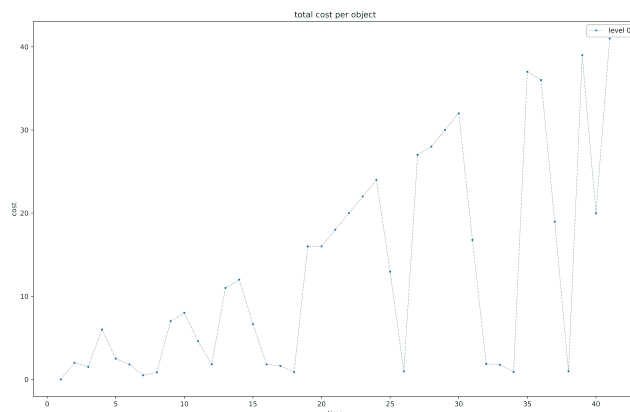


FIGURE 4.4 – Complexité lors de l’acquisition de chaque objet dans le cas sans segmentation.

Les mêmes commentaires s’appliquent à ces diagrammes qu’à ceux de la segmentation par facteurs de taille 2. Seulement, nous constatons que la segmentation aléatoire induit la création de nombreux nouveaux matériaux qui accentuent le coût de segmentation par l’augmentation non optimale de la taille de l’historique, mais également le coût de classification par le parcours important de l’oracle car les facteurs sont très peu redondants.

Ensemble A de règles de segmentation de l’agenda

On présente en figure 4.7 le diagramme formel multi-échelle obtenu en fonction d’un ensemble de règles A et la complexité correspondante.

Les règles sélectionnées sont les règles de segmentation 1,2 et 5 présentées au chapitre 3.

Comme observé pour le niveau 0 dans l’exemple sans segmentation, la redondance des matériaux aux niveaux supérieurs permet de diminuer les coûts lors de l’acquisition des nouveaux objets. Néanmoins, on constate que les opérations de segmentation au niveau 0 ainsi que l’ajout des coûts des niveaux supérieurs induira nécessairement une complexité totale supérieure à l’exemple sans segmentation.

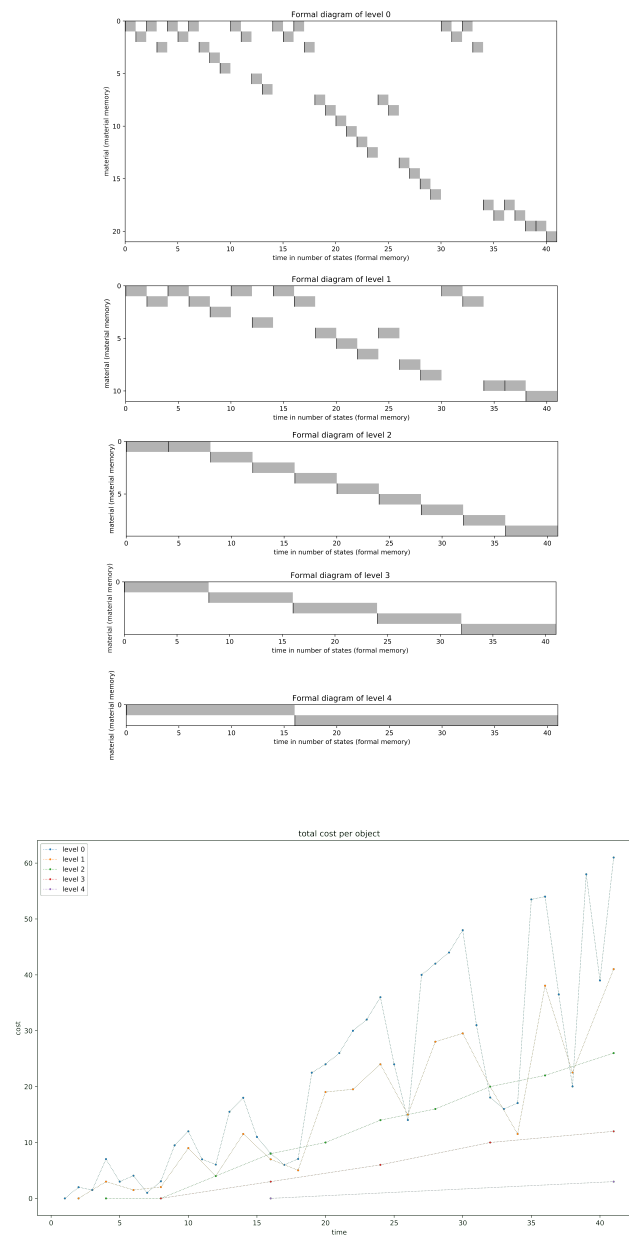


FIGURE 4.5 – Diagramme Formel Multi-échelle et complexité du *Rondo K.545* de W.A. Mozart dans le cas où les facteurs de taille 2 sont segmentés.

Ensemble B de règles de segmentation de l'agenda

On rappelle en figure 4.8 le diagramme formel multi-échelle obtenu en fonction d'un ensemble de règle B, qui correspond aux règles choisies pour l'exemple

4. Un Modèle d'Écoute : coûts et cognition

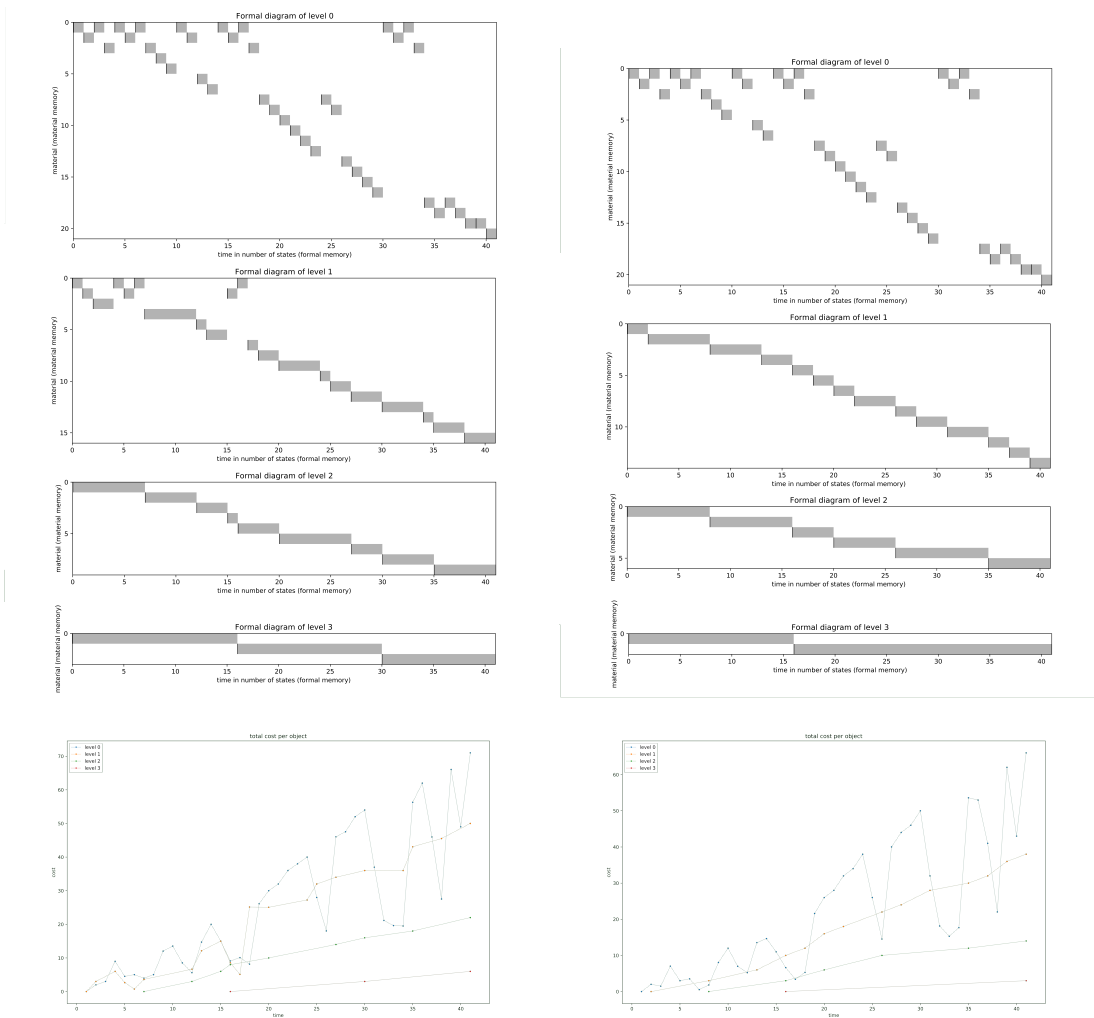


FIGURE 4.6 – Exemples de Diagramme Formel Multi-échelle du *Rondo K.545* de W.A. Mozart dans le cas de deux segmentations aléatoires (haut, 1 à gauche, 2 à droite) et complexités correspondantes (bas).

présenté dès le chapitre 2 et que nous retrouverons au chapitre 5. Les règles sélectionnées sont les règles de segmentation 1, 2, 3 et 5 présentées au chapitre 3.

La complexité calculée lors de l'acquisition de chaque objet est présentée en figure 4.9.

Le comportement obtenu est très proche de l'exemple avec les règles de segmentation A. Néanmoins, on constate une baisse du coût lors de la deuxième acquisition de l'objet e au niveau 2 (instant 26) contrairement à l'exemple précédent.

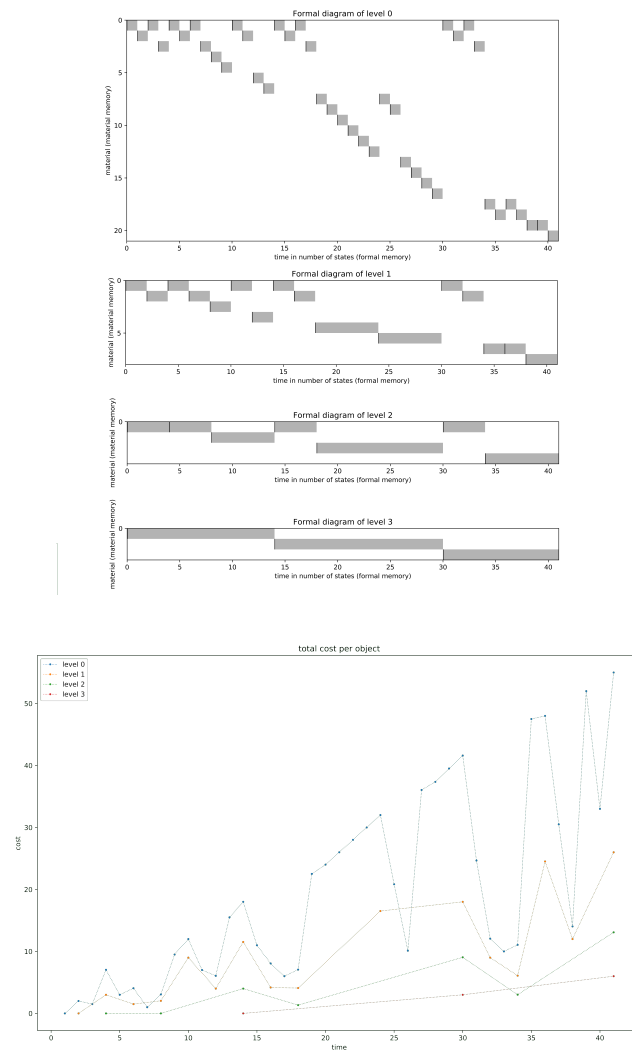


FIGURE 4.7 – Diagramme Formel Multi-échelle et complexité du *Rondo K.545* de W.A. Mozart avec l'ensemble A des règles sélectionnées dans l'Agenda.

Cependant, cela implique l'acquisition de deux objets supplémentaires qui ont eux-mêmes un coût d'acquisition, qui comme on le constatera lors de la comparaison des coûts totaux ne suffit pas à obtenir une complexité plus faible.

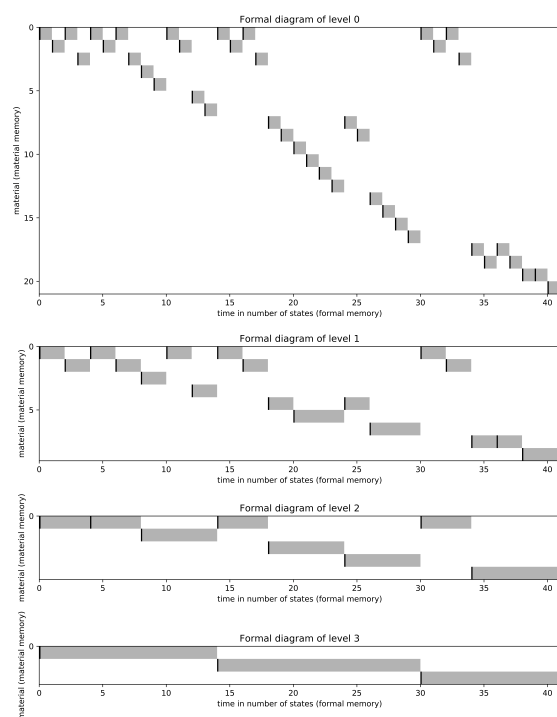


FIGURE 4.8 – Diagramme Formel Multi-échelle du *Rondo K.545* de W.A. Mozart avec l'ensemble B des règles sélectionnées dans l'Agenda.

Complexités

On récapitule dans le tableau ci-dessous les complexités totales obtenues pour chacun des exemples. Pour la segmentation aléatoire, on a calculé une complexité moyenne sur 1000 itérations.

segmentation	complexité totale	coût organi- sation	coût classifi- cation	coût seg- mentation
aucune	13.0	0.9	12.1	0
facteurs de taille 2	17.6	0.8	10.0	6.8
seg. aléatoire 1	21.6	0.9	11.7	9.0
seg. aléatoire 2	19.9	0.9	11.5	7.5
seg. aléatoire moy.	22.9	0.9	12.1	9.9
règles A	14.9	0.8	9.3	4.7
règles B	15.7	0.8	9.5	5.4

Nous constatons que pour tous les diagrammes formels multi-échelles segmen-

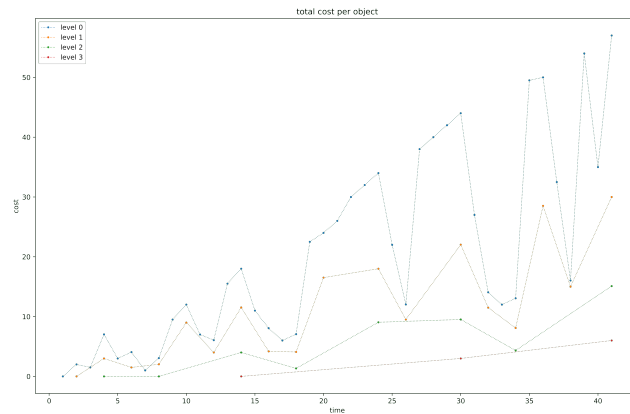


FIGURE 4.9 – Complexité lors de l’acquisition de chaque objet avec l’ensemble B des règles sélectionnées dans l’Agenda.

tés, la segmentation avec l’ensemble des règles A est celle avec la complexité la plus faible, suivie ensuite de la segmentation avec l’ensemble des règles B, puis la segmentation par facteurs de taille 2, la segmentation aléatoire et enfin la segmentation maximale.

Cependant lorsqu’il n’y a pas de segmentation, nous avons toujours une complexité minimale.

On retrouve cet ordre précis lorsque l’on regarde dans le détail le coût de classification et le coût de segmentation. Le coût d’organisation (compris entre 0 et 1) est relativement stable mais supérieur pour la segmentation aléatoire et l’exemple non segmenté.

Cela nous indique qu’une segmentation proposée par l’agenda permet effectivement une organisation interne nécessitant une moins grande complexité de calcul. Cependant, la sur-segmentation du facteur (*hi*) avec les règles B n’a pas d’impact sur les niveaux supérieurs et induit un coût supplémentaire. La segmentation par facteurs de taille 2 n’est pas tout à fait incohérente car la structure de l’œuvre suit un schéma binaire assez conventionnel, tandis qu’une segmentation aléatoire induit logiquement une entropie plus élevée.

4.3 Analyse de second ordre et phénomènes d'attention

On cherche dans cette section à analyser le comportement de l'algorithme afin de mieux comprendre les phénomènes d'attention lors de l'écoute musicale. Pour cela, on se base sur la notion de *phases cognitives* présentées par J.M. Chouvel dans [31].

4.3.1 Phases cognitives

Les phases cognitives correspondent à la caractérisation des comportements liés à la prise en charge du flux d'information dans l'Algorithme Cognitif, c'est-à-dire le flux audio lors de l'écoute musicale.

Ces phases cognitives sont liées aux résultats des deux tests principaux de l'Algorithme Cognitif tels qu'ils sont originalement présentés, c'est-à-dire le test paradigmatique (test de similarité : on compare l'objet actuellement acquis dans l'algorithme avec les objets précédemment connus de même niveau), et le test syntagmatique (test de segmentation : on regarde si l'objet acquis constitue un événement de clôture pour la création d'un objet de niveau supérieur).

Selon Chouvel [31], ces tests sont par ailleurs associés à des fonctionnalités cérébrales. Le test de reconnaissance paradigmatique est lié aux "*fonctions associatives et analogiques*", qui correspondent aux fonctions qui traitent notamment l'information des aires sensibles et sensorielles et donne lieu à des phénomènes cognitifs de reconnaissance. Tandis que le test de reconnaissance syntagmatique correspond aux "*fonctions décisionnelles et logiques*", c'est-à-dire qui émettent l'évaluation d'un résultat et d'une prise de décision, ici concernant la segmentation.

Ainsi, on compte quatre phases cognitives, associées à la combinatoire des résultats possibles des deux tests :

- Phase 1 : similarité invalide et segmentation invalide
- Phase 2 : similarité invalide et segmentation valide
- Phase 3 : similarité valide et segmentation invalide ;
- Phase 4 : similarité valide et segmentation valide.

Ces phases permettent à J.M. Chouvel d'expliciter différents états cognitifs :

Intégration : il y a intégration lorsqu'il y a sensation de continuité, en termes pratiques, lorsque l'objet actuel est concaténé aux objets entendus depuis la dernière

segmentation et qu'un objet de niveau supérieur est en conséquence en cours de construction.

Rétention : J.M. Chouvel écrit dans [32] que "*La mémoire rétentionnelle nécessite par la forme doit permettre la mémorisation des événements élémentaires*". Ainsi, la rétention correspond spécifiquement à la mémorisation de l'information la plus compressée possible, impliquant le moins de perte d'information. Il s'agit ici de mémoriser chaque nouveau matériau. Il y a donc rétention lorsque le test de similarité est invalidé.

Réalisation : J.M. Chouvel dans [32] définit un processus cognitif comme la représentation de la transformation de "*l'image perceptuelle, riviée à la temporalité, en un ensemble de données, pour rester vague, saisissables par le mental*". La réalisation est ainsi la représentation concrète de ces connexions. Dans l'algorithme, la réalisation est donc la représentation concrète d'un objet précédemment perçu dans la mémoire, et correspond à l'acquisition au niveau actuel d'un objet déjà entendu. Il y a ainsi réalisation quand le test de similarité est validé.

Acquisition/récurtivité : L'acquisition correspond au phénomène de groupement des objets dernièrement entendus pour constituer un objet de niveau supérieur, en vue de traiter ce nouvel objet en tant que tout. La récursivité est donc l'appel de l'Algorithme Cognitif au niveau supérieur sur ce nouvel objet acquis. Il y a donc acquisition au niveau supérieur et récursivité au niveau de l'algorithme lorsqu'il y a *achèvement*, c'est à dire que le test de segmentation est validé.

Attente de l'inconnu : Il y a attente de l'inconnu lorsqu'aucune hypothèse ne peut être émise pour l'*objet successeur*. On parle ici d'objet successeur pour désigner l'objet acquis au niveau actuel après l'objet en cours d'acquisition. Sinon, l'individu qui écoute est dans l'*attente du connu*. Cependant bien que le connu est attendu, l'objet effectivement acquis peut ne pas correspondre à un objet attendu. Cela donne ainsi lieu à des possibilités de *prédiction satisfaite* ou de *prédiction insatisfaite*. On a une prédiction satisfaite lorsque l'objet acquis successivement correspond à l'hypothèse émise, et une prédiction insatisfaite sinon.

On propose dans la suite d'expliquer comment ces phases se concrétisent dans MORFOS. Dans le cadre de notre modèle, les tests de similarité et de segmentation sont décalés par rapport au modèle de Chouvel. En effet, nous considérons la segmentation comme antérieure à la classification : avant de pouvoir classer un élément, celui-ci doit d'abord pouvoir être distingué. En tout cas au niveau algorithmique, nous procédons d'abord à l'échantillonnage automatisé du signal, ou bien à une segmentation préalable permettant de fournir une représentation symbolique séquentielle. En réalité, il est probable que ces deux tests aient lieu en

simultané : on constate dans l'implémentation de l'algorithme que la classification succède immédiatement la segmentation, mais que des calculs de similarité sont préalablement opérés pour déterminer la segmentation. Il s'agit d'une discussion qui mériterait de plus amples réflexions.

Ainsi, la modélisation des phases cognitives dans notre logiciel est présentée en figure 4.10. Nous introduisons les termes d'*objet en cours d'acquisition* x_t^j pour désigner l'objet acquis actuellement au niveau j et à l'instant t . Il est à ne pas confondre avec l'*objet en voie de constitution* qui correspond à la concaténation des objets d'un niveau donné en vue de construire un objet de niveau supérieur. On parle d'*objet successeur* pour l'objet qui succède immédiatement l'objet en cours d'acquisition à ce même niveau soit l'objet x_{t+1}^j acquis à l'instant $t + 1$ au niveau j . On pourra étendre cette terminologie pour parler d'*objet acquis de niveau supérieur* x_k^{j+1} qui correspond à l'objet de niveau $j + 1$ clôturé par x_t^j , et d'*objet de niveau supérieur successeur* x_{k+1}^{j+1} , qui est l'objet de niveau j initié par x_{t+1}^j .

Une *itération* de l'algorithme étant l'acquisition d'un objet à un instant t et à niveau j donné et l'application du test de classification et de segmentation sur cet objet, deux itérations de l'algorithme sont représentées en figure 4.10. La première (haut) correspond à l'itération *segmentation/classification* de niveau $j - 1$ permettant d'obtenir l'objet x_t^j tandis que la deuxième (bas) correspond à l'itération *segmentation/classification* effectuée sur l'acquisition de l'objet x_{t+1}^j . Les deux tests concernés pour la détermination de la phase cognitive sont les tests colorés en gris foncé. Il s'agit donc tout d'abord du test de classification permettant de labéliser l'objet en cours d'acquisition x_t^j à partir du facteur de niveau inférieur w_t^{j-1} : l'objet segmenté permettant l'obtention de ce facteur ne nous intéresse pas ici.

Une autre différence majeure de notre algorithme par rapport à celui proposé par J.M Chouvel est que le test de segmentation valide ou non une segmentation entre l'objet actuellement acquis et l'objet précédent. Il ne s'agit donc pas d'informer que l'objet en cours d'acquisition *clôture* l'objet de niveau supérieur acquis, mais que l'objet successeur *initie* l'objet de niveau supérieur successeur.

Le test de segmentation pertinent est donc celui appliqué à l'objet successeur x_{t+1}^j . En effet, lors de l'acquisition de x_{t+1}^j , le test de segmentation permettra de déterminer si x_t^j clôture bien l'objet de niveau supérieur x_k^{j+1} ou non.

Ainsi, le premier test correspond à la classification du facteur $w_t^{(j-1)}$ de niveau $j - 1$: il s'agit de la labélisation de ce facteur en un objet en cours d'acquisition $x_t^{(j)}$ au niveau j . Si cette labélisation correspond à celle d'un matériau déjà vu, alors le test de classification est dit *valide*. Autrement, le test est *invalide*.

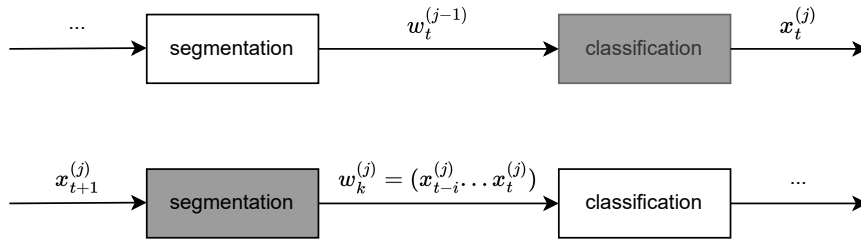


FIGURE 4.10 – Modélisation des phases cognitives.

Ensuite, le deuxième test consiste à déterminer si l'objet $x_t^{(j)}$ constitue ou non un élément de clôture. Cela signifie que lors de l'acquisition de l'objet qui le suit $x_{t+1}^{(j)}$, il y a segmentation en un facteur $w_k^{(j)} = (x_{t-i}^{(j)} \dots x_t^{(j)})$. S'il y a segmentation, ce test est dit *valide*. Autrement, lorsqu'il n'y a pas segmentation le test de segmentation est *invalide*.

On présente ainsi les quatre phases cognitives.

Phase cognitive 1

La phase cognitive 1 correspond à un test de similarité invalide et un test de segmentation invalide. Le test de similarité au niveau j est invalide ce qui signifie que l'objet en cours d'acquisition x_t^j n'est similaire à aucun objet précédemment entendu de niveau j : il y a donc mémorisation du matériau correspondant. Par ailleurs, il y a *intégration* au sens que l'objet est concaténé aux objets précédents entendus depuis la dernière segmentation (construction du facteur $w_k^j = x_{t-i}^j \dots x_t^j$ où $t - i > 0$ est l'indice de dernière segmentation) et le sujet qui écoute est en train de construire un objet de niveau supérieur. Lors de l'acquisition de x_{t+1}^j , il n'y a cependant pas de segmentation : x_t^j ne clôt pas l'objet en voie de constitution $x_{t-i}^j \dots x_t^j$ et rien ne se passe au niveau supérieur.

Voici ce qu'il se passe en terme d'hypothèses. Étant donné que l'objet $x_t^{(j)}$ correspond à un nouveau matériau acquis au niveau actuel, aucune hypothèse spécifique n'est émise concernant l'objet à suivre $x_{t+1}^{(j)}$, ni pour l'objet de niveau supérieur $x_k^{(j+1)}$. Plus précisément, une hypothèse de nouveauté est effectuée : le sujet de l'écoute attend l'inconnu.

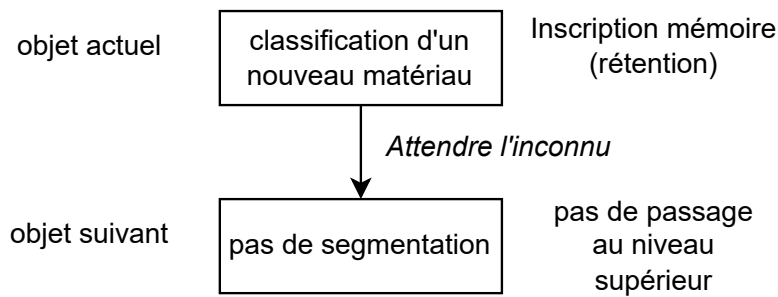


FIGURE 4.11 – Phase 1.

Phase cognitive 2

La phase cognitive 2 correspond à un test de similarité invalide et segmentation valide. On a donc mémorisation et rétention dans la mémoire de l'objet en cours d'acquisition x_t^j comme pour la phase 1. Cependant, il y a cette fois-ci segmentation avant x_{t+1}^j lors de son acquisition, ce qui signifie qu'il y a acquisition de l'objet en voie de constitution $w_k^j = x_{t-i}^j \dots x_t^j$ et donc labélisation (par opération de classification) en un objet de niveau supérieur, et appel récursif de l'algorithme MORFOS.

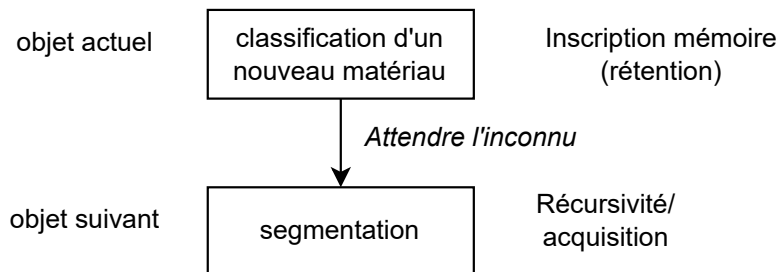


FIGURE 4.12 – Phase 2.

De la même manière que pour la phase cognitive 1, un nouveau matériau est acquis au niveau actuel, une hypothèse de nouveauté est donc émise concernant l'objet à suivre. Nous avons par ailleurs acquisition d'un matériau probablement nouveau au niveau supérieur (sauf s'il s'agit d'une variation considérée comme négligeable) pour l'objet $x_k^{(j+1)}$, et une hypothèse de nouveauté est émise concernant l'objet suivant $x_{k+1}^{(j+1)}$. Le sujet de l'écoute attend donc l'inconnu, autant au niveau actuel qu'au niveau supérieur.

Phase cognitive 3

La phase cognitive 3 correspond à un test de similarité valide et segmentation invalide. L'objet en cours d'acquisition x_t^j est similaire à un objet précédemment entendu. L'instant temporel associé t est mémorisé mais le matériau correspondant n'est pas mémorisé à nouveau. L'objet est concaténé à l'objet en voie de constitution et il y a une phase de *réalisation* : les matériaux potentiels sont concrétisés dans la mémoire. Comme pour la phase 1, il n'y a cependant pas de segmentation après l'acquisition de l'objet successeur x_{t+1}^j .

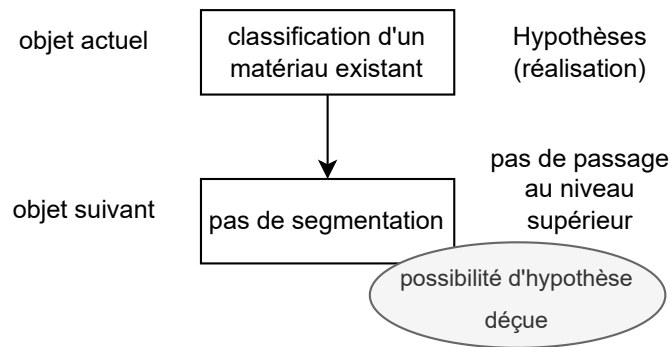


FIGURE 4.13 – Phase 3.

La reconnaissance en x_t^j d'un matériau déjà perçu va permettre d'émettre des hypothèses concernant l'objet successeur x_{t+1}^j qui est attendu au niveau actuel. Les hypothèses sont alors émises en fonction des objets qui ont succédé les occurrences précédentes du matériau correspondant à x_t^j . Ceci est permis par ailleurs par le fait qu'il n'y a pas de segmentation après $x_t^{(j)}$. En effet, s'il y avait segmentation après $x_t^{(j)}$, alors l'objet x_{t+1}^j correspondrait au début d'un nouvel objet au niveau j et aucune hypothèse ne serait effectuée (cas de la phase 4). Ici des hypothèses sont émises : le sujet de l'écoute attend le connu. Cependant, il est possible que l'objet x_{t+1}^j ne corresponde pas aux matériaux attendus : il y aura alors un phénomène de surprise.

Dans notre modélisation, c'est la structure de l'Oracle Variable de Markov qui nous permet de retrouver les occurrences précédentes et leurs successeurs qui sont des candidats potentiels pour $x_{t+1}^{(j)}$.

On note qu'à ce stade, des hypothèses peuvent également être émises sur l'objet $x_k^{(j+1)}$ de niveau supérieur : soit l'objet en voie de constitution w_k^j est le préfixe d'un ou plusieurs objets de niveau supérieur déjà vus et les objets en question sont des hypothèses de réalisation, soit l'objet en voie de constitution ne correspond à

aucun préfixe et alors on attend de l'inconnu au niveau supérieur. En remontant les niveaux, on augmente ainsi potentiellement l'empan d'anticipation.

Phase cognitive 4

La phase cognitive 4 correspond à un test de similarité valide (réalisation) et également un test de segmentation valide (acquisition au niveau supérieur et récursivité de l'algorithme).

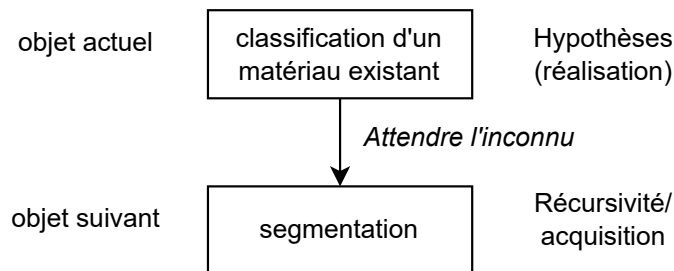


FIGURE 4.14 – Phase 4.

Comme pour la phase 3, le fait d'acquérir un objet x_t^j reconnu permet à priori d'émettre des hypothèses concernant l'objet $x_{t+1}^{(j)}$, seulement il y a en réalité segmentation entre ces deux objets. L'objet $x_{t+1}^{(j)}$ est donc le début d'un nouveau facteur w_{k+1}^j et aucune hypothèse n'est effectuée au niveau j : l'inconnu est attendu.

Au niveau supérieur en revanche, il est possible que l'objet x_k^{j+1} obtenu corresponde à un matériau déjà entendu au niveau $j + 1$: on peut donc émettre une hypothèse sur l'objet successeur x_{k+1}^{j+1} .

4.3.2 Analyses

Le calcul des phases en temps réel a ainsi été implémenté dans MORFOS.

Voici les résultats obtenus pour différents types de segmentation sur le Rondo de Mozart :

- aucune segmentation ;
- les facteurs de taille deux sont segmentés ;
- une segmentation aléatoire ;

- un ensemble 1 de règles de segmentation pertinentes ;
- un ensemble 2 de règles de segmentation pertinentes.

Nous illustrons la segmentation aléatoire par un exemple spécifique ici.

Aucune segmentation

On présente dans le tableau suivant les phases correspondant à une analyse sans segmentation :

niveau	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	a	b	a	c	r	s	r	s	t	t	u
0	1	1	3	1	3	3	3	3	1	1	3	3	1	1	3	3	3	3	1	1	1	1	1	1	3	3	1	1	1	1	3	3	3	3	1	1	3	3	1	3	2

La première ligne correspond aux états acquis au niveau 0 et chaque ligne supplémentaire correspond à un niveau et contient les phases correspondantes à l'acquisition des états associés à la segmentation de l'état au niveau 0. Dans ce cas-ci, on constate qu'il n'y a que les phases 1 et 3 : on retrouve la phase 1 lors de l'acquisition d'un nouveau matériau et la phase 3 lors du retour à un matériau déjà vu, mais il n'y a jamais segmentation, à l'exception du dernier matériau qui est considéré comme segmenté par défaut (et n'a jamais été vu, ce qui donne la phase 2).

Segmentation tous les 2 objets

On présente dans le tableau suivant les phases cognitives obtenues lors d'une analyse où les facteurs de taille deux sont segmentés :

niveau	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	a	b	a	c	r	s	r	s	t	t	u			
0	1	2	3	2	3	4	3	4	1	2	3	4	1	2	3	4	3	4	1	2	1	2	1	2	3	4	1	2	1	2	3	4	3	4	1	2	3	4	1	3	2			
1		1		2	3		3		4	1		4	1		3	4		3		2	1		2	1		3		2	1		3	4		3		2	1		3	2				
2			1					4			1				2					1				2				1				2					1				2	2		
3				1								1				4								1								2										2	2	
4					1											1																											2	2

La segmentation tous les deux objets présente un coup sur deux la phase 1 ou 3 et un coup sur deux la phase 2 ou 4, selon que l'objet acquis a déjà été en mémoire ou non.

Segmentation aléatoire

Nous présentons dans le tableau suivant les phases cognitives associées à la segmentation aléatoire 2 de la section précédente :

4. Un Modèle d'Écoute : coûts et cognition

niveau	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	a	b	a	c	r	s	r	s	t	t	u	
0	1	2	3	1	3	3	3	4	1	1	3	3	2	1	3	4	3	4	1	2	1	2	1	1	3	4	1	2	1	1	4	3	3	3	2	1	4	3	2	3	2	
1		1						2					1					1		2		1				2										2		1		1		2
2								1								2				1						1											1					2
3																1																										2

Les phases obtenues semblent également aléatoires entre 1 et 2 lorsqu'il n'y a pas segmentation (selon que l'objet ait déjà été acquis ou non) et entre 3 et 4 s'il y a une segmentation aléatoire

Ensemble A de règles de segmentation de l'agenda

Voici le tableau des phases correspondantes à l'ensemble de règles A telles que définies dans la partie 4.2.4 de la section précédente :

niveau	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	a	b	a	c	r	s	r	s	t	t	u		
0	1	2	3	2	3	4	3	4	1	2	3	4	1	2	3	4	3	4	1	1	1	1	1	2	3	3	1	1	1	2	3	4	3	4	1	2	3	4	1	3	2		
1		1		2		3		4		1		3		2		3		4						1												4		1		3		2	
2				1				2						2				4																									2
3														1				3																									2

On peut comparer cet exemple avec celui présenté par J.M. Chouvel dans [32], que l'on rappelle en figure 4.15.

C'est l'occasion pour nous d'introduire ici une autre représentation présentée par Jean-Marc Chouvel de l'œuvre appelée *représentation structurelle* : ici le sens des embranchements en plus d'informer sur la structure de l'œuvre, informe sur le phénomène de continuité de l'information (branchement vers l'arrière) ou d'anticipation (branchement vers l'avant). Les branchements vers l'avant qui sont croisés constituent ainsi des anticipations déçues.

On obtient bien les mêmes segmentations que celles obtenues par Chouvel, à quelques exceptions près. La première est le dernier caractère *u*, qui est considéré comme inclu dans le dernier *t* de l'analyse de Chouvel : alors qu'il s'agit pour lui d'un caractère répété et segmentant (phase 4), nous le divisons en deux caractères *t* et *u*, le premier étant répété et non segmentant (phase 3) et le deuxième étant nouveau et segmentant (phase 2).

La deuxième différence consiste en le déroulement des deux développements (*hijklm*) et (*hinopq*). Chouvel considère ces deux développements comme segmentés en deux fois trois groupes de deux objets (*hi*)(*jk*)(*lm*) et (*hi*)(*no*)(*pq*) tandis que nous ne segmentons que le premier facteur (*hi*) ainsi, alors que Chouvel constate les phases 2 à l'acquisition du premier *i*, du *k* et *o*, nous considérons ici les phases 1 pour de nouveaux objets sans segmentation, et la phase 3 pour le

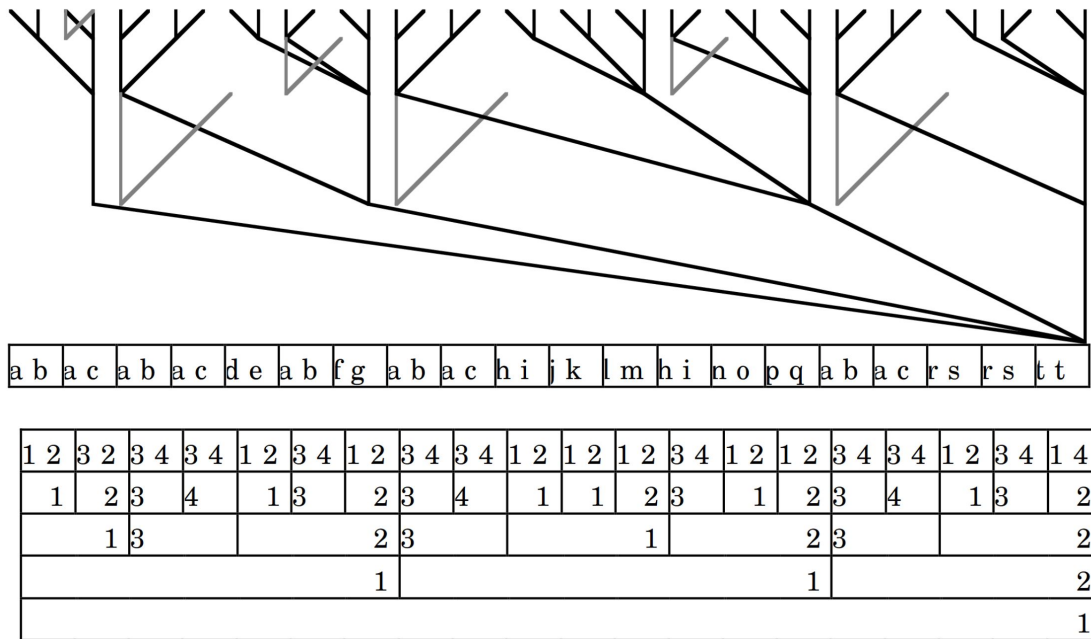


FIGURE 4.15 – Phases cognitives du *Rondo K545* de W.A. Mozart selon Chouvel dans [32].

deuxième i (objet déjà vu sans segmentation) là où il considère la phase 4 car il y a segmentation. Les objets segmentés au niveau 0 sont alors répercutés au niveau supérieur 1 en trois objets qui n'ont eux pas d'impact sur le niveau 2.

Ensemble B de règles de segmentation de l'agenda

Voici le tableau des phases correspondants à l'ensemble de règles B telles que définies dans la section précédente :

niveau	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	a	b	a	c	r	s	r	s	t	t	u					
0	1	2	3	2	3	4	3	4	1	2	3	4	1	2	3	4	3	4	1	2	1	1	1	2	3	4	1	1	1	2	3	4	3	4	1	2	3	4	1	3	2					
1		1		2		3		4		1		3		2		3		4		1		1		2		3		1		2		3		4		1		3		2		2				
2				1				4					2					3						1						2				4								2		2		
3												1																		1														2		2

La différence principale avec l'exemple précédent une fois encore est l'ensemble des phases cognitives lors des deux développements (*hijklm*) et (*hinopq*). Cette fois-ci, il y a segmentation après les deux acquisitions de *i*, ce qui transforme les phases 1 et 3 en phases 2 et 4 et les objets de niveau supérieur 1 sont créés en

4. Un Modèle d'Écoute : coûts et cognition

conséquence et donnent lieu à l'apparition de phases cognitives 1 et 3 au niveau supérieur. Elles n'induisent pas de segmentation et n'impactent pas le niveau 2.

Analyse des phases et calcul des coûts

On peut par ailleurs comparer les coûts spécifiques à l'acquisition de chaque objet calculé dans la section précédente avec les phases cognitives associées.

Par exemple, sur le diagramme formel multi-échelle obtenu avec l'ensemble de règles A, on recense les phases et les coûts associés dans le tableau suivant :

niveau	a	b	a	c	a	b	a	c	d	e	a	b	f	g	a	b	a	c	h	i	j	k	l	m	h	i	n	o	p	q	
0	1	2	3	2	3	4	3	4	1	2	3	4	1	2	3	4	3	4	1	1	1	1	1	2	3	3	1	1	1	2	
coût	0.0	2.0	1.5	7.0	3.0	4.1	1.0	3.1	9.5	12.0	7.0	6.1	15.5	18.0	11.0	8.1	6.0	7.1	22.5	24.0	26.0	28.0	30.0	32.0	20.8	10.1	36.0	37.4	39.5	41.6	
1		1		2		3		4		1		3		2		3		4						1							2
coût		0.0		3.0		1.5		2.0		9.0		4.0		11.5		4.2		4.1						16.5						18.0	
2				1				3						2				3													2
coût				0.0				0.0						4.0				1.3													9.1
3														1																	1
coût														0.0																	3.0

niveau	a	b	a	c	r	s	r	s	t	t	u
0	3	4	3	4	1	2	3	4	1	3	2
coût	24.7	12.1	10.0	11.1	47.5	48.0	30.5	14.0	52.0	33.0	55.0
1		3		4		1		3			2
coût		9.0		6.1		24.5		12			26.0
2				3							2
coût				3.0							13.1
3											2
coût											6.0

Si l'on calcule la moyenne des coûts associés à chacune des phases cognitives sur l'exemple précédent, on obtient les valeurs suivantes :

- phase 1 : $420,9/20 = 21,1$
- phase 2 : $306,3/16 = 19,1$
- phase 3 : $187,6/19 = 9,9$
- phase 4 : $77,9/11 = 7,1$

Les phases 3 et 4 ont un coût moyen 2 à 3 fois inférieur aux phases 1 et 2. Celles-ci correspondent à l'acquisition d'un objet dont le matériau correspondant a déjà été entendu en mémoire : sans segmentation pour la phase 3, et avec segmentation pour la phase 4. On constate ainsi que l'acquisition d'un matériau déjà entendu induit un coût moins important dans cet exemple, et d'autant plus lorsqu'il y a segmentation.

Une perspective par ailleurs serait de calculer les phases cognitives secondaires, que l'on peut définir comme la *dérivée* des phases cognitives, autrement dit, l'enchaînement de celles-ci. Ainsi, J.M. Chouvel présente dans [32] 16 phases cognitives, correspondant à la combinatoire de la succession des 4 phases existantes.

4.4 Comparaison aux modèles cognitifs

L'Algorithme Cognitif et MORFOS étant présentés comme un modèle d'écoute, il convient de comparer le comportement du logiciel avec celui d'un individu réel lors d'une écoute musicale active.

En effet, une corrélation des phénomènes réels liés à l'écoute musicale avec les résultats de MORFOS pourrait d'une part valider les résultats obtenus, d'autre part, peut-être nous permettre de mieux comprendre l'écoute réelle en étudiant le comportement de l'algorithme. En revanche, une dé-corrélation n'invalide pas nécessairement les résultats de MORFOS mais ne nous permettra pas d'émettre d'hypothèse particulière à partir de celui-ci concernant l'écoute réelle.

Cette partie prend en compte les travaux pour cette thèse effectués par Anasztazja Adamczyk, étudiante en Sciences Cognitives, dans le cadre de son stage [1].

4.4.1 Modèles d'écoute par EEG

Stefan Koelsch et Walter A. Siebel [79] proposent un modèle du traitement cognitif de la musique. On peut distinguer les premières étapes de traitement, qui correspondent aux potentiels jusque 400-600ms de latence, des processus de haut niveau liés à l'interprétation et aux réactions physiques. Les premières étapes de traitement peuvent être assimilés au traitement de MORFOS tandis que les processus de haut niveau sont suggérés par la représentation obtenue qui est laissée au soin de l'interprétation du musicologue. Nous présentons le modèle dans cette partie.

Acquisition du signal sonore

Dans un premier temps, les ondes sonores parviennent à l'oreille et sont traitées par celle-ci. L'information acoustique est alors traduite en activité neuronale dans la cochlée, et les différentes caractéristiques musicales telles que la hauteur des notes ou le timbre sont une première fois traduites en différentes réponses neuronales au niveau du colliculus supérieur et du thalamus. Les informations provenant du cortex auditif primaire sont ensuite envoyées à d'autres structures comme l'amygdale et le cortex orbito-frontal médian qui sont également responsables du contrôle des émotions et du comportement.

Dans notre modèle, cette étape peut être mise en comparaison avec l'étape de numérisation dans le module de pré-traitement du signal.

Extraction des caractéristiques musicales

Dans un second temps, des processus plus précis d'extraction des caractéristiques musicales ont lieu dans le cortex auditif. Les informations relatives à la hauteur du son, aux rapports d'octave, au timbre, à l'intensité et à la rugosité sont extraites et représentées dans des potentiels évoqués de 10 à 100 ms après l'apparition du stimulus.

Cette étape peut être comparée à la deuxième partie du module de pré-traitement du signal pour obtenir la donnée dans un format adéquat à l'Algorithme MORFOS. Dans notre cas, on filtre le signal en fonction de la caractéristique qui nous intéresse (fft pour la hauteur des notes et mfcc pour le timbre par exemple). Les caractéristiques peuvent également être fournies directement sous forme de vecteurs calculés par un module externe comme celui proposé par A. Marmoret avec des réseaux de neurones [95], ou bien par de nouveaux modules internes implémentés à fortiori.

Groupement des événements sonores

Une fois les caractéristiques musicales extraites, la mémoire sensorielle auditive rassemble ces éléments musicaux de premier niveau en respectant les lois de la Gestalt [50]. Dans le domaine musical, cela englobe le groupement des événements sonores mélodiques, rythmiques, timbraux et spatiaux. Ainsi, en réponse à une incohérence musicale ou à un nouveau stimulus, le cortex auditif génère une réaction spécifique avec une latence de 100 à 200 ms : la Mismatch Negativity (MMN) [164]. Toutefois, cette réaction peut varier en fonction de l'expérience individuelle, des traits de personnalité ou de l'éducation musicale [76]. Selon S. Koelsch et W. A. Siebel, cette étape permet également de distinguer le flux d'une conversation de l'environnement sonore.

On peut émettre l'hypothèse que l'on peut rapporter les stimulus de Mismatch Negativity à l'opération de segmentation de premier niveau : en effet, dans notre modèle, un élément différent du précédent implique dans le logiciel à très bas niveau (uniquement) une segmentation. Cependant, le fait de considérer deux éléments comme différents ou non peut varier en fonction de la connaissance au préalable. Celle-ci n'est pas inscrite dans le modèle mais est introduite par la notion du seuil

de segmentation. Par ailleurs, on ne gère pas ici l'extraction du flux par rapport à l'ensemble des données fournies au logiciel tels que les bruits environnants ou la polyphonie.

Analyse des intervalles

L'étape suivante implique l'analyse des relations entre les événements sonores en se concentrant sur les intervalles. Selon Levitin [89], les individus peuvent percevoir et distinguer les intervalles musicaux indépendamment de leur formation musicale. Il présente l'exemple suivant : lorsqu'une chanson est chantée, il arrive souvent que nous la chantions involontairement dans une tonalité différente, sans tenir compte de la hauteur absolue des notes. Cet effet, connu sous le nom d' "*effet Levitin*", résulte de notre aptitude à mémoriser non pas les hauteurs absolues, mais les intervalles musicaux. Selon Koelsch et Siebel [79], il existe deux types de perception des intervalles : les intervalles mélodiques et les intervalles temporels. Les intervalles mélodiques concernent le traitement des relations de hauteur entre les tons d'une mélodie ou d'un accord, tandis que les intervalles temporels se réfèrent aux relations temporelles et, par extension, aux rythmes.

Dans MORFOS, ce comportement peut être assimilé aux tests de similarité des deuxièmes niveaux et suivants, dès la détection de deux objets de niveau supérieur associé au calcul de similarité entre ces éléments. Cependant, le logiciel effectue pour le moment une labélisation des hauteurs pures et non pas des intervalles : il s'agit de notions fondamentales à tester et qu'il reste à implémenter.

Analyse structurelle

Ensuite, l'analyse de la structure musicale intervient, nécessitant le calcul de relations structurelles entre les éléments. Par exemple, on considère la relation entre un accord et l'harmonie qui le précède. Diverses études démontrent que la formation musicale n'est pas nécessaire pour traiter la syntaxe musicale, car cette capacité est intrinsèque [117] [127]. Ainsi, bien que les musiciens soient plus compétents pour détecter les incohérences musicales et syntaxiques, les auditeurs non spécialisés sont également conscients de celles-ci [10] [152]. De plus, des processus similaires semblent exister pour le traitement du rythme et de la métrique [157]. Dans les musiques tonales, les musiciens manifestent des potentiels évoqués ERAN en réponse aux violations des règles harmoniques lors de l'écoute musicale, ce qui permet d'analyser le traitement de la syntaxe musicale. Par ailleurs, le potentiel P600 émerge pour des processus syntaxiques plus hauts niveaux, avec une latence

d'environ 600 ms.

Cette étape du modèle peut alors s'apparenter à l'ensemble des règles syntaxiques de segmentation que l'on a mis en place à l'échelle symbolique pour MORFOS.

Attribution d'une signification

Une fois le phénomène musical traité à bas niveau, Koelsch, S. et al. [78] montrent également que le traitement de la syntaxe est probablement lié au traitement du sens et des émotions lors de l'écoute musicale : l'étape suivante vise ainsi à attribuer une signification aux événements musicaux entendus. Bien que définir une signification musicale puisse s'avérer complexe, contrairement à la signification empirique ou apprise d'une phrase linguistique, diverses études montrent que la sémantique musicale existe et partage des caractéristiques similaires à la sémantique linguistique. Koelsch aborde dans [77] trois dimensions de la signification musicale : le sens intra-musical (*intra-musical meaning*), le sens extra-musical (*extra-musical meaning*) et le sens musicogénique (*musicogenic meaning*). Le sens extra-musical englobe des représentations ou qualités qui peuvent évoquer des sons du quotidien, des intentions liées aux émotions ou encore des symboles culturels. Le sens intra-musical dépend de la structure et de la forme musicale, tandis que le sens musicogénique est suscité par les réactions personnelles (mouvements, émotions) et les associations mémorielles [79]. Les potentiels évoqués qui semblent refléter le traitement de la signification musicale incluent le N400, lié au traitement du sens extra-musical et émergeant en réponse à des incohérences sémantiques [82], et le N5, associé au traitement du sens intra-musical.

Dans notre cas, MORFOS n'extrait pas le sens associé aux événements musicaux. On se contente par ailleurs de labeliser les différents matériaux de manière arbitraire. Ce travail est laissé aux musicologues, qui traitent la représentation obtenue de la forme avec leur connaissance sémantique des matériaux manipulés.

Réponse motrice

Lors de l'écoute musicale nous générons souvent une réponse motrice [60]. Même si nous ne bougeons pas réellement, la musique génère une activité dans notre cortex pré-moteur, montrant ainsi que le mouvement est fortement lié à la musique et que la danse est un phénomène profondément associé à notre biologie.

Si le logiciel n'offre pas un traitement des données vers la motricité, les représentations qu'il offre pourraient être ajoutées à un module supplémentaire de génération pour aider à l'interprétation musicale et la performance artistique. À l'inverse, le logiciel pourrait également traiter des données obtenues à partir du mouvement musical, que l'on pourrait comparer avec la donnée musicale, ou bien obtenir des représentations formelles des mouvements afin de créer la musique.

Schéma du modèle

On représente le modèle de Koelsch et Siebel mis en parallèle avec les étapes de MORFOS en figure 4.16. Les étapes de MORFOS sont représentées dans les blocs jaunes à bord épais et les blocs orange à bords pointillés correspondent aux analyses et applications de MORFOS.

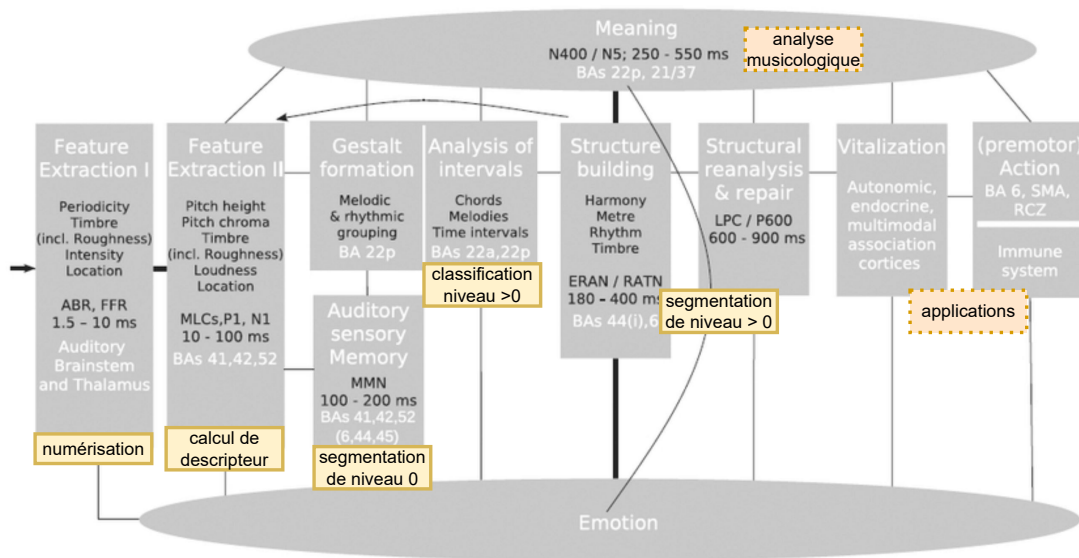


FIGURE 4.16 – Modèle de S. Koelsch et W.A. Siebel [79] et comparaison avec les étapes de MORFOS.

Potentiels impliqués

Certains potentiels impliqués dans les processus de l'écoute musicale sont les suivants :

- Le potentiel MMN (Mismatch Negativity) semble refléter le traitement musical dans la mémoire auditive. Les stimuli musicaux sont initialement encodés comme une répétition périodique d'un phénomène acoustique. Une réponse MMN est déclenchée lorsque nous sommes habitués à un stimulus et détectons un changement soudain.
- L'activité ERAN (Early Right Anterior Negativity) se produit lors de violations des règles musicales et a été introduite par Stefan Koelsch [76] par opposition à l'activité ELAN (Early Left Anterior Negativity) qui se produit pour les stimuli linguistiques qui violent les règles de structure des catégories de mots ou des phrases. ERAN, dans l'hémisphère droit, traite l'information en mémoire à long terme, tandis que MMN concerne les incohérences dans la mémoire à court terme.
- Les signaux P300 (P3a et P3b) réagissent aux stimuli musicaux et non musicaux, liés à l'attention (P3a) et aux processus de résolution incertaine (P3b).
- N400 est utilisé pour les violations de règles sémantiques en langage [78] et dans un contexte musical pour des mélodies familières [26] mais son application pour cette dernière est discutée, la notion de sémantique musicale étant elle-même remise en question [149].
- La composante P600 est cruciale en langage et musique, réagissant aux incongruités linguistiques syntaxiques et musicales harmoniques. plus l'accord est éloigné de la tonalité principale du morceau musical, plus la composante est importante [118].
- Enfin, les LPC et CPS sont deux potentiels similaires d'une latence de 500ms. Le LPC apparaît en relation avec les stimuli anciens appris [116], tandis que le CPS peut être identifié aux frontières de la phrase linguistique ou musicale [115].

Dans le cadre de MORFOS, les potentiels qui semblent les plus pertinents sont les potentiels détectant l'attention et la fin de phrases (MMN, ERAN, CPS). Cependant, le contexte musical est primordial dans la reconnaissance d'un motif, c'est pourquoi il est nécessaire pour observer ces potentiels d'effectuer des études sur des musiques spécialement composées qui permettent de contrôler tous les facteurs tel que la transposition lors de la reconnaissance d'un motif, ce qui rend une telle étude contraignante.

Dans le cadre de cette thèse, nous n'avons eu ni le temps ni les moyens matériels pour mettre en œuvre une expérience impliquant une analyse électro-encéphalographique, mais cette étude pourrait servir de point de départ à d'autres explorations EEG sur le sujet.

4.4.2 Calcul de la charge cognitive par étude double tâche

Nous avons mis en place une expérience nous permettant de mesurer la charge cognitive pour une tâche de segmentation lors de l'écoute musicale active, à partir d'un protocole double tâche.

Mesures de la charge cognitive

Plusieurs procédés peuvent être utilisés pour étudier les phénomènes cognitifs liés à l'écoute musicale. Un premier procédé est une expérience basée sur électroencéphalographie. Cela permettrait de mesurer l'activité cérébrale liée à une situation donnée. On place pour cela un ensemble d'électrodes sur le cuir chevelu permettant de mesurer l'activité électrique provenant du cortex cérébral. L'objectif est de mesurer les potentiels évoqués liés à la tâche que nous souhaitons mesurer et de nous intéresser aux potentiels liés à l'écoute musicale, et plus précisément au phénomène de reconnaissance et de segmentation des objets musicaux tels que présentés précédemment.

A défaut de pouvoir mesurer directement l'activité cérébrale, une autre possibilité est de mesurer la charge cognitive liée à l'activité musicale. L'oculométrie permet de mesurer la charge cognitive grâce à la mesure de la dilatation pupillaire [74] [73]. C'est une méthode utilisée par exemple par T. Endestad et al [49] pour mesurer *l'effort musical* d'un pianiste lorsqu'il joue, écoute et imagine la musique ou pour mesurer la charge cognitive liée au chant en simultané du jeu musical chez le musicien [122].

Dans le cadre de cette thèse, nous cherchons à mesurer la charge cognitive liée à la segmentation lors de l'écoute musicale. Une idée de protocole par oculométrie est donc de proposer à un groupe d'étudiants musiciens une tâche d'écoute active avec annotation des fins de sections et phrases, tout en mesurant les informations sur la charge cognitive grâce à la dilatation pupillaire, et de proposer par ailleurs un questionnaire sur le profil, la charge cognitive et la charge émotionnelle liée à la tâche effectuée. Une difficulté liée à l'oculométrie est en effet de distinguer la charge cognitive de la charge émotionnelle.

L'expérience que nous avons choisi de mener est de une étude double tâche. En effet, les études doubles tâches sont largement utilisées pour l'analyse de la charge cognitive et pour la charge cognitive associée à l'écoute des sons par exemple pour la mesure de la charge cognitive associée à l'écoute dans un environnement bruyant [41] ou encore pour la mesure de la charge cognitive associée à un système

de sonification de données d'impression 3D et la reconnaissance de modification de comportements sonores lors de l'effectuation d'une tâche secondaire [123]. Le paradigme double-tâche est précisément utilisé dans la mesure de la charge cognitive lors de l'écoute musicale par exemple pour évaluer la perception du temps en fonction de la charge cognitive lors de l'écoute musicale [163] ou par M. Imberty pour déterminer le style musical induit de la structure de l'œuvre déterminée lors d'écoutes actives [71].

Objectifs de l'expérience

L'objectif de l'expérience est multiple :

- utiliser la complexité pour évaluer la pertinence des segmentations ;
- comprendre la charge cognitive liée à l'écoute musicale ;
- comparer les coûts computationnels entre différentes œuvres musicales ;
- comparer la charge cognitive et les coûts computationnels de MORFOS.

Protocole expérimental

L'étude mise en place est finalement un paradigme double tâche : La tâche principale consistait à effectuer une écoute active avec annotation des fins de phrase et fins de sections, tandis que la tâche secondaire nécessitait de reproduire une succession de séquences de caractères "X" et "O".

Présentation des tâches

Les deux tâches sont décrites de la manière suivante :

- **Tâche de segmentation lors de l'écoute active :**
les participants écoutent une œuvre et doivent indiquer à l'aide du clavier les fin de phrases en cliquant sur la touche 'T' et les fins de section en cliquant sur la touche 'G' (les définitions sont précisées ci-dessous et fournies en début d'expérience). Le rappel des touches adéquates était constamment affiché à l'écran.
- **Tâche de reproduction de séquences de caractères :**
les participants avaient à l'écran des séquences de caractères constituées de croix et de ronds (figure 4.17). Ceux-ci avaient deux boutons avec pour l'un une croix dessus et pour l'autre un rond, et ils devaient cliquer dans l'ordre sur les boutons adéquats avec la souris afin de reproduire l'exacte séquence affichée. Lorsqu'une séquence est reproduite entièrement, une nouvelle est affichée automatiquement.

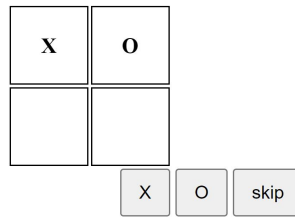


FIGURE 4.17 – Tâche de reproduction de séquences.

Cette deuxième tâche est inspirée de l'expérience de M. Poret dans [123] : la tâche principale consistait dans son expérience à annoter les changements de timbres dans un écosystème de données sonifiées pour alerter des problèmes liés à une tâche d'impression 3D. La tâche secondaire était la reproduction de séquences de 'X' et de 'O', permettant de mesurer ainsi la charge cognitive associée au temps de réaction lié aux alertes données par les sons choisis pour les différentes données.

Cependant, nous avons apporté une légère modification à l'expérience de M. Poret. En effet, celui-ci a avancé que cette tâche pouvait être un peu trop systématique et que les participants pouvaient cliquer aléatoirement jusqu'à la reproduction de la séquence : nous avons légèrement modifié l'interface, de telle sorte à ce que les séquences puissent s'afficher aléatoirement en bleu, et alors l'utilisateur ne doit pas reproduire celle-ci mais appuyer sur le bouton *skip*. Cela permet de casser la monotonie de la tâche de reproduction des séquences.

Groupes d'individus

Les participants sont divisés en deux groupes d'individus. Un premier groupe effectue les deux tâches en simultané, tandis que le second groupe n'effectue que la tâche de segmentation. La répartition des groupes est de 50% des participants dans chacun des deux groupes, répartis de manière aléatoire.

Déroulé de l'expérience

L'expérience se déroule en 4 étapes :

La première étape est une phase d'explications. Dans un premier temps, le déroulé de l'expérience est précisé. Ensuite, un ensemble de définition est présenté. On y retrouve les notions *d'unité*, de *phrase musicale* et de *section musicale*. La figure 4.18 décrit les notions telles qu'elles sont définies et expliquées dans l'expérience.

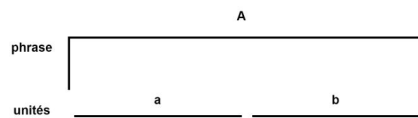
Bien que le choix des définitions soit discutable, l'objectif de ces définitions est que les participants s'y réfèrent pour avoir une référence concernant les échelles de segmentation demandées. Cette considération est spécifiée avant le déroulé de

Quelques définitions:

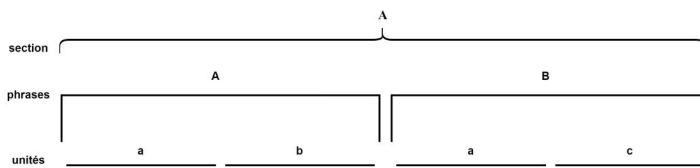
On appelle une **unité** un élément de quelques secondes qui est reconnaissable à l'échelle macroscopique. Il s'agit par exemple d'un motif ou d'une carrure. En linguistique, différentes unités seraient séparées par des virgules.



- Une **phrase musicale** est une idée conclusive. Elle est constituée de plusieurs unités et à l'image d'une phrase linguistique, elle contient potentiellement plusieurs virgules et se conclue par un point.



- Une **section musicale** est un ensemble de phrases. Equivalente à une strophe en poésie, elle est comme une histoire. Un changement de section induit un changement d'ambiance.



Vous indiquerez les fins de phrases musicales et de sections musicales. Essayez de suivre vos intuitions en distinguant les deux.

FIGURE 4.18 – Définitions introduites dans l'expérience cognitive.

l'expérience.

Ensuite, une phase d'entraînement est présentée. Durant cette phase, le participant procède à la tâche de reproduction de séquence pendant une minute uniquement, à la tâche de segmentation sur un morceau d'exemple court, puis en fonction du groupe d'appartenance du participant, soit à un deuxième exemple de segmentation uniquement, soit aux deux tâches en simultané. Des propositions de segmentation sont fournies en vidéo après chacun des deux exemples. Cette phase est reproductible autant de fois que l'utilisateur le souhaite. Les participants peuvent par ailleurs poser les questions qu'ils souhaitent.

Après la phase d'entraînement a lieu la phase expérimentale. Celle-ci se déroule en trois étapes :

- Une première étape est la reproduction de séquence pendant 4 minutes (durée moyenne des morceaux) sur du silence.
- La deuxième étape est la tâche de segmentation uniquement pour les deux groupes d'individus sur 3 morceaux différents
- La troisième étape est la tâche soit double, soit de segmentation uniquement selon le groupe d'individu sur 9 morceaux musicaux.

A la fin de chacune des treize tâches, à savoir la reproduction des séquences en silence puis l'écoute de douze morceaux, le participant indique s'il a déjà entendu le morceau et répond à un questionnaire NASA-TLX.

Enfin, les individus répondent à un questionnaire final concernant leur profil musical. L'ensemble des questionnaires utilisés pour l'expérience sont décrits ci-dessous.

Répertoire musical

Deux exemples et douze œuvres musicales sont présentées durant l'expérience. Les exemples durent une à deux minutes, tandis que les œuvres de l'expérience durent en moyenne 4 minutes. Le choix des œuvres a été fait de telle sorte à éviter les paroles, afin que la segmentation ne soit pas faite en fonction de l'intonation ou de la sémantique. Par ailleurs, nous souhaitons nous situer dans le même contexte que MORFOS, c'est-à-dire pour la première écoute d'une œuvre : nous avons donc souhaité choisir le plus possible des œuvres qui n'étaient pas connues malgré une cible musicienne. Par ailleurs, nous avons choisi les œuvres en fonction de trois niveaux *complexité musicale* subjectifs : *facile*, *moyen* et *difficile*. La complexité musicale est ici définie comme la quantité d'information dans l'œuvre musicale décrite en tant qu'entropie. Parmi ses composants principaux, les figures rythmiques, les schémas harmoniques, le nombre d'instruments, et le nombre de voix.

Nous avons donc un morceau par difficulté dans la première phase expérimentale de segmentation uniquement et trois œuvres par niveau de difficulté dans la deuxième phase expérimentale. Les œuvres ont été sélectionnées par Elia Bichoy, musicologue, et sont référencées en annexe D.

Les morceaux sont par ailleurs joués de manière aléatoire parmi les trois morceaux de la première phase puis parmi les 9 morceaux de la deuxième phase. Il est précisé aux participants que certains morceaux peuvent être très difficiles à segmenter.

Contexte et participants

L'expérience a eu lieu en présentiel pour l'intégralité des participants, par groupe

de 4 personnes maximum espacées d'au minimum un mètre. Chaque participant était muni d'un ordinateur portable ainsi que d'un casque, et tous avaient une souris à l'exception de deux individus qui ont utilisé leur pavé tactile. Les participants pouvaient poser les questions qu'ils souhaitaient jusqu'à la phase d'entraînement incluse.

Questionnaires

Deux questionnaires sont utilisés dans le cadre de cette expérience. Le Questionnaire NASA-TLX est utilisé à chaque écoute d'un nouveau morceau, et un questionnaire sur le profil du participant est présenté à la fin de l'expérience.

NASA-TLX

Le questionnaire NASA-TLX [65] permet de mesurer subjectivement la charge cognitive associée à la tâche évaluée. Il est découpé en deux phases : la première phase est la présentation de six échelles de critères pour évaluer la difficulté de la tâche. Les critères sont *l'exigence mentale*, *l'exigence physique*, *l'exigence temporelle*, *la performance*, *l'effort* ainsi que la *frustration*. Les échelles contiennent vingt niveaux, allant de l'exigence la plus faible à l'exigence la plus élevée. Chaque critère est défini à côté de l'échelle correspondante. On note que la formulation des définitions est fondamentale pour la pertinence du questionnaire, c'est pourquoi nous avons choisi d'utiliser la traduction française qui a fait l'objet de validations dans [28]. Les échelles et la définition des critères sont présentées en figure 4.19. L'individu remplit donc les six échelles en fonction de son expérience concernant la tâche qu'il vient d'effectuer.

Une deuxième phase, facultative et dont l'utilisation a été discutée dans [25] est une étape de pondération. Les critères sont présentés par paires (voir figure 4.20) et l'individu sélectionne le critère qui a eu selon lui plus d'impact lors de l'expérience.

Profil du participant

Nous avons pré-sélectionné les participants à l'étude de telle sorte à ce qu'ils soient musiciens. En effet, il existe un risque pour une personne non musicienne que la tâche de segmentation soit trop complexe et nuire aux résultats de l'expérience. Néanmoins, nous n'avons pas mis de contraintes *a fortiori* sur le niveau musical des participants musiciens. Des questionnaires pour évaluer l'appétence, le niveau ou l'affinité musicale existent tels que Le MEQ (Music Experience Questionnaire) qui est un questionnaire auto-administré qui évalue l'expérience musicale globale d'un

Cliquez sur chaque échelle au point qui indique le mieux votre expérience de la tâche.

Exigence Mentale
Faible Elevée
Dans quelle mesure des opérations mentales et perceptives ont-elles été requises (par exemple, penser, décider, calculer, se rappeler, regarder, chercher, etc.)? Ont-elles conduit à une tâche plutôt facile ou difficile, simple ou complexe, abordable ou exigeante?

Exigence Physique
Faible Elevée
Dans quelle mesure des opérations physiques ont été requises (par exemple, pousser, tirer, tourner, contrôler, activer, etc.)? Ont-elles conduit à une tâche plutôt facile ou difficile, lente ou rapide, détendue ou intense, reposante ou pénible?

Exigence Temporelle
Faible Elevée
A combien estimez-vous la pression temporelle que vous avez ressentie, que ce soit à cause du rythme ou de la cadence de la tâche en elle-même ou des éléments de la tâche? La cadence était-elle lente et tranquille ou rapide et frénétique?

Performance
Elevée Faible
Dans quelle mesure pensez-vous avoir réussi à accomplir les objectifs de la tâche fixés par l'expérimentateur? Dans quelle mesure êtes-vous satisfait de votre performance en ayant accompli ces buts?

Effort
Faible Elevé
Quel a été l'effort à fournir (mentalement et physiquement) sur la tâche pour obtenir votre niveau de performance?

Frustration
Faible Elevée
Au cours de la tâche, avez-vous ressenti d'avantage d'insécurité, de découragement, d'irritation et de stress, ou d'avantage de sécurité, gratification, satisfaction et décontraction?

Continue >>

FIGURE 4.19 – Questionnaire NASA-TLX.

Frustration
OR
Exigence Mentale
Au cours de la tâche, avez-vous ressenti d'avantage d'insécurité, de découragement, d'irritation et de stress, ou d'avantage de sécurité, gratification, satisfaction et décontraction?
Dans quelle mesure des opérations mentales et perceptives ont-elles été requises (par exemple, penser, décider, calculer, se rappeler, regarder, chercher, etc.)? Ont-elles conduit à une tâche plutôt facile ou difficile, simple ou complexe, abordable ou exigeante?

FIGURE 4.20 – Extrait de la phase de pondération du questionnaire NASA-TLX.

individu avec des échelles sur l'engagement musical, l'aptitude musicale, l'implication sociale, l'implication émotionnelle, l'implication psychologique et l'implication dans le comportement (P.D. Werneret al., 2006 [162]). Le Goldsmiths Musical Sophistication Index (Gold-MSI) mesure également la *sophistication musicale* d'un individu. Le questionnaire mesure entre autres l'engagement individuel vis-à-vis de la musique, les capacités perceptives, la formation musicale, les capacités en chant ainsi que le facteur émotionnel lié à la musique (D. Müllensiefen et al. en 2014 [107]). Nous considérons que quelques questions uniquement suffisaient dans le cadre de notre expérience. Afin de prévenir si le niveau musical, l'instrument pratiqué ou les problèmes auditifs éventuels peuvent avoir un impact sur le déroulé de l'expérience, nous avons mis en place le questionnaire présenté en figure 4.21.

Il nous semble que le nombre d'année de formation musicale et/ou de pratique instrumentale et le fait de pratiquer encore ou non peut avoir un impact sur la

Questionnaire fin d'évaluation

Âge :

Genre :

- M
 F
 Autre

Combien d'années de formation musicale avez-vous effectué ? Ecrivez 0 si vous n'avez jamais pris des cours de musique.

Quel(s) instrument(s) pratiquez-vous ? (pour les compositeurs qui ne pratiquent pas d'instrument, précisez "composition" et répondez aux questions suivantes par rapport à ça)

Depuis combien de temps jouez-vous un instrument de musique ? (Si vous jouez plusieurs instruments, précisez pour l'instrument pratiqué depuis le plus longtemps).

Pratiquez-vous toujours un instrument ?

- Oui
 Non

Si oui, combien d'heures par semaine ?

Avez-vous des problèmes médicaux liés à votre audition ?

- Oui
 Non

FIGURE 4.21 – Questionnaire d'évaluation du profil.

facilité à analyser et donc segmenter une œuvre musicale de répertoire classique, tandis que l'instrument même peut avoir une influence sur le répertoire que le participant a l'habitude d'écouter ou de pratiquer. Enfin, un problème d'audition peut avoir un impact lors de l'écoute et donc sur la pertinence de la segmentation effectuée ou bien sur la difficulté à écouter.

Interface

Une interface web, développée en Javascript en collaboration avec Asia Auville (étudiante L3 informatique à l'ENSEIRB-Matmeca) dans le cadre de son stage, est présentée aux participants de l'expérience. Celle-ci a été implémentée à partir de

l'interface présentée par M. Poret dans [123]. Les figures 4.18, 4.19, 4.20 et 4.21 sont extraites de l'interface et la figure 4.22 représente l'interface lors de l'écoute d'un morceau musical et de la tâche de segmentation et reproduction de séquences.

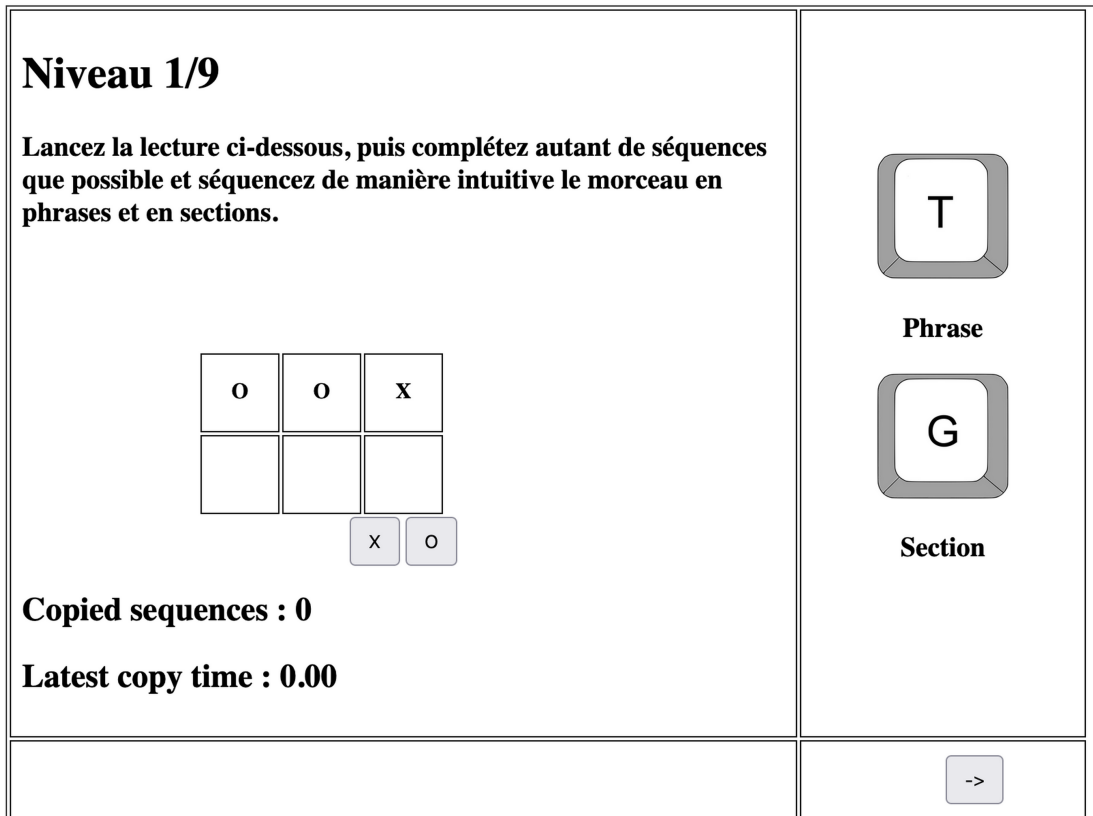


FIGURE 4.22 – Interface de l'expérience cognitive : étape double tâche.

Il est indiqué par ailleurs sur l'interface le nombre de séquences copiées lors de l'écoute du morceau actuel ainsi que le temps de copie du morceau. Ces données sont pour le participant uniquement à titre indicatif.

L'intérêt d'une telle interface est d'une part de proposer l'expérience sur n'importe quelle machine connectée au réseau internet. Par ailleurs, cela permet de récupérer automatiquement les temps de segmentation ainsi que ceux de reproduction de séquences, à l'échelle de la milliseconde.

Mesures

Afin d'évaluer les résultats de l'expérience, trois types de mesures sont proposées. La première concerne la pertinence de la segmentation proposée par le participant. La deuxième concerne la mesure de la charge cognitive, évaluée grâce aux durées de reproduction des séquences tandis que la dernière mesure concerne les valeurs associées au questionnaire NASA-TLX. Ces mesures concernent un individu pour une œuvre donnée.

Mesure de la segmentation

Pour mesurer la segmentation, nous proposons une segmentation de référence. La segmentation de référence a été produite par Elia Bichoy en tant qu'expert, qui a présenté une ou deux segmentations de référence par morceau en fonction des ambiguïtés existantes pour chacune des échelles de segmentation demandées. Cette segmentation a été produite à partir de multiples écoutes et avec références éventuelles à la partition. Dans un second temps, une validation par l'écoute a été effectuée par l'auteur de cette thèse. La segmentation obtenue est alors au même format que les segmentations fournies par les participants à l'expérience. Pour comparer les différentes segmentations, on utilise les mesures de la bibliothèque Python *mir_eval*² [128]. Il s'agit d'une bibliothèque libre qui fournit une implémentation des mesures les plus courantes utilisées pour mesurer la performance des algorithmes de *Music Information Retrieval*.

Nous utilisons spécifiquement deux fonctions du module *segment* qui propose des critères d'évaluation de la segmentation structurelle selon l'annotation des frontières en comparant les frontières estimées des segments aux frontières de référence.

La fonction $detection(s_{ref}, s_{est}, f)$ détermine le nombre de segmentations correctes de telle sorte qu'une segmentation estimée est considérée comme correcte si elle se situe à l'intérieur d'une fenêtre autour d'une segmentation de référence. Elle prend donc en argument un ensemble d'intervalles de référence s_{ref} , un ensemble d'intervalles estimés s_{est} ainsi qu'une taille de fenêtre f en secondes et renvoie un pourcentage de segmentation dans la fenêtre donnée autour de la segmentation de référence.

La fonction $deviation(s_{ref}, s_{est})$ calcule la différence médiane absolue de temps entre une segmentation de référence et la segmentation estimée la plus proche, et inversement. Elle prend donc en argument un ensemble d'intervalles de référence s_{ref} ainsi qu'un ensemble d'intervalles estimés s_{est} et renvoie la différence de temps

2. https://pypi.org/project/mir_eval/

médiane absolue entre la segmentation estimée et la segmentation de référence.

Mesure de la charge cognitive liée à la segmentation

Pour mesurer la charge cognitive liée à la segmentation on mesure le temps pris pour la reproduction des séquences de 'X' et de 'O' successives. L'hypothèse est que plus la tâche de segmentation demande une charge cognitive importante, plus le temps pour reproduire une séquence sera important. On produit ainsi un diagramme représentant le nombre de clics (correspondant au nombre de caractères reproduits) pour 100ms, en fonction du temps. Plus le nombre de clic est élevé, moins la charge cognitive est importante.

Mesure de la charge cognitive avec NASA-TLX

Pour mesurer la charge cognitive avec NASA-TLX, on calcule pour chaque critère la valeur fournie dans l'échelle ramenée à une valeur entre 0 et 100, puis on pondère la valeur obtenue par le nombre de fois où le critère a été choisi parmi les paires de critères proposées sur le nombre de paires totales (soit 15 paires). On somme ensuite les valeurs obtenues pour chacun des critères.

On pose ainsi $C_{c_{tlx}}$ la charge cognitive obtenue en fonction de la valeur des critères v_i et de leur poids respectif $p_i = n_i/15$ avec n_i le nombre de fois où le critère a été prépondérant à un autre :

$$C_{c_{tlx}} = \sum_i p_i * v_i$$

Mesure du niveau musical

Le niveau musical nm est mesuré en faisant la somme des trois critères suivants obtenus à partir du questionnaire sur le profil :

- le nombre d'années de formation musicale pratiquées ;
- le nombre d'années de pratique instrumentale ;
- le nombre d'heures pratiquées par semaine.

le groupe est ainsi divisé en trois catégories de niveaux :

- Catégorie 1 : nm entre 0 et 15. Cela correspond au profil d'un individu qui pratique la musique aujourd'hui mais a commencé il y a peu de temps, ou bien d'un individu qui a appris et pratiqué un nombre d'année moyen (jusque 7 ans) mais ne pratique plus aujourd'hui.
- Catégorie 2 : nm entre 15 et 30. Cela correspond à un étudiant de musique qui a une expérience moyenne à avancée et qui pratique encore aujourd'hui, ou bien à un individu qui a pratiqué un nombre d'années avancées mais ne pratique plus aujourd'hui
- Catégorie 3 : nm supérieur à 30. Il s'agit d'un individu qui a une expérience musicale avancée et pratique activement la musique de manière hebdoma-

daire.

Présentation des résultats

L'expérience a été menée sur un ensemble de 22 individus, 6 femmes et 16 hommes entre 18 et 65 ans répartis en deux groupes distincts de 11 individus. Les différentes pratiques sont la guitare (10 individus), le piano (8), le violon (4), la batterie (4), la basse (2), la composition (2), la flûte baroque, la flûte traversière, la trompette, la contrebasse, le cor d'harmonie, la clarinette, l'harmonica, le chant et le saxophone. On note qu'une partie des participants (8) pratiquent plusieurs instruments et deux individus ont des problèmes d'audition. On a compté 3 individus pour la catégorie de niveau instrumental 1, 8 individus pour la catégorie 2 et 11 individus pour la catégorie 3.

Nous avons dans un premier temps comparé les résultats obtenus sur chacun des deux groupes expérimentaux : nous avons ainsi tracé des diagrammes représentant les résultats NASA-TLX totaux en fonction des morceaux étudiés. Les morceaux sont classés sur l'axe des abscisses en fonction de la difficulté estimée de segmentation de l'œuvre.

Nous avons calculé la moyenne pour chaque morceau ainsi que la moyenne pour chaque niveau de difficulté. Bien que la tendance soit globalement ascendante, nous constatons en réalité un écart type beaucoup trop élevé entre les individus pour pouvoir en déduire des résultats pertinents. Notre hypothèse est que, malgré le choix d'une population musicienne, le corpus d'œuvre choisit était trop complexe et la durée de l'expérience trop longue. Par ailleurs, nous constatons pour l'œuvre numéro 3 un résultat moyen beaucoup plus élevée que les œuvres du même groupe de difficulté, il est donc possible que la difficulté de segmentation n'ait pas été correctement estimée. Nous avons donc cherché à décliner les résultats en regroupant en sous groupes d'individus selon leurs niveau musical ou selon le type d'instrument pratiqué, mais le nombre d'individu ne nous permet pas réellement d'en déduire des résultats généraux. Il serait sans doute plus pertinent d'effectuer alors des études individuelles.

Le temps nous a manqué pour procéder à l'analyse complète de l'ensemble de cette étude. Par ailleurs, nous rencontrons des difficultés concernant la visualisation des résultats dont la dimension est élevée. Nous souhaitons ainsi essayer de retrouver des comportements pertinents via l'analyse statistique des résultats avec l'utilisation du logiciel R. Pour une prochaine expérience, nous pourrions simplifier d'une part les œuvres étudiées, mais également le protocole et sa durée.

Une perspective, et d'autres analyses à effectuer sont bien entendu de comparer les résultats obtenus également avec la segmentation, les coûts effectifs et les phases cognitives mesurées avec MORFOS.

Limites du modèle

On constate que l'expérience, qui dure environ une heure et demi est très longue, ce qui peut avoir un impact important pour la mesure de la charge cognitive. Par ailleurs, les œuvres musicales choisies étaient probablement assez difficiles à segmenter pour la plupart et un corpus plus simple aurait pu être plus adéquat.

Nous ne prenons par ailleurs pas en compte la connaissance externe du participant en dehors de la connaissance du morceau écouté, tel que le genre musical habituellement écouté ou la sensibilité musicale. Sur ce dernier point, il peut être difficile de déterminer parfois si la difficulté de la tâche peut être due à une charge émotionnelle ou à un autre facteur externe tel que l'état de fatigue du participant.

Le contexte même d'une première écoute unique de chacune des œuvres ne rend par ailleurs pas la tâche aisée, et n'est pas vraiment le reflet d'une situation quotidienne où un individu aura tendance à écouter plusieurs fois une œuvre de manière active pour la comprendre et la segmenter. Par ailleurs, bien qu'il s'agisse d'une première écoute, un individu réel n'aura jamais une connaissance externe vide, contrairement à l'implémentation actuelle de MORFOS.

Enfin, bien qu'un effort sur les définitions et les échelles de segmentation aient été fournies, il est difficile de s'y conformer et chaque individu peut être sensible à des échelles différentes de segmentation.

Conclusion

Dans ce chapitre nous avons comparé le comportement de MORFOS avec les fonctions cognitives lors de l'écoute musicale active.

Nous avons ainsi distingué trois phénomènes cognitifs :

- les opérations de classification et de segmentation. Ces opérations sont considérées au niveau cognitif par l'action d'écoute et la représentation en mémoire de l'œuvre, tandis qu'elles sont représentées au niveau du logiciel par l'obtention du diagramme formel multi-échelle ;
- l'analyse du comportement et les phénomènes d'attention. Ces opérations

correspondent aux phénomènes d'acquisition, d'attente, des hypothèses au niveau cognitif, qui sont modélisés au niveau du logiciel par l'étude du comportement algorithmique ;

- le calcul des coûts liés aux phénomènes d'attention. Il s'agit en d'autres termes de la charge cognitive associée à l'écoute musicale, et à la charge computationnelle associée aux opérations de MORFOS.

Nous présentons par ailleurs un bref état de l'art de ces trois différentes notions.

Nous avons ensuite cherché à mesurer la pertinence de la segmentation selon la complexité de Kolmogorov, avec deux méthodes distinctes. Une première proposition est de partir de la définition initiale, à savoir compter le nombre d'opérations nécessaire pour écrire un objet acquis en fonction de son contexte. Pour cela, nous proposons de calculer l'alignement entre les mots définissant les objets acquis dans l'algorithme, et ceux présents dans la mémoire. L'alignement étant une mesure de similarité de notre algorithme, nous proposons d'étendre la notion non plus au nombre d'opérations mais à un calcul de similarité entre l'objet acquis et son contexte. Cela nous permet d'étendre le calcul de la complexité à toutes les similarités utilisées pour le calcul des diagrammes formels multi-échelles.

Dans un second temps, nous proposons de classer le comportement de l'algorithme selon quatre phases cognitives telles qu'elles sont décrites et présentées par J.M Chouvel dans [32]. Chaque phase correspond à une trajectoire selon le résultat *oui* ou *non* associé aux deux tests de classification et de segmentation. Ces phases cognitives nous informent ainsi sur les processus de mémorisation et d'attention lors de l'acquisition d'un nouvel objet en fonction de son contexte. Après avoir présenté notre modélisation de l'automatisation du calcul des phases cognitives, nous proposons de mesurer la complexité, définie dans la section précédente, associée aux différentes phases cognitives. Ainsi, nous pourrions étudier quels sont les comportements qui permettent de minimiser, maximiser ou varier la complexité au cours de l'écoute. Afin de présenter une analyse plus fine de ces comportements, nous pourrions également mesurer les phases secondaires, qui correspondent aux 16 enchaînements possibles des 4 phases cognitives.

Dans une troisième section, nous proposons de comparer l'implémentation de MORFOS avec les modèles cognitifs existants : nous nous concentrons sur un modèle d'écoute par EEG proposé par S. Koelsch et W. A. Siebel [79]. Chacune des étapes du modèle de S. Koelsch et W.A. Siebel est ainsi comparé avec les différents processus de l'algorithme, soulignant ainsi l'implication des potentiels MMN, ERAN, P300, P600, N400 et LPC et CPS dans les processus de segmentation de l'écoute musicale et plus spécifiquement des potentiels MMN, ERAN et CPS pour la détection de la fin de la phrase musicale qui pourraient être impliqués dans la

modélisation de MORFOS.

Par ailleurs, nous avons mis en place une expérience double tâche afin de déterminer le comportement d'individus lors d'une tâche de segmentation pendant l'écoute active. Les individus indiquent les segmentations des fin de phrase et de fin de section telles qu'elles sont définies dans l'expérience, en effectuant une deuxième tâche de reproduction de séquence de caractères. Nous cherchons ainsi d'une part à mesurer la pertinence de la segmentation des individus, et d'autre part la charge cognitive associée à la tâche effectuée, selon différents niveau de difficulté pour les œuvres à segmenter. Les résultats obtenus sont d'un premier abord difficilement exploitable, bien qu'il nous reste à effectuer des études plus pointues avec l'utilisation de logiciels statistiques comme R³. Il reste ensuite à comparer les comportements étudiés avec le comportement du logiciel.

3. <https://www.r-project.org/>

Chapitre 5

Visualisations des diagrammes formels multi-échelle

La visualisation d'informations augmente la performance des activités cognitives pour l'acquisition et la compréhension des connaissances. On cherche ainsi à améliorer la compréhension du phénomène musical au delà de sa notation et de sa représentation et donc à présenter une visualisation appropriée pour l'analyse musicale.

Au delà d'une traduction de l'idée musicale, on souhaite proposer une représentation qui permet de mettre en lumière les processus musicaux et plus encore l'implication cognitive de ces processus musicaux à travers leur développement temporel.

L'information musicale dans les diagrammes formels multi-échelles se trouve à différentes échelles temporelles. Comment peut-on alors visualiser une telle structure efficacement malgré la quantité d'information à chaque niveau structurel ? Quelle représentation permet de représenter efficacement tous les niveaux ainsi que les imbrications entre chacun des niveaux ? Souhaite-t-on toujours représenter tous les niveaux à la fois ou peut-on représenter chaque échelle de manière indépendante et pertinente ?

Par ailleurs, la musique par définition est une donnée temporelle : elle fait sens par l'ordre dans lequel les données qui la constituent apparaissent. La représentation des diagrammes formels hiérarchisés tient compte de l'ordre d'apparition des éléments. En étendant la dimension temporelle sur les trois dimensions de la durée des objets musicaux, du déroulement de l'œuvre et de la première date d'appari-

tion des matériaux, on peut espérer alors visualiser l'organisation temporelle et l'organisation structurelle (délinéarisation du temps) entre les éléments. On peut par ailleurs chercher à afficher la construction de la visualisation en même temps que la diffusion de l'œuvre musicale.

On propose trois représentations différentes :

La première visualisation envisagée, est la superposition des diagrammes formels à deux dimensions pour tous les niveaux de structure, mais elle ne rend pas bien compte visuellement des imbrications structurelles.

Une seconde proposition de visualisation superpose en trois dimensions les différents diagrammes précédents en donnant aux éléments musicaux une taille proportionnelle à leur durée. Sur une telle représentation on peut visualiser la forme et la structure musicales comme deux projections sur deux plans partageant l'axe du temps.

Pour faciliter la lecture des liens structurels, une troisième représentation est proposée, qui synthétise les liens en donnant aux éléments musicaux l'apparence de sphères de même taille.

Ces visualisations peuvent être proposées soit comme construction en temps réel soit comme lecture dynamique dans un diagramme déjà établi. Du fait de leur cohérence avec des procédures cognitives, l'ensemble de ces visualisations est particulièrement utile pour mieux comprendre des phénomènes musicaux particulièrement complexes.

Par ailleurs, nous pouvons automatiser ces visualisations grâce au fait que MORFOS est lui même une automatisation de l'Algorithme Cognitif.

5.1 État de l'art pour la visualisation musicale

Bien que la visualisation musicale soit un domaine de recherche largement étendu, nous précisons que notre travail ne porte pas sur la visualisation du contenu audio de l'œuvre pour améliorer l'expérience de l'écoute musicale, mais sur la visualisation d'une représentation musicale (dans ce cas, le diagramme formel) qui met en évidence la construction interne et le développement temporel des œuvres musicales dans le but de permettre une meilleure analyse musicologique. Ainsi, les

représentations musicales et leurs visualisations nous permettent non seulement de mieux comprendre le phénomène musical mais elles nous aident également à représenter et à concevoir de nouvelles formes musicales. Nous présentons dans cette partie des notations et des représentations symboliques et audio qui permettent la visualisation de la forme musicale.

5.1.1 Représentations manuscrites

Une des plus anciennes représentations de l'œuvre musicale est la partition. La partition est à la fois notation et visualisation de l'intention du compositeur et représente la hauteur, la durée, l'intensité ainsi que le timbre. Seulement, celle-ci indique les caractéristiques des éléments qui constituent l'œuvre mais ne représente pas la relation entre ceux-ci, en fonction du temps. Nicolas Ruwet propose une première analyse paradigmatique de l'œuvre musicale et introduit au monde de la musique les notions de linguistiques [134]. Lerdhal et Jackendhoff propose dans *a Generative Theory of Tonal Music* [88] une représentation arborescente de la grammaire musicale. Seulement la musique ne peut pas se résoudre à la linguistique d'une part, et ces différentes représentations sont restreintes aux œuvres dont il existe des notations écrites telles que la musique classique.

On observe alors des premières tentatives de description des musiques électro-acoustiques avec la création de la musique concrète et la définition des objets musicaux de P. Schaeffer [141]. Jean-Louis Di Santo s'inspire par ailleurs des unités sémiotiques temporelles [43] pour proposer à travers l'Acousmoscribe [137], un éditeur de musiques acousmatiques, un ensemble de notations pour ces musiques.

On constate cependant une difficulté à extraire avec des notations manuelles certaines caractéristiques propres aux données du signal et donc à représenter puis visualiser ces données.

5.1.2 Vers une représentation de la musique audio

Avec la Théorie de l'information présentée par Claude Shannon [146] et l'analyse spectrale effectuée avec la transformée de Fourier [22], on accède à la forme d'onde et les spectres sonores, qui permettent de représenter respectivement l'évolution temporelle de l'amplitude de l'onde sonore ainsi que les composantes fréquentielles de l'onde sonore. Ces analyses sont utilisées pour proposer de nouvelles

visualisations telles que le sonagramme qui représente l'amplitude pour chaque fréquence en fonction du temps.

A partir du sonagramme, le GRM a proposé l'Acousmographe [57], un outil qui permet de représenter et d'annoter les musiques électroacoustiques. Le logiciel E-Analysis, développé par P. Couprie [38] propose également des représentations graphiques des analyses de musiques électroacoustiques, avec de potentielles interactions dynamiques.

D'autres logiciels, comme PRAAT¹ [19] ou Spear² [75], proposent des représentations des signaux vocaux en découpage des fréquences et de leur intensité dans le temps. Ces différentes représentations cependant ne mettent pas en avant l'aspect structurel de l'audio. Les matrices d'auto-similarité [53] permettent de palier cela en représentant la similarité du morceau de musique au cours du temps avec lui-même. D'autres représentations permettent de mettre en valeur la structure de l'œuvre musicale, comme le Tonnetz du modèle Système & Contraste de V. Gillot [58]. Dans cette représentation, les relations entre les éléments constitutifs de l'œuvre sont donnés à priori et la topologie n'est pas construite en fonction de l'œuvre fournie. Par ailleurs, la représentation temporelle de la musique est ici tout à fait délinéarisée. L. Bigo propose également une visualisation délinéarisée du temps dans l'œuvre [14] à travers une présentation des notes et accords les plus courants dans l'œuvre musicale sur un empan temporel donné et un nombre de notes principales fixés. Il ne s'agit pas d'une visualisation dynamique au court du temps, mais cela permet de comparer facilement deux extraits différents.

A. Marmoret quant à lui propose une représentation des classes d'éléments dans l'œuvre musicale à travers le temps. Il utilise la *Nonnegative Tucker Decomposition* pour représenter la structure des œuvres musicales [94]. Il fournit un nombre fixé de classes d'éléments à retrouver dans l'œuvre et obtient automatiquement avec des réseaux de neurones une représentation.

Ces visualisations de la donnée musicale permettent de représenter les musiques non écrites et les musiques composées à partir du son d'une part mais également pour certaines de mettre en avant leur structure. Mais celles-ci nécessitent de fixer en amont l'espace topologique qui constitue l'œuvre, là on l'on propose avec les diagrammes formels une visualisation qui crée l'espace constitutif de l'œuvre au fur et à mesure de l'acquisition de la donnée musicale.

1. <https://www.fon.hum.uva.nl/praat/>

2. <https://www.klingbeil.com/spear/>

Ainsi, le diagramme formel s'inspire de l'analyse paradigmatique de François Ruwet, selon une topologie hiérarchisée dans la continuité des travaux de Lerdhal et Jackendoff. Néanmoins, nous étendons cette représentation à l'ensemble de la musique électro-acoustique où comme l'objet musical de P.Schaeffer, l'objet du diagramme formel correspond à une unité paradigmatique sonore quelconque. Notre représentation est calculée à partir d'une analyse du signal audio obtenue à partir de techniques de traitement du signal et des visualisations qui en ont découlé telles que l'Acousmographe et E-Analysis. Si des visualisations des données musicales existent pour représenter les fréquences telles que PRAAT, la structure telle que les matrices d'auto-similarité, ou selon un espace topologique donné comme le Tonnetz de V. Gillot, on cherche dans notre cas à proposer une visualisation de la forme, de la structure et de l'organisation tels que définis au chapitre 2. On souhaite que cette visualisation soit dynamique, c'est-à-dire que le diagramme formel multi-échelle soit construit en tenant compte uniquement des informations analysées jusqu'à l'instant t .

5.2 Les trois visualisations

Nous proposons trois visualisations distinctes pour étudier les diagrammes formels. La première, utilisée jusque là, est en deux dimensions et permet une lisibilité simple des diagrammes de chaque niveau temporel. Les deux autres représentations en trois dimensions permettent de mettre en avant les liens entre les objets des différents niveaux.

L'objectif est ainsi de visualiser au mieux les plans de la forme (axes matériaux/temps), de la structure (axes niveaux/temps) et de l'organisation (axes matériaux/niveaux), présentés au chapitre 2.

5.2.1 Visualisation 2D

La première représentation que nous présentons est la représentation en deux dimensions et est celle qui a été utilisée jusqu'à aujourd'hui et présentée jusqu'ici dans ce manuscrit. La figure 5.1 représente une succession de diagrammes formels selon cette représentation.

Dans cette visualisation, l'axe du temps est décrit sans ambiguïté par une unité temporelle (la seconde ou l'unité minimale d'un objet par défaut), et l'axe

des matériaux est décrit par le label du matériau inscrit. L'axe des niveaux n'existe pas et correspond à la succession des différents diagrammes alignés de haut en bas.

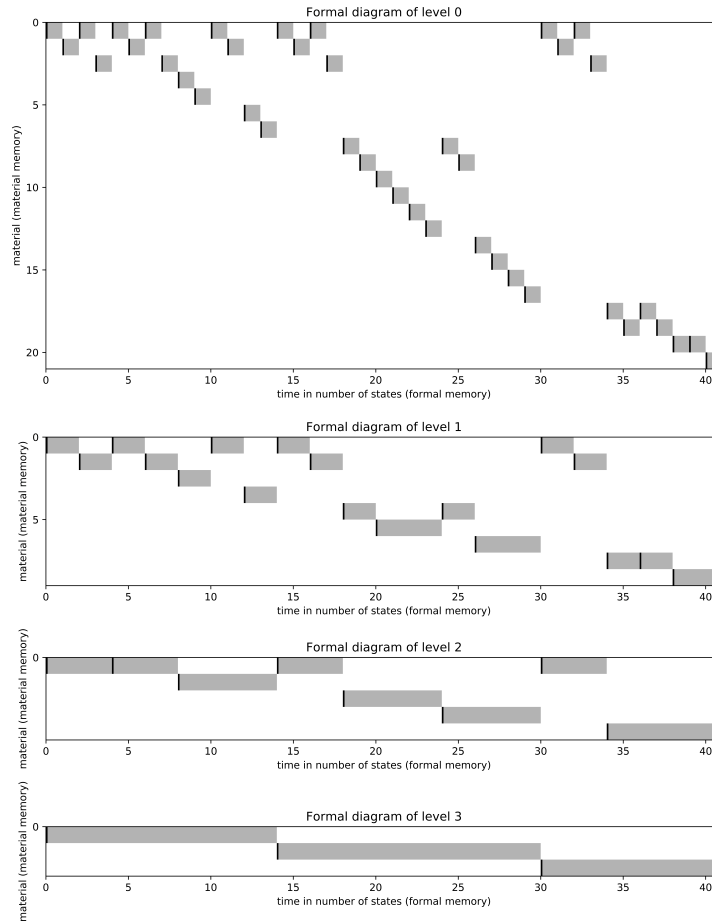


FIGURE 5.1 – Visualisation 2D du diagramme formel multi-échelle obtenu automatiquement à partir du *Rondo K545* de W.A. Mozart.

Définition des diagrammes formels 2D

Les diagrammes formels sont superposés et alignés selon l'axe temporel. Ainsi, les objets alignés sur le même segment temporel appartiennent à la même réalité sonore. Les objets de temporalité plus petite constituent ceux de temporalité plus grande sur le même segment. Pour tous les niveaux, les premières millisecondes de chaque objet sont affichées en noir afin de distinguer deux objets successifs.

Chaque diagramme a donc son propre axe des matériaux, qui correspond aux

matériaux de la même œuvre musicale mais pour une durée de temps donnée. La représentation de l'axe des matériaux permet une compression de l'information. En effet, d'une part, l'information récurrente est condensée sur une unique ligne. D'autre part, tout l'espace nécessaire pour représenter l'œuvre musicale est utilisé et augmenté lorsqu'un nouveau matériau apparaît.

Inspiré de la Gestalt Theory, les motifs musicaux deviennent des motifs graphiques, et également les formes sont reproduites indépendamment des contenus des matériaux. Un musicologue peut ainsi détecter les reproductions de forme à une même échelle dans un morceau donné avec différents matériaux, ou même une reproduction d'une même forme à différentes échelles au sein d'un morceau. La ligne du front de découverte qui relie les derniers matériaux découverts peut ainsi être distinguée : en fonction de ses pentes, elle donne une information sur la quantité d'information découverte au cours du morceau.

Construction des diagrammes formels 2D

La visualisation du diagramme formel 2D est représentée par une matrice $M = (m_{i,j})_{0 \leq i \leq n, 0 \leq j \leq t}$ de taille $n \times t$ où n est le nombre de matériaux entendus jusque là et t le nombre d'unités de temps (égal à un multiple du nombre d'objets entendus au niveau 0) depuis le début de l'analyse. Pour chaque paire d'indices (i, j) :

- si le matériau i est absent à l'instant j , alors $m_{i,j} = 0$;
- sinon, $m_{i,j}$ prend une valeur numérique telle que, par exemple, la dynamique de l'objet associé ou une valeur par défaut fixée à 0, 5.

Sur cette visualisation, l'axe temporel est gradué de la même manière pour tous les niveaux des diagrammes formels d'une analyse donnée. Pour un diagramme formel de niveau l donné, chaque objet entendu est donc associé à un segment temporel $[j : j+k[$ où j et $j+k$ correspondent aux bornes respectivement inférieures et supérieures des segments temporels des objets de niveau inférieur qui débutent et terminent l'objet actuel. Chaque entier inclus dans le segment temporel correspond à un indice temporel dans la matrice, associé au matériau du nouvel objet.

Ainsi, une *coloration* associée à l'objet $x_{ind}^{(l)}$ est définie par la coloration de tous les indices qui constituent l'objet, obtenus récursivement par construction des diagrammes inférieurs.

- La mise à jour du diagramme formel se déroule alors en trois étapes distinctes :
- ajout d'une ligne si nécessaire ;
 - détermination des indices ;

— coloration.

L'ajout d'une ligne a lieu si le nombre de matériaux dans la structure de données du Diagramme Formel est supérieur à la matrice de visualisation M .

On cherche ensuite à définir les indices dans la matrice correspondant à la coloration du nouvel objet. La coloration se faisant en temps réel, la borne inférieure b_{inf} correspond à l'indice de la taille de la matrice. La borne supérieure b_{sup} quant à elle correspond à la somme de la borne inférieure avec le nombre d'états au niveau 0 de valeur maximale dans la table des liens accédée récursivement depuis la table des liens de niveau $l - 1$.

Pour colorer dans la matrice, k colonnes sont donc ajoutées telles que $k = b_{sup} - b_{inf}$

$$M_{n,t} \xrightarrow[\text{col}, x_t]{\text{concat}} M_{n,t+k}$$

La phase de coloration consiste ensuite à colorer les k éléments des indices temporels t à $t + k$ et de rang du matériau $rg(x_t)$ où $rg(x_t)$ est la fonction qui renvoie le rang du matériau associé à x_t (voir chapitre 2).

Par ailleurs, chaque objet de niveau 0 est lui même subdivisé en f sous-unités. L'algorithme multiplie donc la valeur associée à l'indice des bornes à tous les niveaux par ce facteur f de manière à donner une épaisseur aux matériaux de niveau 0. En effet, cela permet d'afficher le premier élément concerné avec une valeur numérique constante, différente des éléments suivants et ainsi délimiter les frontières des matériaux.

Reprenons l'exemple du Mozart, cette fois-ci lors de l'acquisition du 2e objet de niveau 1, c'est-à-dire avant classification du deuxième facteur de la représentation segmentée de niveau 0. On choisit dans un premier temps $f = 1$.

Avant classification au niveau 1, on a :

$$\begin{aligned} r^{(0)} &= abaca \\ s^{(0)} &= (ab)(ac) \\ r^{(1)} &= a \end{aligned}$$

$$df^{(0)} = \begin{matrix} a \\ b \\ c \end{matrix} \begin{pmatrix} v & 0 & v & 0 & v \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & 0 & v & 0 \end{pmatrix} \text{ avec } a, b, c \in A^{(0)} \text{ et } df^{(1)} = a \begin{pmatrix} v & v \end{pmatrix} \text{ avec } a \in A^{(1)}$$

Avec v la valeur par défaut. Pour $f = 1$, on remarque que la structure de données du diagramme formel de niveau 0 contient la même information que la structure permettant la visualisation au niveau 0 du diagramme formel. Au niveau

1, $x_0^{(1)}$ correspond aux deux objets $x_0^{(0)}x_1^{(0)} = ab$ défini sur le segment temporel $[0 : 2[$ et donc sur les instants d'indices 0 et 1.

Après classification de (ac) et donc acquisition du 2e objet au niveau 1, on a :

$$r^{(0)} = abaca$$

$$s^{(0)} = (ab)(ac)$$

$$r^{(1)} = ab$$

1) Au niveau 1, il y a création d'une nouvelle ligne dans la structure du diagramme formel : une nouvelle ligne est concaténée et correspond au nouveau matériau b dans la visualisation du diagramme formel de niveau 1 :

$$df^{(1)} = \begin{pmatrix} v & v \\ 0 & 0 \end{pmatrix}$$

2) Pour déterminer les indices, $t = 2$, donc $b_{inf} = 3$.

Par ailleurs, la table des liens de niveau 0 est $Tl^{(0)} = (1, 1, 2, 2)$ En parcourant le tableau, on constate que l'on a deux états de valeur 2, d'où $b_{sup} = 5$.

Deux colonnes de longueurs égales au nombre de matériaux sont concaténées à la matrice :

$$df^{(1)} = \begin{pmatrix} v & v & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

3) Les indices de la matrice contenus entre les deux bornes calculées au rang du matériau acquis sont enfin colorés :

$$df^{(1)} = \begin{pmatrix} v & v & 0 & 0 \\ 0 & 0 & v & v \end{pmatrix}$$

Prenons maintenant $f = 3$.

Cela signifie que chaque objet du niveau 0 est subdivisé en $f = 3$ unités de temps. La première unité est colorée avec s une couleur désignant la frontière, et l'intérieur du bloc est coloré avec v la valeur par défaut.

On a alors avant classification :

$$df^{(0)} = \begin{pmatrix} s & v & v & 0 & 0 & 0 & s & v & v & 0 & 0 & 0 & s & v & v \\ 0 & 0 & 0 & s & v & v & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s & v & v & 0 & 0 & 0 \end{pmatrix}$$

$$df^{(1)} = (s \ v \ v \ v \ v \ v)$$

1) il y a toujours création d'un nouveau matériau, ce qui implique de concaténer une nouvelle ligne à la matrice :

$$df^{(1)} = \begin{pmatrix} s & v & v & v & v & v \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2) Par ailleurs $t = 6$, donc $b_{inf} = 7$ et la table des liens de niveau 0 est $TI^{(0)} = (1, 1, 2, 2)$ Il y a toujours deux états de valeur 2, donc $b_{sup} = 7 + 2 * f = 13$. Il y a donc $13 - 7 = 6$ colonnes de longueur égale au nombre de matériaux concaténées à la matrice :

$$df^{(1)} = \begin{pmatrix} s & v & v & v & v & v & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

3) Enfin, tous les indices $(m_{i,j})$ pour i entre b_{inf} et b_{sup} et $j = 1$ sont colorés.

$$df^{(1)} = \begin{pmatrix} s & v & v & v & v & v & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s & v & v & v & v & v \end{pmatrix}$$

En fonction des différentes versions du logiciel, nous notons que la visualisation des diagrammes formels en 2D a été modifiée : dans la version 2.0, tous les diagrammes formels, quelque soit leur niveau, représentent les objets avec un liseré noir qui initie l'objet et un aplat de gris choisi par défaut. Cette couleur par défaut pourrait être à l'avenir modifiée par une information supplémentaire à afficher à l'utilisateur et consistante à tous les niveaux. Dans la version 1, nous proposons une visualisation distincte pour le niveau 0 et les niveaux supérieurs. La visualisation de niveau 0 dans une version 1.0 décrit tous les objets en noir, ce qui rend la distinction des objets successifs de même matériau difficile. A partir de la version 1.1, chaque objet est décrit par un niveau de gris correspondant au volume calculé pour l'objet. Les niveaux supérieurs ont également différents affichages. Une première proposition, dans la version 1.1, est d'alterner la couleur des objets de manière arbitraire entre quatre couleurs afin de distinguer les objets successifs. La représentation du diagramme formel étant temporelle, l'argument pour une telle proposition est que le dessin d'une frontière entre deux objets définit une réalité temporelle qui n'existe pas, ou bien dont l'épaisseur réelle est floue. Néanmoins, cette représentation a donné lieu à des ambiguïtés auprès des utilisateurs et nous nous sommes finalement penchés dans la version 1.2 sur la même représentation que la version 2.0 de MORFOS.

Malgré l'alignement des diagrammes au niveau de l'axe temporel, cette représentation ne permet pas toujours de faire les liens aisément entre les matériaux de niveaux supérieurs et ceux de niveaux inférieurs. C'est pourquoi nous proposons des représentations en trois dimensions qui permettraient de superposer les niveaux hiérarchiques.

5.2.2 Visualisation 3D en pavés

Une première proposition de représentation en trois dimensions provient de l'idée de superposer les différents diagrammes hiérarchisés de la représentation en deux dimensions. On représente les objets constitutifs de l'œuvre par des pavés.

Dans cette visualisation, l'axe du temps est toujours décrit par une unité temporelle et l'axe des niveaux est décrit par une échelle linéaire des niveaux des diagrammes formels. L'axe des matériaux quand à lui n'est pas uniforme en fonction des niveaux. En effet, bien que chaque matériau soit associé sur l'axe par son ordre d'apparition et décrit par son label, le même point sur l'axe des matériaux ne correspond pas au même rang de matériau (selon la définition présentée au chapitre 2) en fonction des niveaux.

Les trois dimensions suivantes sont définies pour les pavés :

- la longueur est mesurée sur l'axe du temps et correspond à la durée de l'objet ;
- la hauteur est mesurée sur l'axe des matériaux et donne une information sur le niveau de l'analyse. Elle est proportionnelle au niveau n du diagramme et est définie par $h(n) = (2^n) * h(0)$ avec $h(0)$ la hauteur du niveau 0, constante et différente de 0 ;
- la profondeur est mesurée sur l'axe de la hiérarchie et est constante.

La figure 5.2 est une représentation d'un ensemble de diagrammes formels hiérarchisés pour le *Rondo* de la *Sonate n°16 K545* de Wolfgang Amadeus Mozart. Cette représentation a été obtenue manuellement à l'aide de l'outil Blender [34]. L'objectif à terme est d'obtenir la représentation automatiquement à partir des objets calculés avec le logiciel implémenté. Les pavés sont de différentes couleurs en fonction des niveaux auxquels ils appartiennent. Ainsi, dans cet exemple, les pavés de couleur bleue appartiennent au niveau 0, les pavés de couleur verte au niveau 1, jaune au niveau 2 et orange au niveau 3.

Afin de pouvoir voir l'ensemble des niveaux de diagramme en simultané, les niveaux les plus petits sont affichés devant, et les niveaux les plus grands derrière, ainsi, les objets de niveaux inférieurs qui constituent les objets de niveaux supérieurs sont physiquement inclus dans les blocs plus gros. Par ailleurs on propose de représenter chaque niveau par une couleur différente afin de facilement les distinguer.

Cependant, la hauteur des diagrammes, choisie arbitrairement proportionnelle au niveau hiérarchique correspondant, induit un décalage de hauteur à un moment

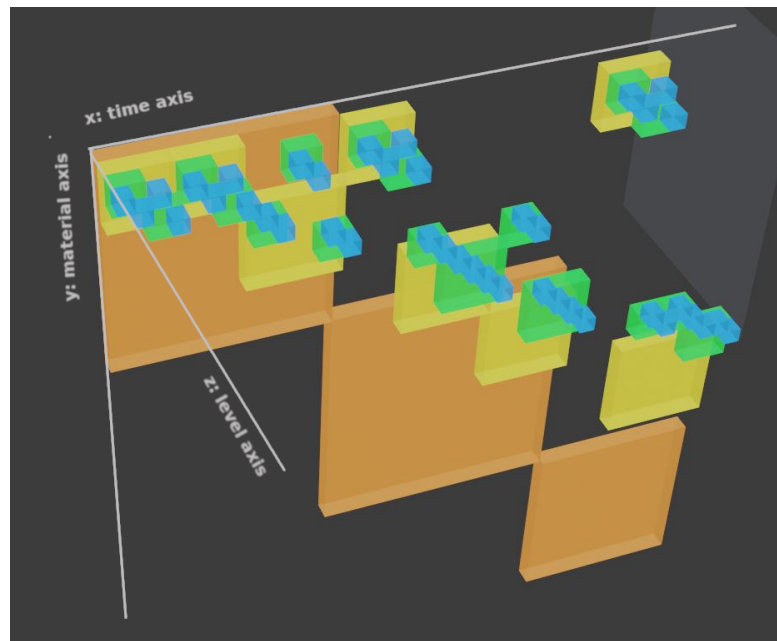


FIGURE 5.2 – Visualisation 3D en pavés du diagramme formel multi-échelle représenter via Blender d’un extrait du *Rondo K545* de W.A. Mozart.

donné entre les objets d’un niveau donné et les objets équivalent de niveau inférieurs, qui ne se trouvent plus alignés.

On propose alors une deuxième représentation en trois dimensions qui permet de pallier cela en alignant sur l’axe des matériaux la hauteur des objets aux différents niveaux.

5.2.3 Visualisation 3D sphère

Pour cette deuxième représentation 3D, les objets sont représentés par des sphères de diamètre unique quelque soit le niveau du diagramme et la durée de l’objet. Leur centre est disposé sur l’axe du temps à l’instant t de la segmentation de l’objet. Cela permet d’éviter la confusion qui existait sur les deux représentations précédentes entre deux objets adjacents. Par ailleurs, il n’y a plus de problème de hauteur pour les matériaux qui apparaissent en même temps à des niveaux différents. La figure 5.3 est une représentation en trois dimensions avec des sphères du *Rondo K.545* de la même sonate de Mozart vue précédemment.

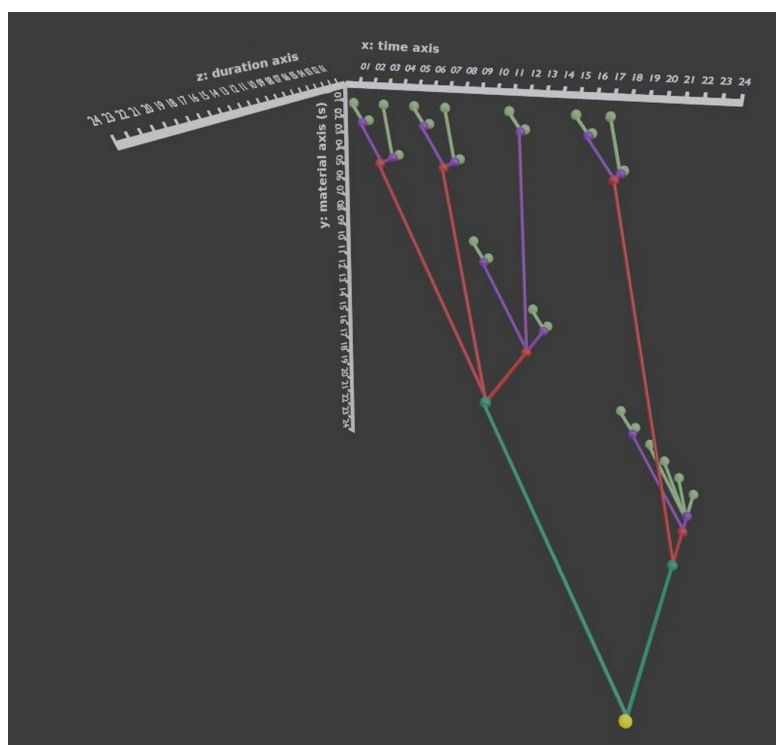


FIGURE 5.3 – Visualisation 3D en sphères du diagramme formel multi-échelle représenté via Blender d'un extrait du *Rondo K545* de W.A. Mozart.

Présentation de la visualisation

Dans cette visualisation, chaque axe est une représentation du temps : l'axe du temps correspond au déroulement temporel de l'œuvre, l'axe des matériaux correspond à la date de première apparition des matériaux et l'axe des niveaux correspond à la durée des objets. Ainsi, la représentation est de la taille d'un cube dont chaque arrête est de la longueur de la durée du morceau, la taille de la représentation finale est donc connue avant l'analyse. Cependant le fait d'afficher la date d'apparition du matériau et non pas une numérotation du matériau fait qu'il s'agit d'une représentation moins compressée, il faudrait supprimer les temporalités inutiles pour recompresser cet axe temporel.

On définit les trois dimensions suivantes pour les sphères :

- la longueur est mesurée sur l'axe du temps et correspond à la durée de l'objet sonore. Il s'agit donc de la distance entre la projection sur un même plan parallèle au plan de la forme du centre de l'objet actuel et du centre de l'objet qui le précède ;

- la hauteur est mesurée sur l'axe des matériaux. Elle est constante et égale au rayon de la sphère représentant l'objet ;
- la profondeur est mesurée sur l'axe de la hiérarchie et donne une information sur l'écart de l'objet actuel au premier matériau du niveau actuel. Il s'agit donc de la distance entre la projection sur un même plan parallèle au plan de l'organisation du centre de l'objet actuel et du centre du premier objet de même niveau.

Les objets étant classés par niveaux dans la structure de données, il est toujours possible de distinguer les niveaux en les représentant de couleur différente. Cependant, les objets pour un même niveau ne sont plus disposés de manière strictement alignée sur l'axe des niveaux mais selon leur durée. Cela permet d'assurer une continuité entre les différents niveaux. Par ailleurs, les liens de filiations entre les objets d'un niveau donné et les objets qu'ils constituent au niveau supérieur sont représentés par des liens physiques connectant les deux objets. Cela forme un arbre, où un objet fils n'appartient qu'à un seul objet parent de niveau supérieur et est donc relié à une unique sphère, mais un objet de niveau supérieur possède plusieurs matériaux enfants qui le constituent et est donc relié à plusieurs sphères.

Avec cette représentation, lorsque l'axe des matériaux n'est pas compressé la lecture du front de découverte change : il faut regarder la diagonale. Plus il y a d'éléments successifs sur la diagonale, plus il y a découverte de nouveaux éléments.

Visualisation et analyse des 3 plans

Cette superposition d'objets permet une meilleure lecture des diagrammes en trois dimensions comme le montre la figure 5.4, selon les trois les trois de la forme, de l'organisation et de la structure (chapitre 2).

Analyse du plan de la forme

Dans l'exemple de Mozart, le refrain revient de manière récurrente (matériaux apparaissant pour la première fois aux moments 01, 02 et 04), alterné avec des développements que nous appelons phases de découverte. Ces phases permettent à l'auditeur d'acquérir une quantité croissante d'informations et d'éviter l'ennui, tandis que le retour régulier à des matériaux connus peut l'aider à ne pas être perdu sous un flot d'informations nouvelles.

Analyse du plan de la structure

Il y a une subdivision quasi systématique des objets de niveau supérieur en deux objets de niveau inférieur. Nous sommes donc sur une structure binaire très conven-

5. Visualisations des diagrammes formels multi-échelle

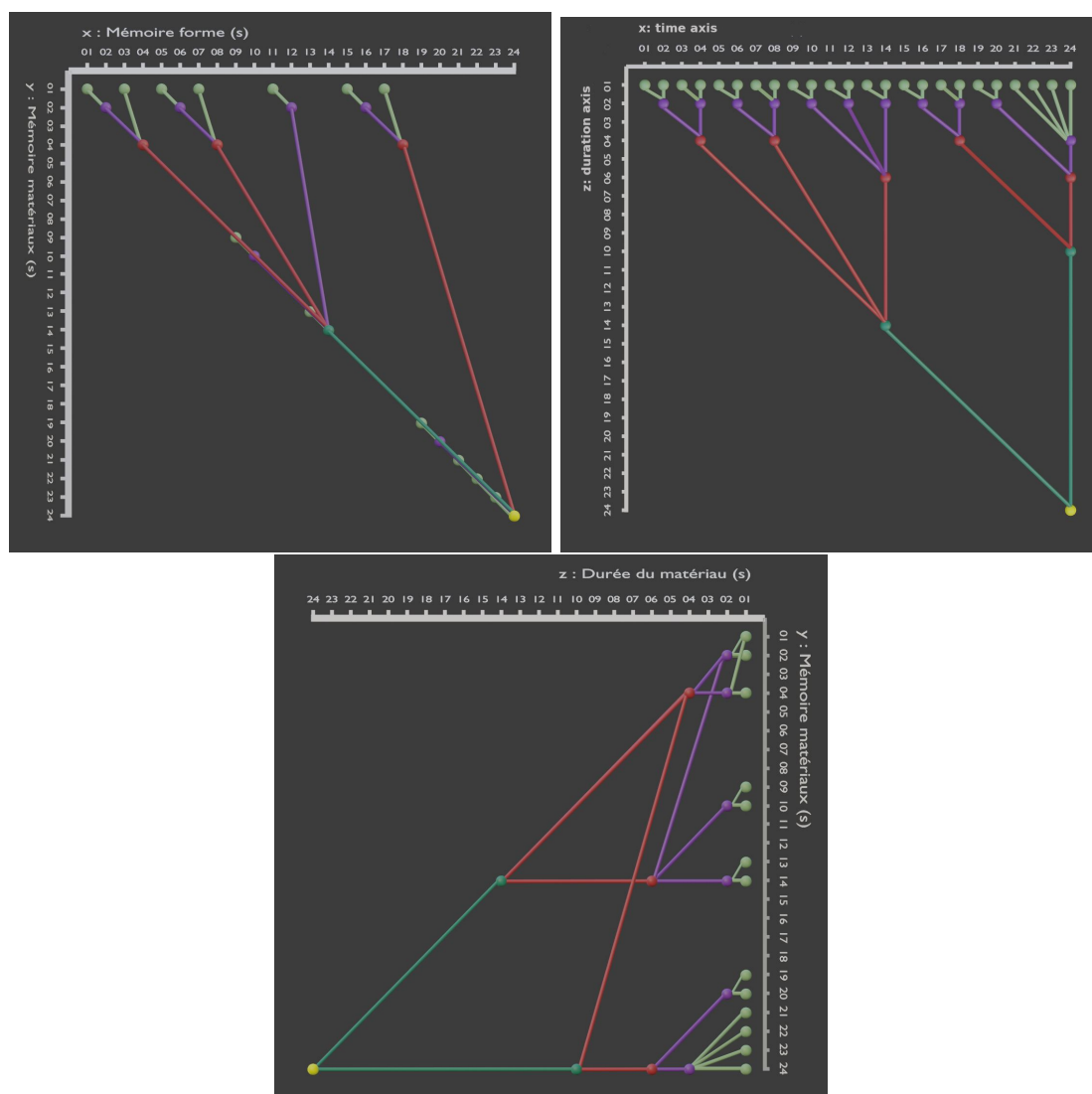


FIGURE 5.4 – Visualisation 3D en sphères du diagramme formel multi-échelle représenté via Blender d’un extrait du *Rondo K545* de W.A. Mozart. De haut en bas : le plan de la forme, le plan de la structure, le plan de l’organisation.

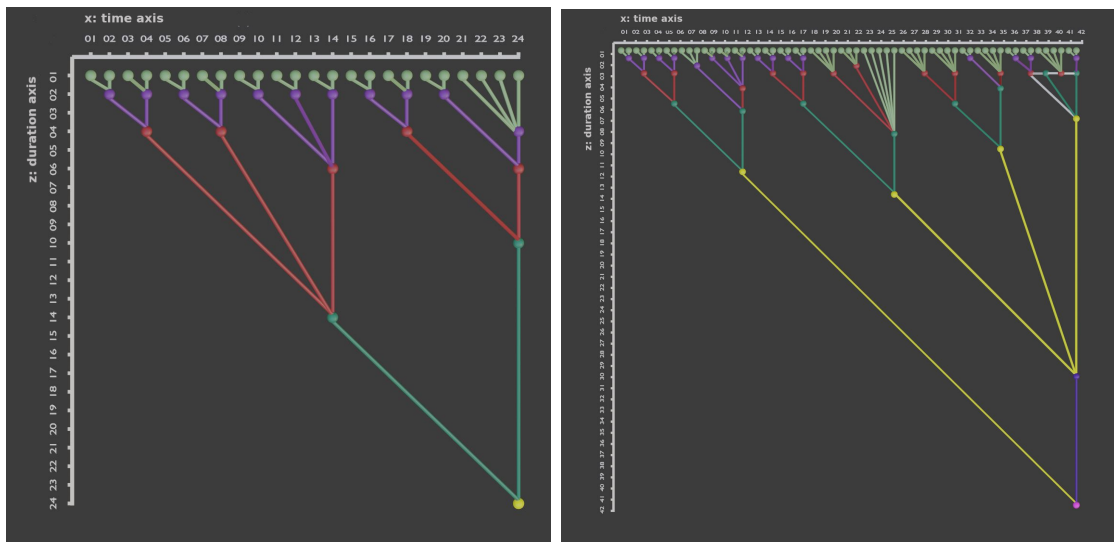


FIGURE 5.5 – Visualisation 3D en sphères du plan de la structure du diagramme formel multi-échelle représenté via Blender d'un extrait du *Rondo K545* de W.A. Mozart (gauche) et de *l'Hommage à Rameau* de Debussy (droite).

tionnelle, avec parfois des subdivisions en trois ou quatre éléments permettant d'apporter un peu de variation et un effet de surprise.

Analyse du plan de l'organisation

Trois matériaux sont utilisés à plusieurs reprises pour constituer des matériaux de niveau supérieur, mais l'organisation des matériaux est globalement sérielle, les matériaux apparaissant successivement pour la première fois constituant un nouveau matériau de niveau supérieur.

Nous présentons une étude comparative de la pièce de Mozart avec *l'Hommage à Rameau* de Claude Debussy, d'un point de vue structurel.

La figure 5.5 montre un plan de la structure du *Rondo K545* de Mozart et de *l'Hommage à Rameau* de Claude Debussy. Contrairement à la visualisation du plan de la structure de Mozart, la visualisation de ce plan chez Debussy montre une ambiguïté entre les niveaux de structures. En effet, pour rappel, les couleurs des sphères correspondent à un niveau donné. La couleur vert clair correspond au niveau 0, la plus petite échelle de temps, le violet correspond au niveau 1, le rouge au niveau 2, le vert foncé au niveau 3, et ainsi de suite. Dans l'œuvre de Mozart, les sphères de même couleur sont globalement alignées sur l'axe de la durée et les niveaux se succèdent sans ambiguïté sur cet axe. En revanche, on remarque qu'il y

a une ambiguïté sur la succession des niveaux pour la pièce de Debussy. En effet, par exemple, le niveau 1 (en rouge) n'existe pas pendant les moments 22 à 25 : cela peut créer une sensation de flottement lors de l'écoute de l'œuvre musicale. De plus, les objets du niveau 3 ont une durée très variable, passant de 10 à 20 objets du niveau 0. Le dernier objet jaune (instants 35 à 42) est considéré comme faisant partie du niveau 3 alors qu'il est plus court que le quatrième objet (instants 17 à 25) du niveau 2. Cela maintient une ambiguïté rythmique dans l'œuvre.

Améliorations

Chaque visualisation comporte des intérêts d'un point de vue utilisateur. D'après les retours d'étudiants en Master 2 Musicologie à Sorbonne Université, la visualisation en deux dimensions est plus proche de ce que les usagers ont l'habitude de voir et de travailler. La théorie des diagrammes formels multi-échelles demande un effort de compréhension importante et cette visualisation évite une sur-couche de compréhension supplémentaire. En revanche, les représentations en trois dimensions proposent peut-être une lecture plus fine de la superposition des objets aux différentes échelles temporelles.

La question d'une représentation permettant la comparaison de différentes visualisations d'un même morceau n'a pas encore été étudiée, de même que la représentation de la polyphonie. Sur les diagrammes 2D, il est tout à fait possible de représenter manuellement deux matériaux présents au même instant temporel, mais ce n'est pas géré dans la modélisation automatisée des diagrammes formels, et on peut se poser la question de la lourdeur de la visualisation 3D avec plusieurs matériaux activés à un même instant donné.

Par ailleurs, si le degré de similarité entre les matériaux est calculé dans le modèle, il n'est pas représenté dans la visualisation. Dans la représentation 2D, on pourrait envisager de jouer avec les niveaux de gris : à un instant t donné, chaque matériau serait plus foncé si le matériau actuel est plus proche du matériau en question. Si l'idée de modifier l'ordre de l'axe des matériaux afin de rapprocher les matériaux similaires semble pertinente, cela déstructure en réalité le principe des diagrammes formels qui est de représenter les matériaux par ordre d'apparition. Nous laissons donc de côté cette idée.

5.3 Visualisation des phénomènes d'attention

Les phénomènes d'attention peuvent également être visualisés sur les diagrammes formels. Comment représenter alors la mémoire ainsi que les hypothèses et l'attente de segmentation au cours de l'acquisition de l'œuvre musicale ?

5.3.1 La mémoire

La mémoire à long terme, qui désigne dans notre logiciel les objets qui ont été segmentés, peut être représentée avec une opacité réduite. La mémoire en voie de constitution, c'est-à-dire les objets de niveaux supérieurs en cours de construction peut quant à elle être représentée en opacité à 100%.

5.3.2 Les hypothèses

On propose également de représenter les hypothèses effectuées dans le logiciel d'une couleur distincte de celles utilisées pour représenter les autres niveaux. La couleur bleu par exemple peut être réservée exclusivement pour ces matériaux. Ainsi, un déplacement du matériau sera observé en fonction des modifications ou non des hypothèses ainsi qu'un changement de couleur lors de l'écoute effective du matériau : le déplacement visuel est donc équivalent à la charge cognitive associée à la modification et à l'invalidation des hypothèses.

Un exemple de la mémoire et des hypothèses est présenté en figure 5.6. Il s'agit d'une vue du plan de la structure, où les objets verts correspondent au niveau 0, les objets violets au niveau 1 et les objets rouges au niveau 2. Les objets en opacité plus faible (couleur plus claire) correspondent aux objets enregistrés dans la mémoire à long terme, tandis que les objets d'opacité plus élevée (couleur plus foncée) correspondent aux objets en voie de constitution pour chaque niveau. L'hypothèse effectuée en fonction de la mémoire à long terme et de l'objet en voie de constitution est représentée en bleu pour chacun des niveaux. Sur le plan de la structure, le matériau correspondant n'est pas distingué.

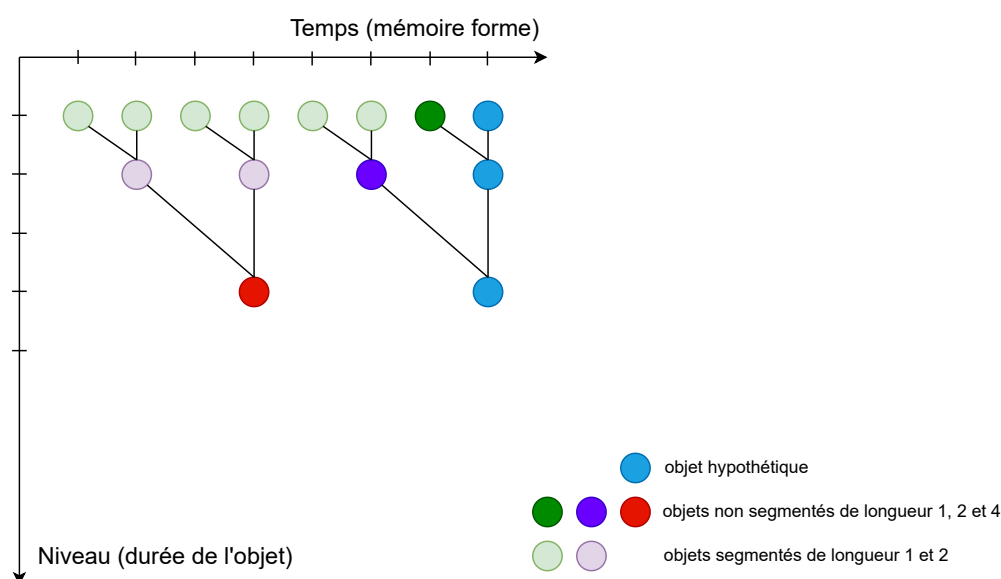


FIGURE 5.6 – Représentation de la mémoire et des hypothèses.

5.4 Interface

Une première version de l'interface pour le logiciel d'analyse a été développée. Elle a été développée avec l'aide d'un étudiant en sciences cognitives, Erwan Michel [104], pour le maquettage, ainsi qu'un groupe d'étudiants en informatique de l'ENSEIRB-Matmeca [17] pour le développement logiciel. L'objectif est que l'application soit facilement accessible et utilisable pour les musicologues.

Lors de l'ouverture de l'application, on tombe sur une fenêtre classique, avec une barre de menu permettant de créer, ouvrir et sauvegarder un projet. Ensuite, l'interface est divisée en deux parties : une première page permet de choisir l'agenda via une interface de paramétrisation, et une deuxième page affiche le rendu de l'analyse, permettant diverses interactions avec la représentation des diagrammes formels obtenus. Les fonctionnalités associées sont également présentées dans [17].

Le logiciel a été implémenté avec Electron³ : il s'agit d'un framework Javascript qui permet de développer et transformer des applications web en applications portables et multi-plateformes.

3. <https://www.electronjs.org/>

5.4.1 Agenda

La première page affichée lors du lancement du logiciel est l'interface de paramétrage (voir figure 5.7). Initialement, l'utilisateur est invité à importer un fichier. Actuellement, seuls les fichiers audio aux formats .wav et .mp3 sont compatibles. Juste en dessous du bouton d'import, un bouton donne accès à des ressources théoriques concernant l'Algorithme Cognitif et MORFOS. Une fois le fichier audio sélectionné, les paramètres à affiner s'affichent.

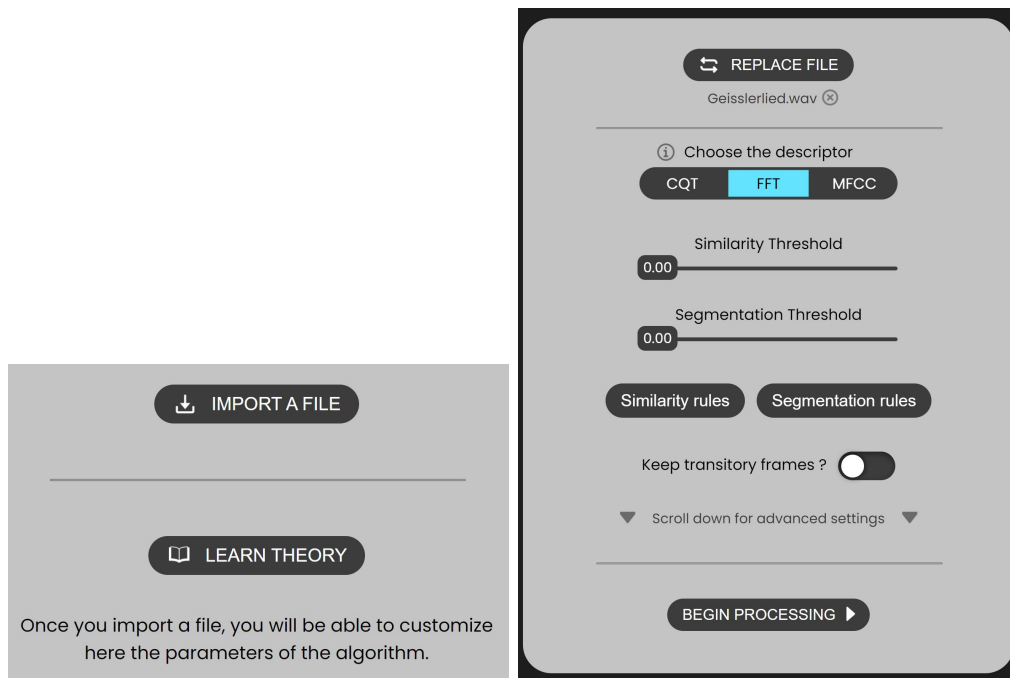


FIGURE 5.7 – Interface de paramétrage.

Paramètres définis

Une fois que le fichier audio est choisi, l'utilisateur accède à une série de paramètres à définir. Comme MORFOS comporte une dizaine de paramètres (voir partie [agenda]), tous ces paramètres ne sont pas encore intégrés à ce stade. Les paramètres proposés sont :

- les seuils de segmentation et de similarité ;
- le choix d'un descripteur à calculer pour le pré-traitement ;
- le choix de laisser ou non les matériaux transitoires de niveau 0 ;
- des hyper-paramètres tels que la taille de la fenêtre d'échantillonnage et le taux d'échantillonnage.

Ainsi, les paramètres laissés à disposition permettent une analyse assez peu fine pour le moment.

Widgets

Pour l'instant, les paramètres disponibles se présentent sous la forme de quatre types distincts de widgets : les *glisseurs*, les *relecteurs*, les *entrées libres* et les *boutons* (voir figure 5.8). Cette conception a été adoptée afin de permettre l'ajout aisé de nouveaux paramètres appartenant à ces quatre catégories. Néanmoins, il n'est pas possible malgré cette spécification d'introduire de nouvelles règles de classification et de segmentation, ni de charger un pré-réglage d'agenda. Cette fonctionnalité pourrait s'avérer utile, notamment si l'utilisateur constate que des agendas similaires fournissent des résultats pertinents pour un genre particulier, ou s'il souhaite ajuster seulement un ou quelques paramètres par rapport à un test précédent.

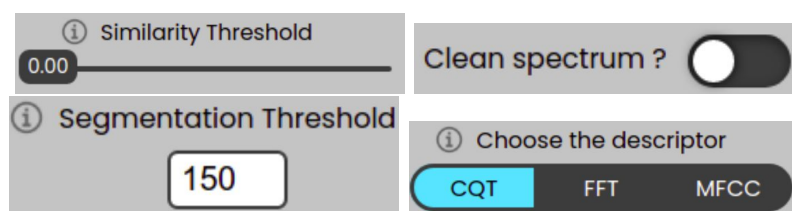


FIGURE 5.8 – Widgets. De haut en bas et gauche à droite : glisseur, bouton, entrée libre, sélecteur.

5.4.2 Interface graphique : visualisation de l'analyse

La fenêtre d'affichage de la vue 3D en figure 5.9 contient les fonctionnalités suivantes :

- affichage de la visualisation 3D ;
- affichage des plans 2D ;
- déplacement dans l'affichage ;
- sélection d'un ou plusieurs objets ;
- sélection d'un lien ;
- affichage de statistiques sur les objets ou les liens ;
- activation/désactivation de l'affichage d'objets/liens en fonction de leur niveau d'appartenance ;
- lecture du fichier audio ;
- gestion des projets.

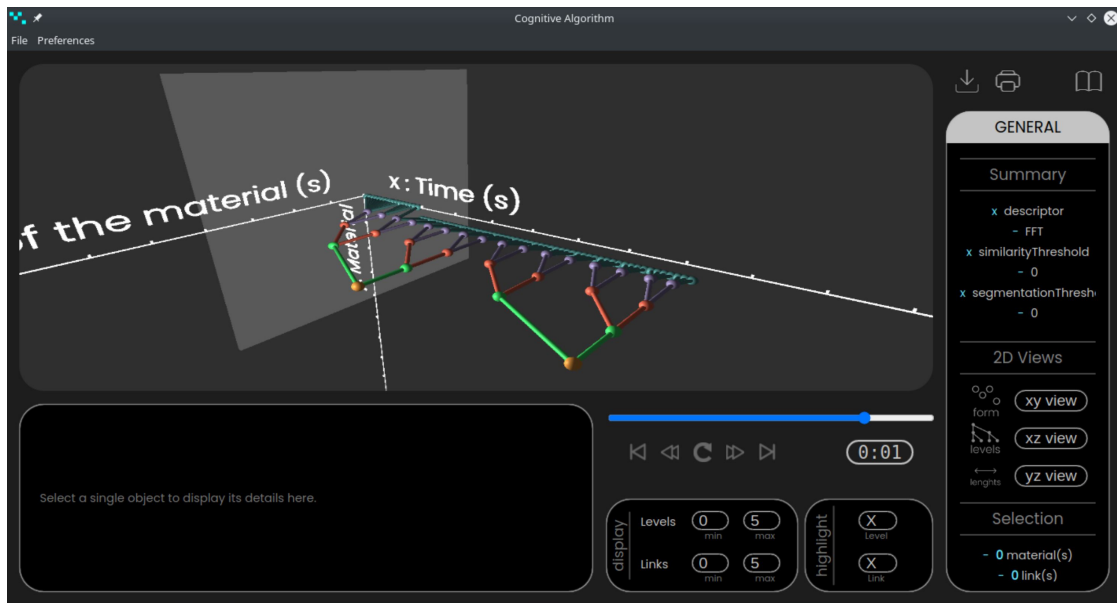


FIGURE 5.9 – Fenêtre d’affichage de la vue 3D.

Représentation des objets

Pour afficher la visualisation 3D il faut d’abord modéliser l’objet côté interface. Dans l’interface comme dans la modélisation présentée précédemment, un objet peut être situé dans le diagramme formel grâce à sa position dans l’espace 3 dimensions. Ainsi, ses coordonnées correspondent, sur l’axe x à la date d’apparition de l’objet, sur l’axe y : à la date de première apparition du matériau correspondant, et sur l’axe z à la durée du matériau, correspondant au niveau de l’analyse. Un objet est également déterminé par un identifiant $IDMap$ qui est la concaténation des 3 coordonnées du point, son niveau $level$, et le caractère correspondant à son label ID .

Par ailleurs, un objet est intégré dans une hiérarchie : le système de hiérarchisation est modélisé par la classe *family*. Cette classe permet d’ajouter, supprimer et accéder à l’enfant de l’objet de niveau n , c’est-à-dire à l’objet de niveau $n + 1$ qu’il constitue, et aux parents, qui sont les objets de niveau $n - 1$ qui le constituent. Les objets et leurs attributs sont stockés en tant qu’instance de la classe *sound_material*. Les classes *sound_material* et *family* sont présentées en figure 5.10.

Ainsi, une fois la fenêtre 3D créée et la scène chargée, l’interface affiche les

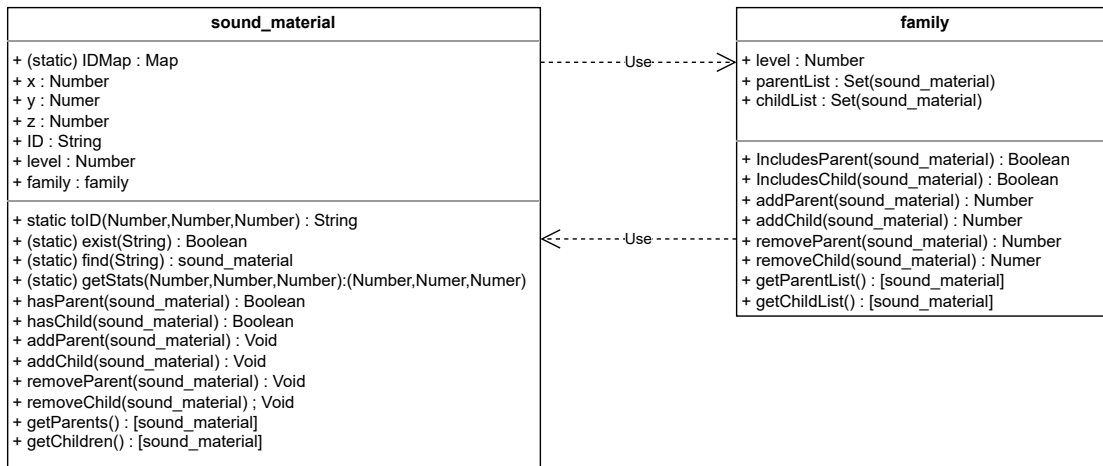


FIGURE 5.10 – Diagrammes de classe de *sound_material* et *family*.

matériaux et les liens générés par MORFOS. Pour cela, un fichier JSON *materials.json* est chargé. Celui-ci contient les objets successifs calculés par MORFOS, en particulier les informations représentées par l’identifiant *id*, la table des liens *links*, les coordonnées *coordinates* *x*, *y*, *z*, le numéro du matériau *mat_num*, le niveau *level* ainsi que l’extrait sonore concerné *sound* (voir figure 5.11). Le fichier est ensuite ouvert dans le contexte de la fenêtre 3D et analysé afin de construire au fur-et-à-mesure les objets de la classe *sound_material*.

```

{"id": 53, "links": [], "coordinates": {"x": 2.507755102040816, "y": 2.2755555555555556, "z": 0.046439909297052155}, "mat_num": 4, "level": 0, "sound": [-0.039960071444511414, -0.2
    
```

FIGURE 5.11 – Extrait du fichier JSON *materials.json* envoyé au logiciel après traitement de MORFOS.

Déplacements dans l’espace 3D et affichage des plans 2D

Concernant les déplacements sur le plan, le clic gauche permet de sélectionner les matériaux; le clic molette sert à la rotation, et le clic droit au déplacement dans la vue. La rotation de la molette sert quant à elle à zoomer ou dézoomer.

La figure 5.12, représente l’intégration des boutons d’affichage des trois plans 2D *x/y*, *y/z* et *x/z*. Lorsque l’utilisateur clique sur l’une de ces vues, la rotation est désactivée et seuls sont possibles la translation et le zoom/dé-zoom.

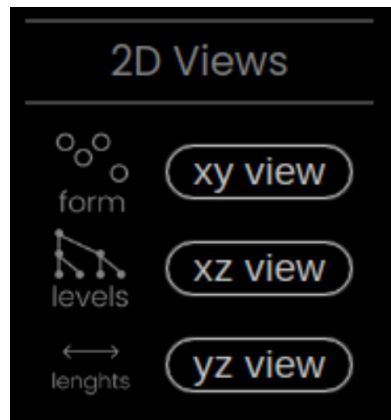


FIGURE 5.12 – Affichage des plans 2D.

Sélection des objets

On peut également sélectionner des éléments : un unique élément peut être sélectionné directement en cliquant dessus, tandis qu'un groupe d'élément sera sélectionné en maintenant la touche *shift*.

Cela nous permet d'une part de mettre en avant les éléments sélectionnés en les encadrant en jaune, mais cela permet également d'afficher dans un cadre au bas de l'écran un certain nombre de statistiques sur l'élément sélectionné. Pour un objet, y seront affichés notamment ses coordonnées, et pour un lien, les coordonnées des objets de départ et d'arrivée. Si l'utilisateur sélectionne plusieurs éléments, ils seront alors tous mis en avant avec le mécanisme de surlignage des éléments. Lorsque il sélectionne un objet, il est possible également de jouer le fichier audio correspondant. Pour le moment, si plusieurs éléments sont sélectionnés, ce sont les informations du dernier élément sélectionné qui sont affichées. Ces informations sont récupérées à partir de la position de l'objet, dont l'interface récupère l'identifiant unique à partir de ses coordonnées et qui permet d'identifier les informations de l'objet ainsi que les statistiques qui le concernent à partir de la classe *sound_material*.

Options d'affichage

L'utilisateur peut choisir d'afficher ou non certains éléments de la représentation (figure 5.13). L'affichage des liens peuvent alourdir la représentation, c'est pourquoi il est possible de ne pas les afficher. Par ailleurs, l'utilisateur peut choisir

le nombre de niveaux qu'il souhaite afficher. Pour cela, il est possible d'afficher ou non les matériaux en accédant à leur ID. L'utilisateur peut également choisir de surligner un ou plusieurs niveaux pour les mettre en valeur.



FIGURE 5.13 – Options d'affichage.

5.4.3 Gestion de l'audio

Un aspect qui n'a pas été souligné jusque là et qui est fondamental à l'analyse musicale, encore plus dans un modèle représentatif de l'écoute, est de pouvoir écouter l'œuvre analysée. Diverses fonctionnalités sont donc proposées pour pouvoir écouter celle-ci.

Bouton de lecture du fichier audio

Une barre de lecture permet de jouer le fichier audio dans son intégralité (figure 5.14). Celle-ci comporte les fonctionnalités suivantes :

- lire l'audio ;
- mettre l'audio en pause ;
- avancer/reculer de 5 secondes ;
- aller à la fin de l'audio ;
- aller au début de l'audio ;
- rejouer l'audio.

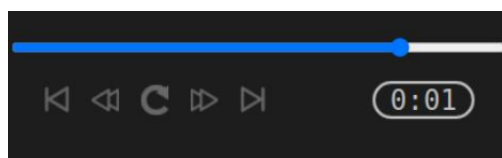


FIGURE 5.14 – Bouton de lecture.

L'intérêt et l'originalité de cette barre de lecture est d'afficher les objets correspondants aux matériaux entendus en temps réel. L'objectif est de permettre

une meilleure compréhension de la segmentation lors de l'écoute de l'œuvre, et des mécanismes d'attention lors de l'écoute musicale. La barre de lecture est placée à la fin de l'audio par défaut, avec tous les matériaux affichés. Le bouton *replay* est censé permettre de relancer l'audio et de faire disparaître les objets afin de les ré-afficher au fur et à mesure de la lecture. Pour le moment cette fonctionnalité n'a pas été implémentée et les matériaux restent affichés.

Lecture des sons de matériaux

Comme vu précédemment, l'interface possède une fenêtre dédiée à l'affichage d'informations. Les informations affichées sont les coordonnées du matériau, les statistiques liées à ces coordonnées, et un bouton permettant de jouer le son identifié par ce matériau.

En sortie d'exécution de MORFOS, les sons des matériaux sont représentés dans le fichier *material.json* par une séquence d'échantillons. Selon le théorème d'échantillonnage de Shannon [147], le logiciel de l'interface reconstruit alors à partir de cette séquence le signal correspondant au format *.wav*.

5.4.4 Lien entre l'interface et MORFOS

Pour lancer l'analyse depuis l'interface implémentée en Electron, il faut la faire communiquer avec le code Python de MORFOS. Cette communication doit d'une part pouvoir s'effectuer sans connexion internet, et par ailleurs pouvoir accepter l'ensemble des paramètres fournis par l'utilisateur via l'interface.

Barre de chargement

MORFOS ne permet pas encore des calculs en temps réel par rapport au déroulement de l'œuvre musicale. Afin d'améliorer l'expérience utilisateur lors du calcul du diagramme formel multi-échelle, une barre de chargement est affichée une fois que l'utilisateur a cliqué sur « lancer l'analyse » et avant d'afficher les résultats obtenus. Cependant, une telle fonctionnalité nécessite de faire communiquer l'application avec l'algorithme MORFOS implémenté en Python. Pour cela, nous avons ainsi implémenté un mécanisme de *checkpoints* dans l'algorithme MORFOS, permettant d'envoyer en sortie l'état de l'avancement de celui-ci. Une fois l'analyse terminée, l'algorithme MORFOS génère le fichier *materials.json* afin de stocker les résultats de l'analyse.

Connexion générale

Le script Python est lancé depuis l'interface via un processus enfant, ce qui permet une maintenance fiable sur toutes les plateformes, car le code Python mis à jour est directement exécuté au niveau de l'interface. Cependant, cela nécessite que l'utilisateur ait installé Python sur sa machine. Ainsi, il existe un guide⁴ pour installer Python en fonction de la plateforme de l'utilisateur, ainsi que l'ensemble des dépendances du logiciel.

Ainsi, les interactions entre l'interface et MORFOS sont représentées en figure 5.15

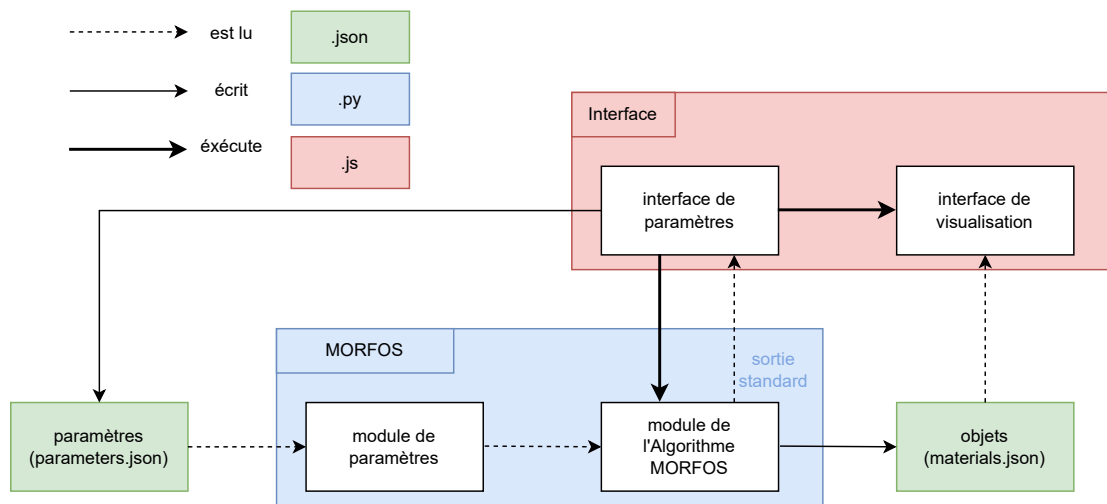


FIGURE 5.15 – Système d'interactions entre l'Interface et le MORFOS.

Tout d'abord, l'interface de paramétrage écrit dans le fichier *parameters.json* destiné à envoyer la valeur des paramètres au module de paramètres de MORFOS. MORFOS est exécuté depuis l'interface de paramétrage, et va analyser l'œuvre musicale avec les paramètres adéquats lus dans le module de paramètres. Au fur et à mesure de l'analyse, l'interface de paramétrage lit la sortie standard de MORFOS afin d'afficher la barre de chargement de l'analyse. Une fois l'analyse terminée, l'algorithme MORFOS écrit les séquences d'objets obtenus dans le fichier *materials.json*. L'interface graphique lit ensuite les données écrites dans *materials.json* et les affiche dans l'écran de visualisation 3D.

4. voir le guide accessible depuis le github de l'application.

5.4.5 Gestion des projets musicologiques

Il est important que les utilisateurs puissent garder en mémoire le projet d'analyse pour ne pas avoir à relancer l'analyse à chaque fois et pouvoir revenir ultérieurement sur l'étude. Les musicologues peuvent également souhaiter partager les résultats de MORFOS avec leurs pairs. C'est pourquoi un mécanisme de sauvegarde et de partage de projet est proposé au sein de l'interface.

Sauvegarde

Deux icônes (figure 5.16) permettent d'exporter le projet réalisé. La première icône permet d'exporter le fichier en PDF. Sur le PDF apparaît la vue actuelle du diagramme formel, ainsi qu'un texte décrivant l'ensemble des paramètres choisis pour le diagramme formel multi-échelle obtenu. La deuxième icône quant à elle permet d'imprimer physiquement la page de l'application.



FIGURE 5.16 – Bouton de sauvegarde et d'impression.

Le mécanisme de sauvegarde repose sur un système de *projet*. Lorsque l'algorithme est exécuté pour la première fois sur un fichier, les données pertinentes telles que les paramètres et la liste des matériaux sont enregistrées dans un dossier nommé *current_project*, principalement sous forme de fichiers JSON. C'est à partir de ce dossier que la visualisation en 3D est générée. Par la suite, il est possible de sauvegarder ce projet dans un répertoire de destination choisi. La sauvegarde s'effectue en copiant le contenu du dossier *current_project* vers le répertoire sélectionné.

Pour charger un projet existant, la procédure inverse est appliquée : le contenu du répertoire de destination est copié dans le dossier *current_project*. En effectuant cette opération, les résultats peuvent être visualisés sans nécessiter de relancer l'analyse.

Partage

De plus, un mécanisme de partage de projets permet aux utilisateurs de partager leurs résultats aisément. Une fonction dédiée permet d'exporter toutes les

données d'un projet en un seul fichier qui constitue une archive ZIP du dossier *current_project*. Pour importer un projet, l'archive ZIP est décompressée et son contenu est copié dans le dossier *current_project*. Le projet peut ensuite être visualisé ou sauvegardé selon les besoins.

Pour garantir la validité des projets importés ou ouverts, une mesure de sécurité est implémentée. Le logiciel vérifie que le fichier ou l'archive ciblée contient les fichiers JSON requis, qui devraient normalement être présents dans un dossier valide.

5.4.6 Améliorations

Afin d'améliorer l'interface, une visite guidée lors du premier lancement de celle-ci pourrait être implémentée et permettrait de montrer à l'utilisateur ce qu'il est possible de faire avec l'interface. En terme d'affichage, il faudrait afficher dans l'espace visualisation 3D les hypothèses effectuées au fur et à mesure de l'analyse. Il reste à ajouter par ailleurs l'intégralité des paramètres proposés dans l'Agenda.

Validation

Il reste également à valider la visualisation et l'interface du point de vue de l'expérience utilisateur. Pour ce faire, nous pourrions présenter les visualisations et l'interface à des utilisateurs afin d'obtenir leurs réactions à l'aide de questionnaires et d'entretiens. En outre, nous devrions effectuer des tests utilisateurs pour évaluer la facilité d'utilisation de l'interface. Pour cela, nous pourrions utiliser par exemple le questionnaire DEEP (Design-oriented Evaluation of Perceived Usability). L'utilisateur effectue une tâche sur l'interface, ce qui permet d'évaluer le contenu, la structure, l'architecture de l'information, la navigation, l'effort cognitif, et la cohérence et l'orientation visuelle. Nous pourrions alors les ajuster en conséquence. Enfin, nous pourrions effectuer une inspection en suivant les critères ergonomiques de Bach et Scapin [6]. Nous pourrions alors utiliser ces résultats pour réaliser de nouvelles versions de la représentation 3D et de l'interface, améliorant ainsi l'expérience des utilisateurs.

Conclusion

Afin de visualiser au mieux les plans de la forme, de la structure et de l'organisation présentés au chapitre 2, trois visualisations des diagrammes formels multi-échelle sont proposées.

Une première visualisation est en deux dimensions et il s'agit de celle qui a été exploitée jusque là dans la théorie des diagrammes formels. Cette visualisation consiste en l'alignement des diagrammes formels des différents niveaux sur la hauteur. Ainsi, les axes du temps sont alignés mais chaque diagramme formel possède son propre axe des matériaux. Cette visualisation ne permet pas une lecture efficace des plans de la structure et de l'organisation mais est la plus claire concernant le plan de la forme.

Une deuxième visualisation consiste à superposer les diagrammes des différents niveaux en les empilant de telle sorte à les aligner sur un troisième axe qui est celui des niveaux. Les objets sont représentés par des pavés de *profondeur* constante sur l'axe des niveaux, de *longueur* proportionnelle à la durée de l'objet sur l'axe du temps et de *hauteur* proportionnelle au niveau sur l'axe des matériaux. Si la vue temps/niveaux nous permet d'avoir une représentation de la structure, la vue matériaux/niveaux n'est pas cohérente et n'apporte pas d'information supplémentaire. Néanmoins, la vue matériaux/temps reste très proche de la visualisation en deux dimensions : on peut directement retrouver les diagrammes formels de chaque niveau en proposant une tranche par niveau de cette visualisation 3D. Afin de visualiser efficacement le troisième plan de l'organisation, une troisième visualisation est présentée. Celle-ci est également en 3 dimensions mais les objets sont représentés par des sphères de diamètre constant quelque soit la durée de l'objet et le niveau d'analyse. Les coordonnées du centre de la sphère dépendent alors sur l'axe du temps de l'instant d'acquisition de l'objet qui correspond à la segmentation des objets qui le constituent au niveau inférieur, sur l'axe des niveaux de la durée du matériau et sur l'axe des matériaux de la date de première apparition du matériau correspondant à l'objet. Des liens sont ajoutés pour connecter les objets de niveau supérieur et les objets de niveau inférieur qui les constituent. Le plan de la forme est légèrement modifié car l'axe des matériaux n'est plus compressé, tous les nouveaux matériaux apparaissant ainsi sur la diagonale du plan, mais le plan de l'organisation explicite les relations entre les matériaux des différents niveaux.

La mémoire et les phénomènes d'attention peuvent être également représentés en jouant sur l'opacité et la couleur des matériaux. Si une couleur différente est choisie par niveau d'analyse, les objets dans la mémoire long terme sont représentés en opacité faible tandis que les objets en voie de constitution sont représentés

en opacité 100%. Les hypothèses quant aux matériaux à venir, qui peuvent être calculées grâce à l'Oracle de Markov Variable, peuvent être représentées par une couleur choisie spécifiquement. Ainsi, si l'hypothèse est validée, l'objet apparent se colore dans la couleur du niveau, tandis que si l'hypothèse est invalidée il y a un déplacement de l'objet effectivement inscrit en mémoire.

Pour afficher ces visualisations, une interface en Electron a été implémentée et propose également un mécanisme de sauvegarde et de partage de projet. L'objectif de cette interface dans le futur est par ailleurs de proposer une visualisation dynamique de la représentation.

Chapitre 6

Applications et Analyses Musicologiques

Nous présentons dans ce chapitre des cas d'application de MORFOS. Dans un premier temps, nous présentons des cas d'études au niveau 0 du signal et sur la hiérarchisation d'œuvres précédemment utilisées comme exemples au cours de cette thèse qui sont le *Geisslerlied Maria Muoter Reinu Mait* et le *Rondo* du *Menuet K.545* de W.A. Mozart. Ces analyses nous ont permis d'étudier les limites et le comportement du logiciel et d'orienter notre implémentation.

Nous présentons ensuite un travail effectué avec E. Bichoy concernant l'analyse d'un corpus d'œuvre avec MORFOS et plus spécifiquement le Choral *O Haupt voll Blunt Wunder* de la *Passion selon Saint Matthieu BWV.244*, J.S. Bach ainsi que le *Concerto pour violon en la mineur RV356* de A. Vivaldi.

Nous avons effectué des études automatisées de la segmentation sur la base de données de musique Pop japonaise RWC Pop évaluées avec le protocole MIREX (*Music Information Retrieval Evaluation eXchange*), et nous proposons l'analyse spécifique des deux morceaux RWC Pop 21 et RWC Pop 31 afin de mieux comprendre les résultats obtenus. Enfin, nous avons également eu l'occasion d'utiliser ce logiciel dans le cadre d'un cours de Musicologie pour les étudiants en Master 1 de Sorbonne Université, dont nous présentons ici des retours et mises en perspectives. Ces différentes analyses, bien que toutes effectuées sur la version 1 de MORFOS, nous permettent de mettre en lumière les limites du logiciel.

L'intégralité des analyses présentées dans ce chapitre ont été effectuées avec la version 1.2 du logiciel, à l'exception de la section 1.1, qui bien que les études ont pu être effectuées dès le début de la thèse, présente les résultats obtenus avec la

version 1.3.

6.1 Analyses pour l'étude du logiciel

Nous présentons dans un premier temps des analyses effectuées afin de comprendre le fonctionnement du logiciel en fonction des paramètres de l'œuvre étudiée, et de ceux inférés dans l'Agenda de l'algorithme. Nous proposons ainsi dans un premier temps de présenter des analyses sur différentes versions du *Geisslerlied Maria Muoter Reinu Mait* à un seul niveau d'analyse audio. L'extrait du *Geisslerlied* étudié étant assez simple et peu pertinent aux échelles de temps plus élevées, nous nous sommes penchés dans un second temps sur diverses études de *Rondo du Menuet K.545* de W.A. Mozart. Nous proposons d'abord d'effectuer une analyse à partir de différentes segmentations sur une représentation symbolique puis une étude à partir de différentes performances audio et fichier MIDI au niveau 0, afin d'étudier le comportement hiérarchisé à partir d'un fichier audio. Enfin, nous présentons différents exemples de la base de donnée RWC Pop et l'étude que nous en avons faite pour déterminer de nouvelles règles de segmentation.

Toutes les fonctionnalités d'affichage n'ayant pas été implémentées dans l'architecture 2.0, nous présentons les résultats de la version 1.2 et 1.3.

6.1.1 Geisslerlied

Dans un premier temps, nous nous sommes concentrés sur l'étude du *Geisslerlied* présentée au chapitre 2. Afin de s'abstraire des variabilités de timbre et de dynamique qui peuvent être très importantes dans un enregistrement à partir d'instruments réels, nous avons cherché à voir si l'on réussissait à répondre aux problèmes de structuration et de similarité de l'audio avec des fichiers les plus propres possible : on a donc généré des fichiers audio à partir de synthèse de fichiers MIDI. Nous nous concentrons donc à un seul niveau d'étude qui est l'analyse du fichier MIDI du *Geisslerlied* synthétisé avec un son de Grand Piano au tempo 120bpm au niveau 0. Nous étudions ainsi l'impact des paramètres sur cet extrait ainsi que sur des variations du *Geisslerlied*.

Étude des paramètres

La figure 6.1 présente une analyse formelle de référence effectuée à la main à l'échelle de la note (déjà présentée au chapitre 2). Après génération du fichier MIDI du *Geisslerlied*, différents paramètres ont été testés empiriquement. On obtient un résultat conforme à la référence avec un paramétrage d'une taille de fenêtre de 1024 échantillons et un seuil de 0,97.

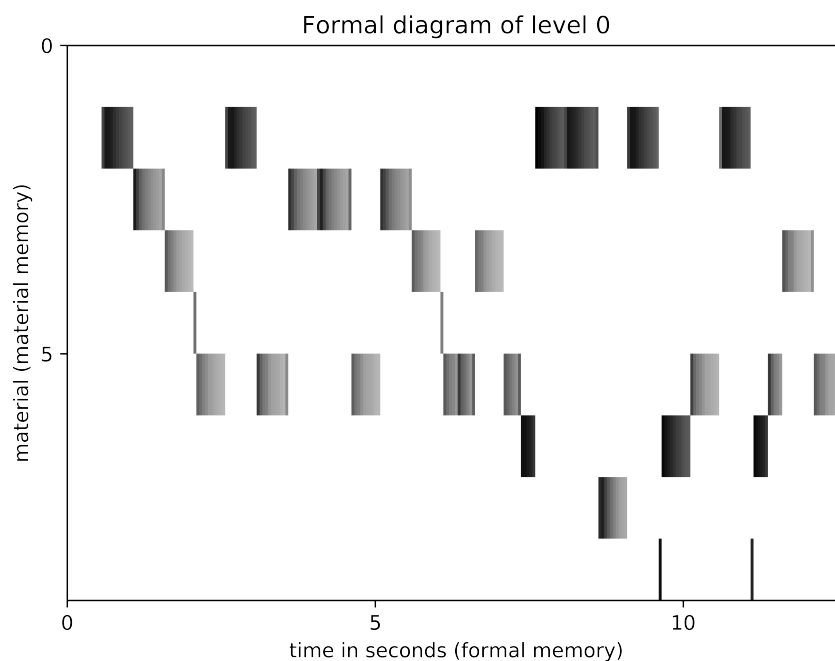


FIGURE 6.1 – Diagramme formel de niveau 0 du *Geisslerlied* avec une fenêtre de taille 1024 et un seuil de 0.97.

Le niveau de gris de chacun des objets correspond ici au volume associé à l'objet : c'est pourquoi nous avons un espace blanc d'environ une seconde au début du diagramme, qui correspond à un silence ajouté au début du morceau et associé au matériau 0.

On observe sur ce diagramme des objets intermédiaires entre les matériaux correspondant à des notes. Cela est dû au fait que le niveau est en réalité celui de la fenêtre, où chaque objet a la durée de 1024 échantillons : sont donc représentés des éléments de transition ainsi que des éléments correspondant à une hauteur de note. Le niveau hiérarchique de la note est donc le niveau direct supérieur. En effet, avec une règle de segmentation sur le changement de fréquence (dissimilarité fréquentielle), et un seuil de segmentation de 150, on obtient le diagramme formel

de niveau 1 en figure 6.2.

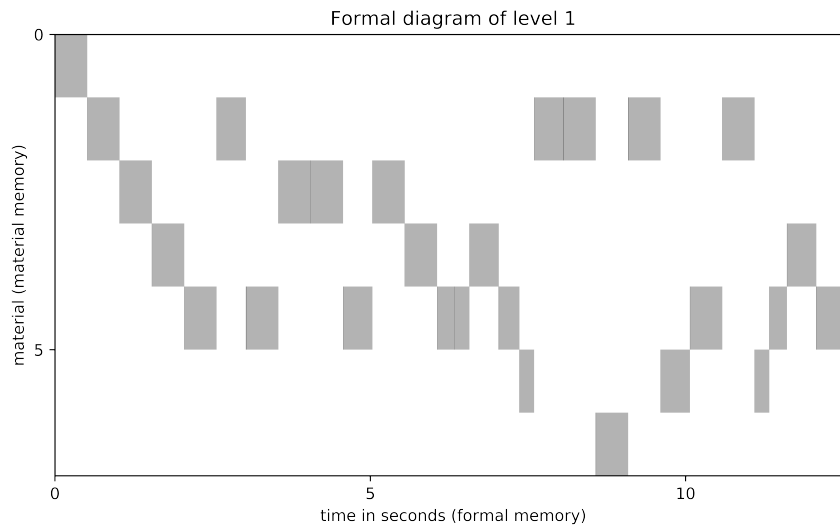


FIGURE 6.2 – Diagramme formel de niveau 1 du *Geisslerlied* avec une fenêtre de taille 1024, un seuil de similarité de 0.97 et un seuil de segmentation de 150.

Au niveau supérieur, l'information retenue est symbolique et la couleur grise est une couleur arbitrairement choisie, chaque matériau étant par ailleurs débuté par un liseré noir permettant de distinguer deux matériaux qui se succèderaient.

Sur cette version de l'algorithme, on propose alors de regrouper la frame transitoire avec l'objet qui le précède où le succède en fonction de celui qui a la similarité la plus proche. Nous nous permettons ainsi de regrouper les fenêtres de transitions avec les fenêtres correspondantes à une même note, d'une part car l'ordre de grandeur des transitions par rapport aux regroupements de fenêtres d'une même hauteur est très inférieure (moins d'un dixième de la note), et que l'on ne souhaite pas considérer simultanément deux ordres de grandeur différents. D'autre part, les éléments transitoires ne sont pas autonomes et sont liés aux éléments qui les précèdent et les succèdent, il est donc légitime de les regrouper avec ceux-ci. Cet algorithme se déroule en temps réel lors de l'acquisition de l'élément qui le succède. Le diagramme obtenu est représenté en figure 6.3. Ce diagramme semble identique (aux couleurs près) à celui de la figure 6.2, mais en figure 6.2 chaque bloc correspond à un unique objet de la durée d'une note, là où chaque note est constituée d'environ 12 objets en figure 6.3.

Dans les paragraphes suivants, nous comparons les résultats de la figure 6.1 avec de nouvelles analyses obtenues en faisant varier les paramètres de la taille de la fenêtre d'analyse et du seuil de similarité. On a ensuite synthétisé de nouveaux

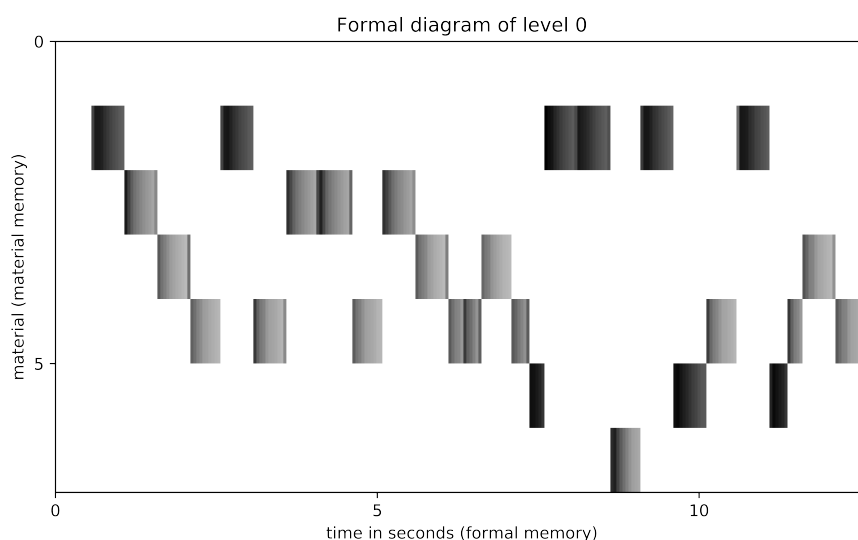


FIGURE 6.3 – Diagramme formel de niveau 0 du *Geisslerlied* avec une fenêtre de taille 1024 et un seuil de similarité de 0.97 avec algorithme de correction.

fichiers audio en concaténant deux versions du *Geisslerlied* avec un changement d’octave, avec un changement de nuance, et nous avons également comparé les diagrammes formels obtenus avec différents timbres.

Taille de la fenêtre d’analyse

En modifiant la taille de la fenêtre, on constate que la précision de l’étude de la similarité est modifiée. En effet, lorsque l’on diminue la taille de la fenêtre (512 échantillons en figure 6.4), celle-ci est trop petite pour obtenir des résultats cohérents à l’échelle auditive. Il est possible d’une part que la fenêtre soit assez petite pour que l’évolution de la dynamique de la note soit détectée, ce qui n’est pas forcément pertinent. Notamment, l’attaque d’une note pour un son percussif tel que le piano peut être perçue comme très différente de la suite de la note. D’autre part, c’est toute l’étude de similarité qui pourra être faussée car deux attaques percussives de notes différentes peuvent être considérées comme similaires car ayant des fréquences similaires à fortes amplitudes à une telle échelle de temps. Néanmoins, le parcours de l’oracle associé au phénomène continu du son lisse l’ambiguïté de la dynamique et l’algorithme va classer tous les matériaux dans la même catégorie.

Lorsque la fenêtre est plus grande (2048 échantillons en figure 6.5), l’étude de la similarité est moins précise : on obtient des éléments transitoires beaucoup plus gros avec une fenêtre de taille 2048 par rapport à une fenêtre de taille 1024.

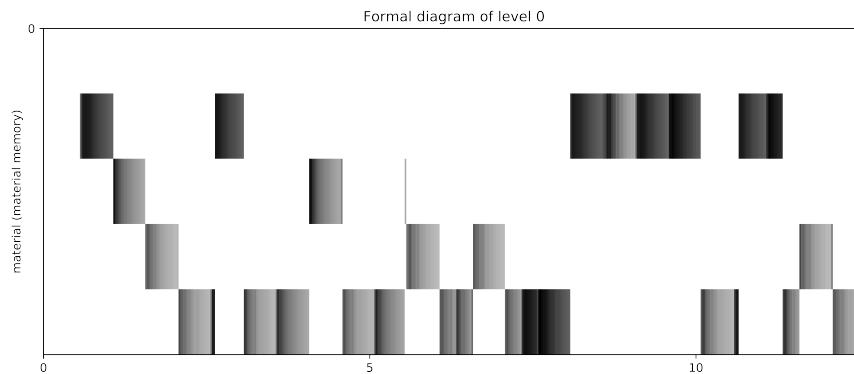


FIGURE 6.4 – Diagramme formel de niveau 0 du *Geisslerlied* avec une fenêtre de taille 512 et un seuil de similarité de 0.97 sans algorithme de correction.

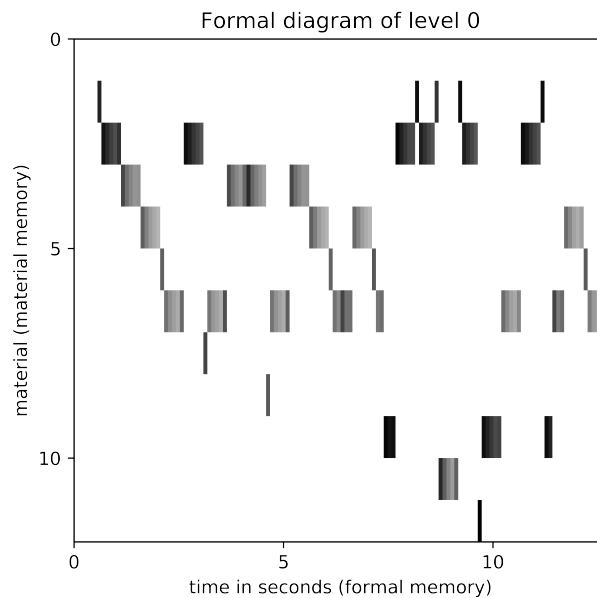


FIGURE 6.5 – Diagramme formel de niveau 0 du *Geisslerlied* avec une fenêtre de taille 2048 et un seuil de similarité de 0.97 sans algorithme de correction.

Variation du seuil de similarité

On a également regardé les résultats obtenus en faisant varier le seuil de similarité. On observe que lorsque l'on diminue ce seuil (seuil de 0,94 en figure 6.6), on perd en précision et en nombre de matériau, simulant ainsi une écoute beaucoup plus générale de l'œuvre, tandis que si l'on augmente ce seuil (seuil de 0,98 en figure 6.7), on aura une précision beaucoup plus fine de l'écoute, où chaque transition voire modification de dynamique du timbre dans la note sera repérée, augmentant ainsi le nombre de matériaux obtenus.

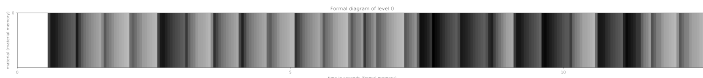


FIGURE 6.6 – Diagramme formel de niveau 0 du *Geisslerlied* avec une fenêtre de taille 1024 et un seuil de similarité de 0,94 sans algorithme de correction.

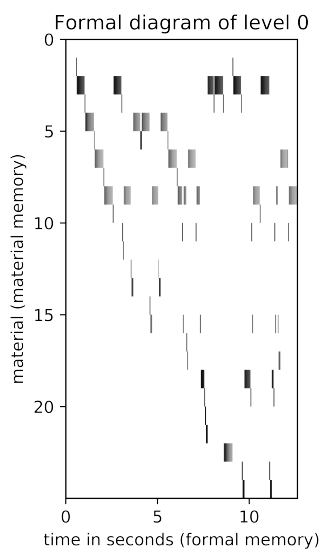


FIGURE 6.7 – Diagramme formel de niveau 0 du *Geisslerlied* avec une fenêtre de taille 1024 et un seuil de similarité de 0,98 sans algorithme de correction.

Comparaison des nuances

La deuxième modification a consisté à concaténer le *Geisslerlied* deux fois avec deux volumes différents : un volume élevé en premier et un volume plus faible en second. Le diagramme obtenu est présenté en figure 6.8.

On remarque que les nuances ne sont pas repérées, ce qui semble cohérent avec la normalisation lors du calcul de la distance cosinus. On peut voir cependant la différence de volume dans les nuances de gris de objets.

Comparaison des timbres

Nous avons également comparé le *Geisslerlied* synthétisé avec différents sons : le deuxième son que nous proposons est une synthèse de son de guitare classique. Nous obtenons en figure 6.9 les résultats avec exactement les mêmes paramètres

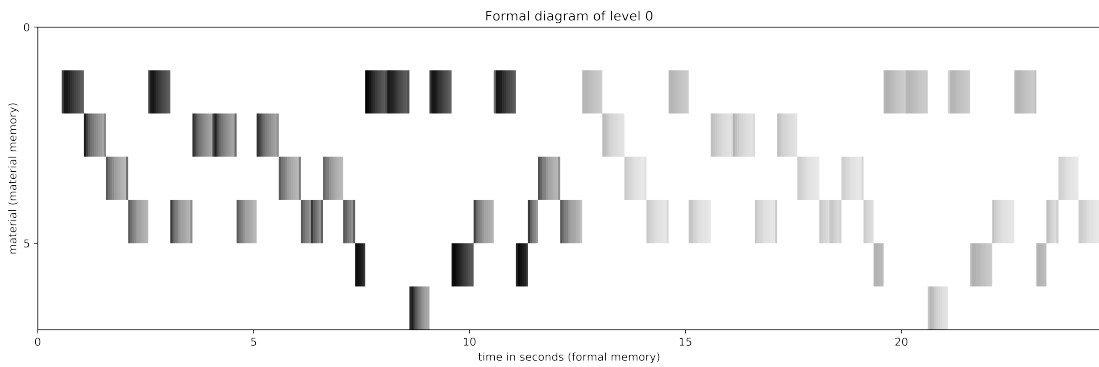


FIGURE 6.8 – Diagramme formel de niveau 0 du *Geisslerlied* joué deux fois : une première fois à volume élevé et une deuxième fois à volume faible avec une fenêtre de taille 1024 et un seuil de similarité de 0,97 et algorithme de correction.

que la figure 6.1.

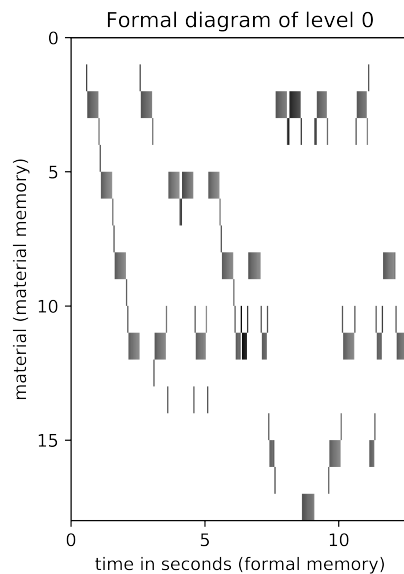


FIGURE 6.9 – Diagramme formel de niveau 0 du *Geisslerlied* synthétisé à la guitare avec une fenêtre de taille 1024 et un seuil de similarité de 0.97, sans algorithme de correction.

On constate que l'on obtient encore plus de matériaux transitoires : ce timbre semble être plus sensible aux variations et il faudrait diminuer légèrement le seuil de similarité. Lorsque l'on applique l'algorithme de correction, on obtient cependant

le même résultat, affiché en figure 6.10.

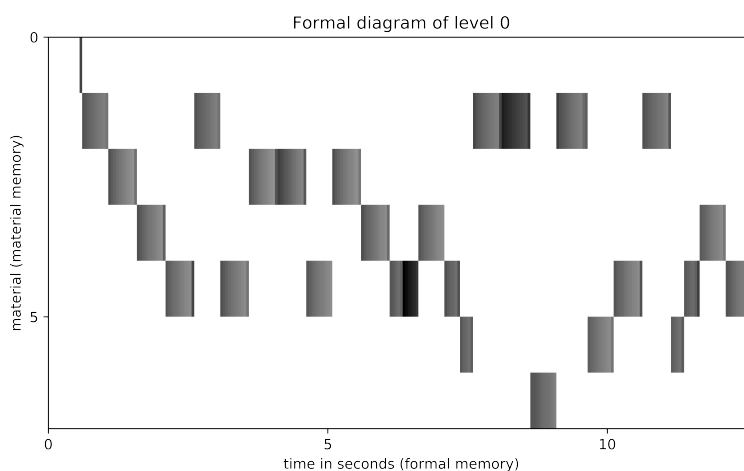


FIGURE 6.10 – Diagramme formel de niveau 0 du *Geisslerlied* synthétisé à la guitare avec une fenêtre de taille 1024 et un seuil de similarité de 0.97, avec algorithme de correction.

Nous avons ensuite souhaité étudier le résultat obtenu lorsque l'on analyse successivement le *Geisslerlied* synthétisé au piano avec celui synthétisé avec un son de guitare. On propose en figure 6.11 le diagramme formel obtenu avec les mêmes paramètres que les diagrammes de référence au piano et à la guitare, sans algorithme de correction (haut) et avec algorithme de correction (bas).

Sans surprise, nous retrouvons un plus grand nombre de frames transitoires lors du passage de la guitare. Nous constatons par ailleurs que les deux timbres ne sont pas distingués lors de l'analyse, et seules les notes déterminent les classes des matériaux : cela est dû notamment au choix du critère de pré-analyse du son, traité par calcul de la Transformée à Q Constant. Si l'on choisit de calculer des *Mel-frequency cepstral coefficients* avec calcul de 50 coefficients, le résultat obtenu est présenté en figure 6.12.

On constate alors que non seulement les timbres sont distingués, mais également toutes les notes, à l'exception du fa (matériau 1) qui est confondu pour les deux instruments.

Étude de la polyphonie

On propose d'étudier maintenant une version du *Geisslerlied* jouée en canon dont la deuxième voix débute 4 temps après la première. On présente la partition

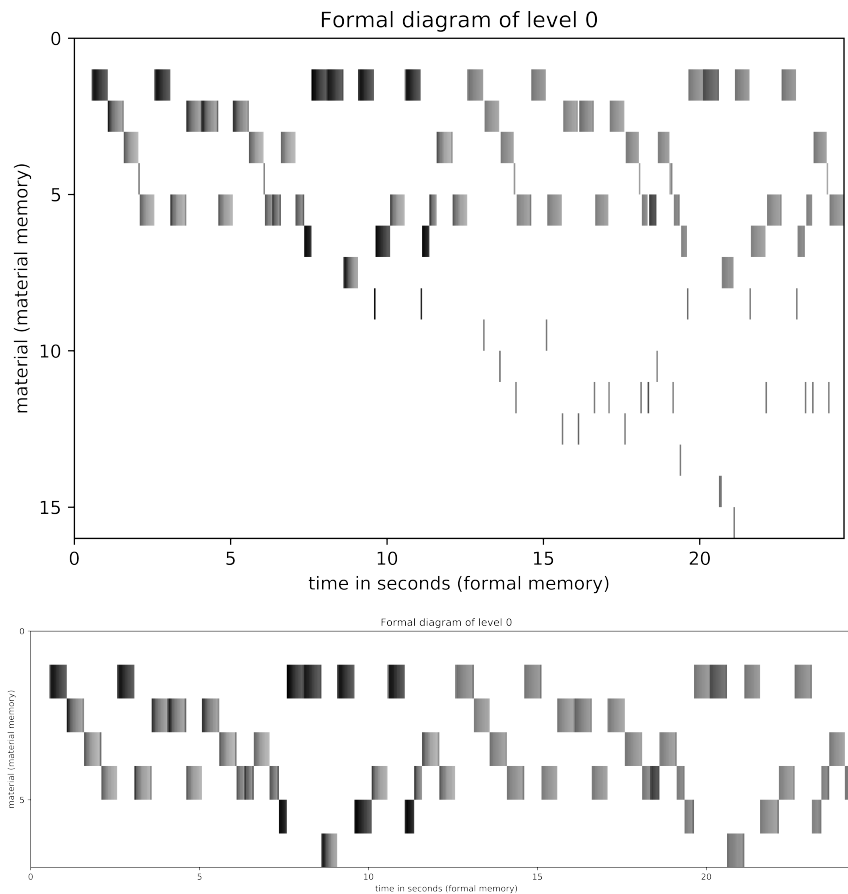


FIGURE 6.11 – Diagramme formel de niveau 0 du *Geisslerlied* synthétisé au piano puis à la guitare avec une fenêtre de taille 1024 et un seuil de similarité de 0.97, sans (haut) et avec (bas) algorithme de correction.

(figure du haut) ainsi que l'analyse formelle manuelle (figure de gauche) associée en figure 6.13. On suppose ici qu'à partir d'un certain seuil de similarité, la combinaison de notes, qui peuvent avoir déjà été entendue, donne la création de nouveaux matériaux. On présente également en figure 6.13 une analyse (figure de droite) avec utilisation de la Transformée à Q Constant (CQT) pour le pré-traitement du signal ainsi qu'un seuil de 0.978. Les résultats obtenus sont similaires à deux erreurs près.

On propose alors de représenter une forme de polyphonie sur le diagramme formel : il s'agit, pour chaque objet acquis, d'afficher en niveau de gris la similarité avec tous les matériaux précédemment entendus. Dans la figure 6.14, nous présentons à gauche une analyse manuelle des résultats attendus avec activation de ce

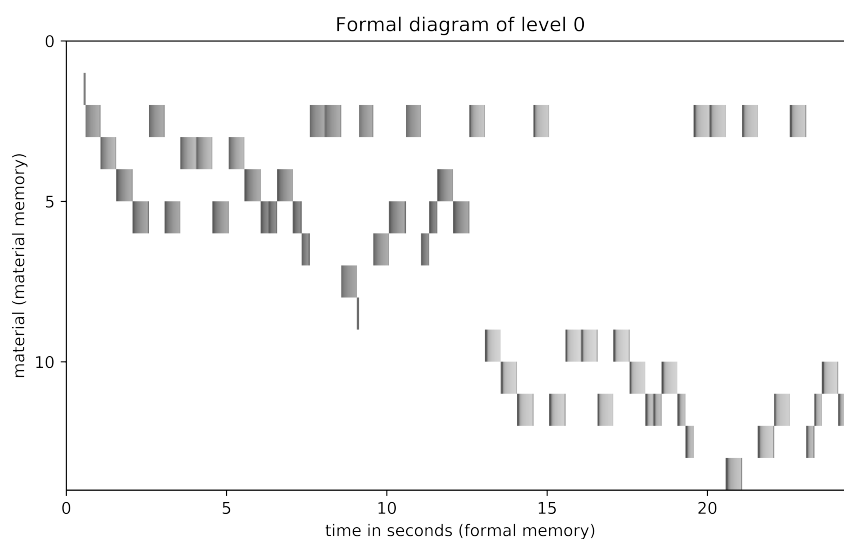


FIGURE 6.12 – Diagramme formel de niveau 0 du *Geisslerlied* synthétisé au piano puis à la guitare avec calcul de 50 coefficients cepstraux.

paramètre et à droite une analyse automatisée avec l’activation de ce paramètre et calcul de la CQT. On constate du bruit lié à l’apparition de nouveau matériau, qui est dû à l’imprécision de la constante Q transform dans les fréquences basses (voir figure 6.15). On propose alors une analyse avec pré-traitement du signal en calculant la Transformée de Fourier Rapide (FFT) (droite) : celle-ci est beaucoup plus précise, à deux erreurs près. Néanmoins, comme observé déjà sur les analyses manuelles, nous constatons qu’un tel affichage ne simplifie pas la lecture du diagramme formel. D’une part, l’apparition chronologique des matériaux combinés, et non pas des matériaux en tant que flux dissociés, modifie intrinsèquement la forme du diagramme. D’autre part, il existe énormément de matériaux intermédiaires contenant une des notes de la polyphonie, ce qui rend la représentation chargée, sans tenir compte des imprécisions éventuelles de l’analyse spectrale préalable qui peut également ajouter du bruit.

Une suggestion pour aller vers la polyphonie serait d’intégrer des flux séparés pour les différentes voix dans le logiciel mais qui partageraient la même mémoire. Chaque objet des différentes voix doit être analysé dans l’ordre chronologique d’apparition, le logiciel devra alors parcourir alternativement chacune des voix pour sélectionner le prochain objet. Les objets joués en simultanés seront donc traités successivement mais en tant que processus groupé : une note d’un accord traitée en deuxième sera évaluée sans considérer l’acquisition dans la mémoire de la première note traitée de l’accord.

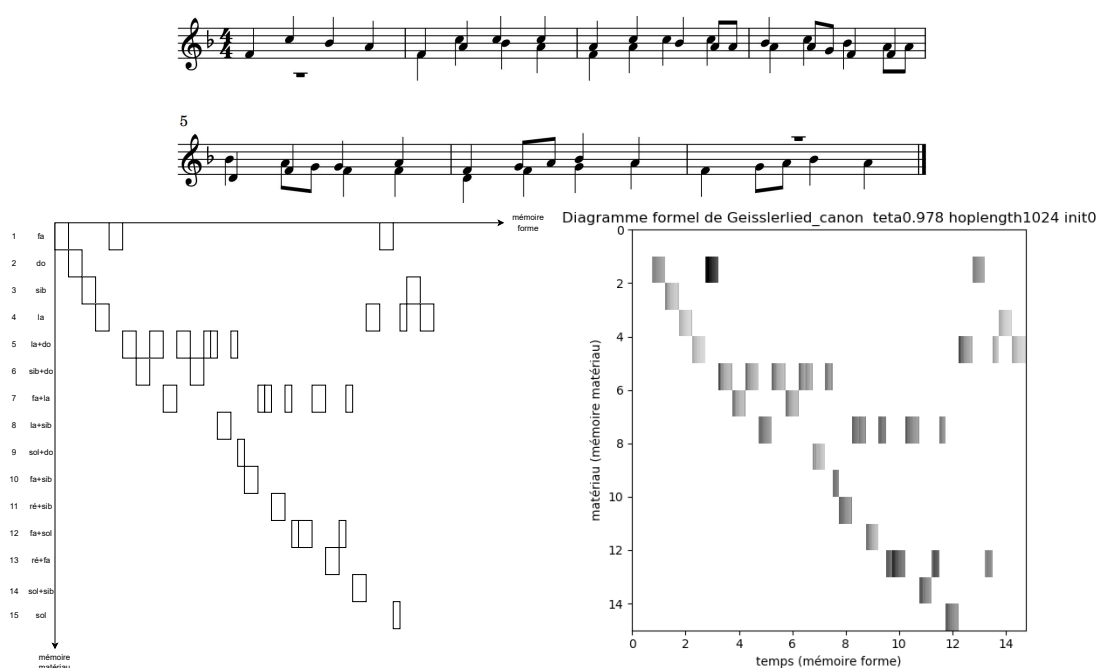


FIGURE 6.13 – Partition (haut) et Diagramme formel de niveau 0 du *Geisslerlied* joué en canon obtenue manuellement (gauche) et automatisée avec calcul de la cqt (droite).

Bilan

Nous constatons ainsi que les résultats de MORFOS sont très sensibles aux variations de paramètres. Bien que cette sensibilité n'aille pas en faveur d'une robustesse du logiciel, cela permet également d'illustrer différent mode d'écoutes à travers le logiciel. Ainsi, nous laissons le soin à l'utilisateur via l'Agenda de choisir l'ensemble des paramètres pour l'analyse effectuée.

6.1.2 sonate pour piano n°16 en Do Majeur de W.A. Mozart

Nous avons ensuite étudié le *Rondo* extrait de la *sonate pour piano n°16 en Do Majeur* de Wolfgang Amadeus Mozart introduit dès le chapitre 2. Nous proposons dans un premier temps une étude à partir d'une analyse signal au premier niveau, puis nous reprendrons brièvement l'étude symbolique.

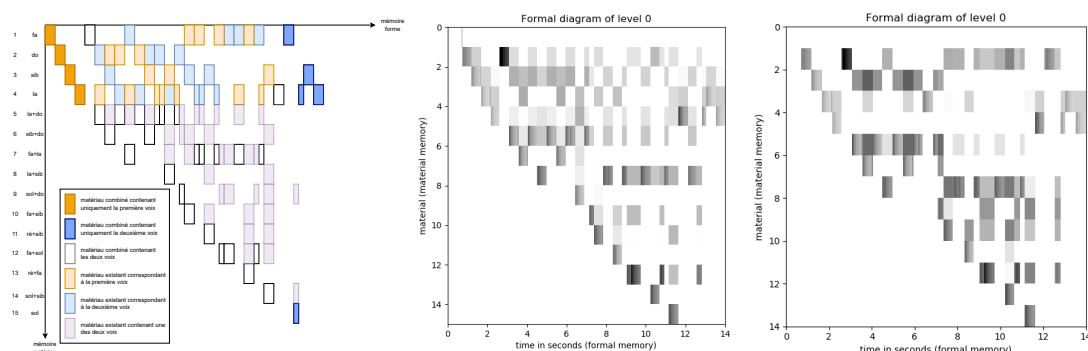


FIGURE 6.14 – Diagramme formel de niveau 0 du *Geisslerlied* joué en canon obtenue manuellement (gauche) et automatisée avec calcul de la cqt (milieu) et de la fft (droite) avec affichage de la polyphonie.

Étude signal de niveau 0

Dans un premier temps, nous avons analysé avec MORFOS 1.2 deux versions sonores du *Rondo*. Une première analyse est une synthèse du fichier MIDI correspondant avec un son de grand piano présentée en figure 6.16 et le second est une analyse à partir d'une performance musicale réelle (droite) présentée en figure 6.17.

Dans les deux versions, nous pouvons remarquer certains matériaux artefacts qui brulent la représentation. Néanmoins, et c'est là que l'interface de MORFOS 2.0 présentée au chapitre 5 nous sera utile également, il serait intéressant de constater précisément à quels objets audio ces artefacts correspondent. On constate dans la version MIDI que le refrain du *Rondo* est particulièrement explicite : celui-ci se décompose en 2 parties A (0' à 4') et B (4' à 8'), elles-mêmes subdivisées en deux sous-parties *ab* (0' à 2' et 2' à 4') et *ac* (4'-6', 6'-8'). Nous constatons également le déploiement de deux développements (aux secondes 18' et 38' environ) avec la découverte de nombreux nouveaux matériaux. Enfin, nous retrouvons la coda (à partir de la seconde 76'). Celle-ci est séparée en deux parties G (76' à 82') et H (82' à la fin) : la partie G est la répétition du même motif (76' à 79' et 79' à 82'), lui-même subdivisé en un premier motif répété deux fois (76' et 77' environ) et un second motif (77' à 79' environ). La partie H quand à elle est constituée du même motif répété deux fois (82' et 85') et d'un troisième motif distinct (88'). On retrouve donc la forme générale du *Rondo* à l'aide de cette analyse.

Sur l'exemple performé, on constate que les refrains, bien que présents, sont beaucoup moins distingués, tout comme la forme de la coda. En revanche, il est beaucoup plus frappant de constater que le deuxième développement contient en

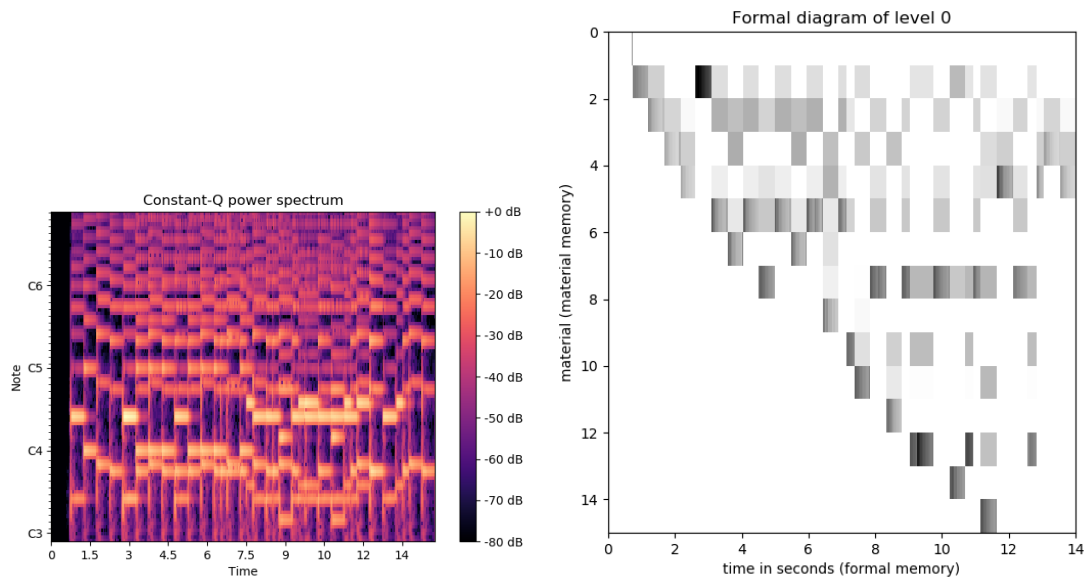


FIGURE 6.15 – Spectre de la cqt (gauche) et Diagramme formel de niveau 0 (droite) du *Geisslerlied* associé.

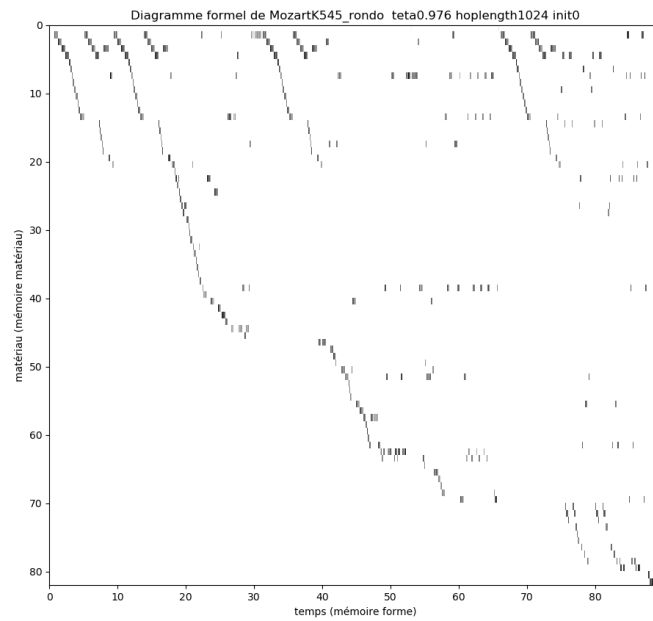


FIGURE 6.16 – Diagramme formel de niveau 0 du *Rondo K.545* de W.A. Mozart obtenu à partir d'une synthèse MIDI.

réalité quelques répétitions : le début du développement (55') est repris vers 70', et

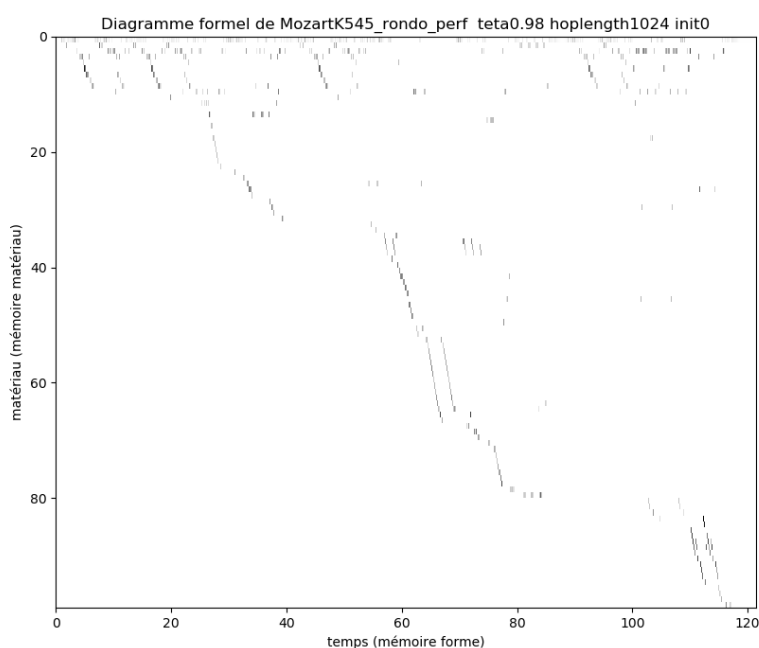


FIGURE 6.17 – Diagramme formel de niveau 0 du *Rondo K.545* de W.A. Mozart obtenu à partir d’une performance musicale.

c’est tout une partie du développement qui est reprise immédiatement des secondes 65 à 70.

Étude symbolique

A partir de la partition du *Rondo K545* de W.A Mozart, nous extrayons à l’échelle des termes une représentation symbolique de l’œuvre sous la forme de la séquence symbolique suivante : *abacabacdeabfgabachijklmhinopqabacrsrsttu*. À cette séquence symbolique, nous avons inféré via l’Agenda des règles de classification (stricte égalité des chaînes de caractères ou mesure de l’alignement) et de segmentation (sous ensemble de règles de grammaires présentées en chapitre 3).

Différentes segmentations sont largement présentées dans les chapitres 3 et 4, auxquelles nous invitons le lecteur à se référer. Dans le chapitre 3, nous rappelons comment l’obtention des différents diagrammes selon les règles appliquées nous a permis d’inférer de nouvelles règles. Dans le chapitre 4, nous présentons différentes segmentations (dont certaines implémentées pour l’expérience et qui ne sont pas des règles de grammaire pertinentes) et nous comparons les segmentations obtenues avec la complexité correspondante. Nous proposons néanmoins de nous attarder

dans cette section sur le cas spécifique où la règle de l'alignement est choisie et les 4 règles 1 (textithypothèse d'un objet de niveau supérieur), 2 (*reconnaissance d'un objet de niveau supérieur*), 3 (*reconnaissance d'un objet de niveau supérieur précédent*) et 5 (*objet de niveau supérieur en voie de construction*) sont activées pour la segmentation.

Le diagramme formel obtenu est présenté en figure 6.18.

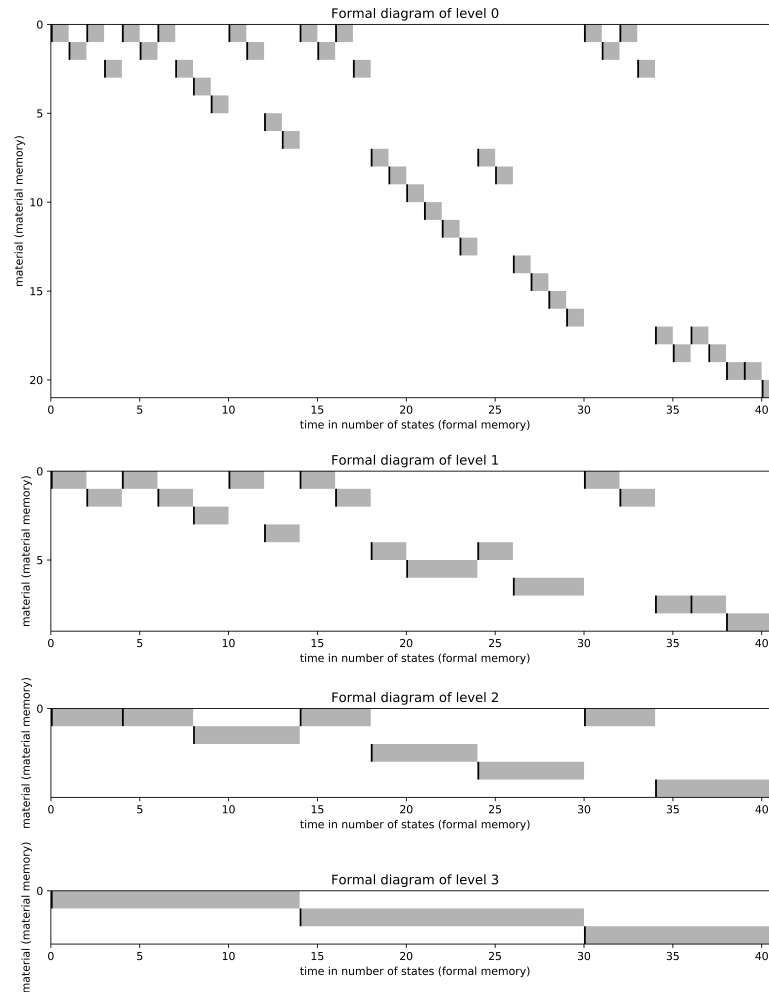


FIGURE 6.18 – Diagramme Formel Multi-échelle du *Rondo K.545* de W.A. Mozart à partir d'une représentation symbolique.

Le niveau 0 correspond littéralement à la représentation formelle équivalente à la chaîne de caractères extraite depuis la partition. On remarque que l'on retrouve le découpage structurel présenté à partir de l'analyse audio du Menuet dans la

partie précédente : le refrain est visible, constitué de deux parties subdivisées chacune en les matériaux a, b et a, c . On observe ensuite les deux développements $hijklm$ et $hinopq$ qui commencent tous deux par les deux matériaux hi et enfin la coda constituée d'une première partie qui est la répétition des matériaux rs et une deuxième partie qui est la répétition du matériau t suivi du matériau u .

Le niveau 1 synthétise les objets à une échelle intermédiaire qui est celle de la phrase musicale : chaque répétition interne aux différentes parties de l'œuvre donne lieu à un nouvel objet. Ainsi, le refrain est représenté avec ces deux parties, initiées par a au niveau 0, par les matériaux AB. Les deux développements sont respectivement représentés, pour le premier par les matériaux CAD : la première partie du refrain est donc ici reprise dans ce premier développement. Le deuxième développement est segmentée en deux nouveaux matériaux EF, tous deux initiés par le matériau h de niveau inférieur. Enfin, la coda est constituée de trois phrases : les deux premières sont la répétition de rs de niveau 0 pour donner G au niveau 1, et les deux itérations de t sont fusionnées avec u en un matériau H. C'est le niveau 2 qui correspond alors à l'échelle de la section : on retrouve bien le refrain avec le matériau A, les deux développements B et C ainsi que la coda D.

Enfin, le niveau 3 représente les différentes parties de l'œuvre. celui-ci peut être sujet à discussion : ici, chaque partie débute par l'expression du refrain. Cependant, et c'est une remarque qui pourrait se poser à différents niveaux, il n'est pas évident de définir si la répétition d'un élément débute nécessairement un nouvel objet, ou bien s'il peut le conclure.

6.2 Analyses musicologiques sur un corpus spécifique

Avec Elia Bichoy, doctorant en musicologie à Sorbonne Université, nous avons procédé à l'analyse de la forme multi-échelle obtenue à partir de la représentation audio de diverses œuvres. Deux d'entre elles sont présentées dans cette section :

- Choral *O Haupt voll Blunt Wunder* de la *Passion selon Saint Matthieu BWV.244*, J.S. Bach ;
- *Concerto pour violon en la mineur RV356* de A. Vivaldi.

6.2.1 *Passion selon Saint Matthieu BWV.244, J.S. Bach*

La passion selon Saint Matthieu est un oratorio composé par J.S. Bach et créé dans la première moitié du dix-huitième siècle. Il décrit la fin de l'Évangile selon Mathieu, à savoir la partie du jugement jusqu'à la crucifixion du Christ dans une pièce composée pour deux orchestres, deux chœurs et solistes. Nous nous intéressons spécifiquement ici au Choral *O Haupt voll Blunt Wunder*. Ce choral se divise en une partie répétée deux fois : cette partie est constituée de la répétition du même refrain deux fois, avant un développement. Chaque refrain est constitué de deux phrases musicales, et le développement de quatre phrases.

La structure hiérarchique est représentée dans le tableau suivant :

A						A									
A		A		B		A		A		B					
a	b	a	b	c	d	e	f	a	b	a	b	c	d	e	f

La figure 6.19 présente l'analyse formelle du Choral *O Haupt voll Blunt Wunder* de la *Passion selon Saint Matthieu BWV.244*, J.S. Bach obtenue avec MORFOS 1.2 et commentée manuellement. En vert sont encadrées les sections correctement répétées, en jaune les sections avec quelques modifications ou ambiguïtés et en violet les sections correspondant à du bruit ou des variations timbrales.

Nous constatons que le logiciel reproduit assez fidèlement les différentes sections, et ce jusqu'au niveau 2. Des légères variations sont observées entre la première et la deuxième occurrences alors qu'il s'agit d'une reprise stricte : cela peut être dû aux différentes variations de timbre, voir de phonèmes liés aux paroles. Le niveau 1, bien que constitué de blocs plus larges que le niveau 0 (et induisant un peu plus d'imprécisions), n'apporte pas vraiment plus d'information sur la forme du choral, alors que l'échelle des paragraphes n'est pas reproduite ici.

6.2.2 *Concerto pour violon en la mineur RV356, A. Vivaldi*

Le *Concerto pour violon en la mineur RV356* est extrait de l'*estro armonico*, un ensemble de 12 concertos pour un, deux, ou quatre violons, orchestre et basse continue, composé par Antonio Vivaldi à l'attention de Ferdinand III de Médicis. Il s'agit du concerto n°6 pour violon et cordes. Nous nous intéressons plus spécifiquement à l'Allegro.

La pièce alterne entre les ritournelles, au nombre de 6, jouées par l'orchestre et

6. Applications et Analyses Musicologiques

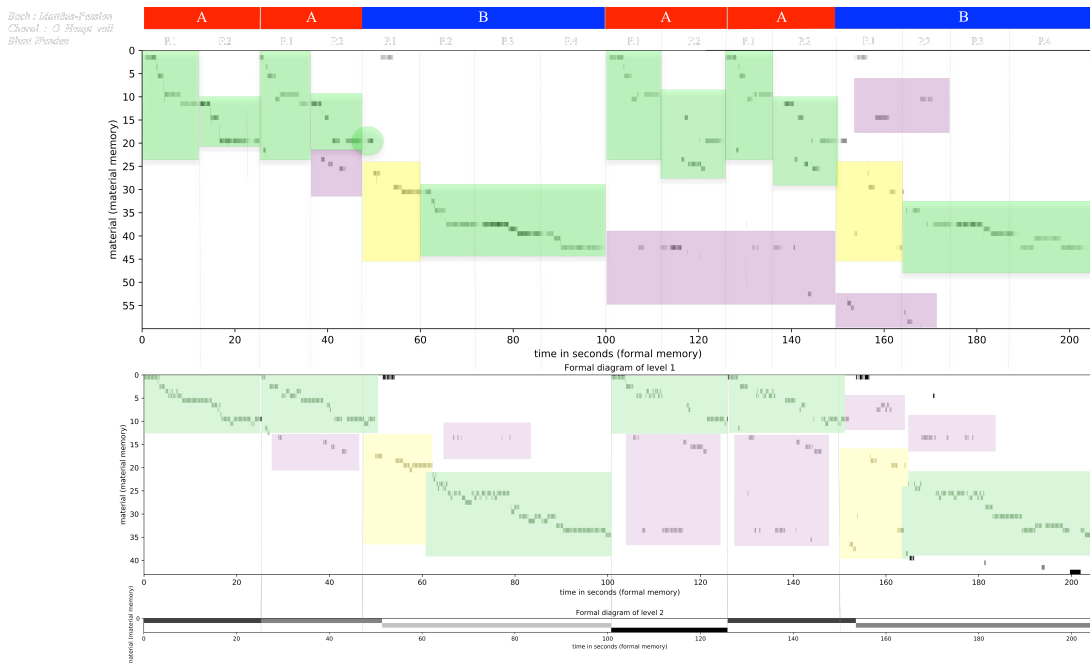


FIGURE 6.19 – Diagrammes formels de niveau 0, 1 et 2 du *Choral O Haupt voll Blunt Wunder* de la *Passion selon Saint Matthieu BWV.244*, J.S. Bach.

les cinq solos du violon.

Nous présentons en figure 6.20 l'analyse formelle du *Concerto pour violon en la mineur RV356* de A. Vivaldi. Comme pour l'analyse précédente, nous colorons en vert les segments retrouvés, en jaune les segments modifiés et en violet les segments attendus qui ont été analysés différemment. Les parties non colorées sont des développements qui ocurrent une seule fois dans l'œuvre.

Au niveau 0 et 1, bien que les phrases *a* et *c* sont retrouvées au début des solos 1 et 2 de la ritournelle 4 (pour la phrase *a*) et de la ritournelle 6 (pour la phrase *c*), nous constatons que de nombreuses phrases ne sont pas retrouvées : cela est dû aux nombreuses transpositions qui ont lieu au court de l'œuvre. En effet, la première occurrence de la phrase *d* dans le solo 1 est immédiatement transposée, et ainsi transcrite par d'autres matériaux. Il en est de même pour le retour du *a* et du *b* dans la ritournelle 3. L'avant dernier *c* n'est pas transposé, ceci dit le matériau le plus prépondérant de la première occurrence a été analysé. Ainsi, le logiciel n'est pour le moment pas robuste aux transpositions : une première proposition serait d'utiliser d'autres descripteurs pour étudier l'œuvre : selon la rythmique par exemple, ces phrases musicales auraient peut-être pu être repérées.

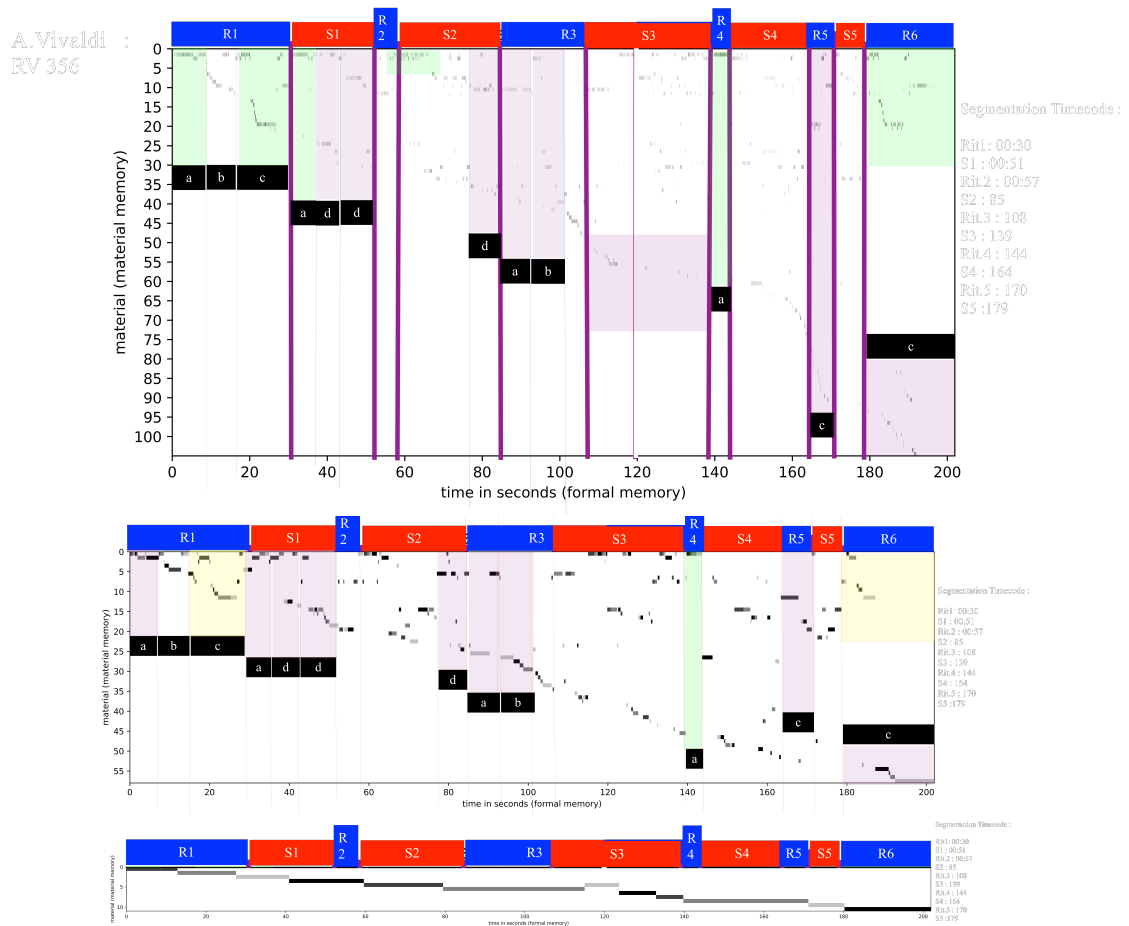


FIGURE 6.20 – Diagrammes formels de niveau 0, 1 et 2 du *Concerto pour violon en la mineur RV356* de A. Vivaldi.

Une autre possibilité est de s'intéresser non pas à l'étude des hauteurs absolues, mais aux intervalles entre les notes jouées. Par ailleurs, bien que les solos 3 et 4 soient de nouveaux développements thématiques, nous constatons le retour de nombreux matériaux déjà vus qu'il serait intéressant de déterminer. Le niveau 1 présente les mêmes résultats avec une perte de précision. Au niveau 2, nous pouvons comparer la segmentation de référence avec la segmentation obtenue avec le logiciel. Certaines parties sont bien segmentées, telles que le solo 5 et la ritournelle 6, et d'autres parties sont sur-segmentées, comme le refrain 1, et/ou fusionnées, comme le solo 1 et la ritournelle 2 qui sont fusionnés en deux matériaux *C* et *D*, ainsi que la ritournelle 4, le solo 4 et la ritournelle 5 fusionnés en un matériau *H*. Cependant, entre le solo 2 et le solo 3 inclus, il est difficile de déterminer une segmentation pertinente.

6.3 Étude de la base RWC Pop

Une étude a également été effectuée sur la base de donnée RWC pop¹, un corpus constitué de 100 morceaux de musique Pop produits par le Japan's National Institute of Advanced Industrial Science and Technology (AIST). Chaque morceau est fourni au format audio et MIDI, et une analyse symbolique a été calculée par Bimbot et al. [16]. Nous nous sommes intéressés ici à l'analyse du format symbolique. Chaque œuvre est structurée à trois échelles musicales que sont le temps, la mesure et la section. Chaque case du fichier correspond à une mesure musicale, et les accords contenus dans les mesures sont annotés selon une méthodologie précise. Chaque ligne du fichier correspond par ailleurs à une section dont le label est spécifié dans la première colonne selon des conventions de nommage.

Nous avons traité le fichier de telle sorte d'une part à décrire chaque temps de l'œuvre par un accord, puis à convertir cette notation en une chaîne de caractère au format pris en compte par MORFOS 1.2 selon l'algorithme présenté dans la section 3.2.1. Ainsi, chaque notation d'accord différente est traduite en un symbole différent, même s'il s'agit d'une variation d'un accord donné.

Nous avons ensuite fourni la chaîne de caractère au logiciel et analysé les segmentations obtenues.

6.3.1 Création de nouvelles règles et uniformisation des niveaux

Les diagrammes formels ont été calculés sur l'intégralité de la base de données. Une première analyse sans les règles bivalentes (6,7 et 8) présentées au chapitre 3 et qui ne sont pas utilisées pour le Rondo de Mozart nous a permis de constater une structure différente pour les morceaux de Pop étudiés par rapport au *Rondo* de Mozart. En effet, là où le *Rondo* est principalement constitué de développements segmentés par le retour à un début de segment précédemment vu, nous constatons que ces œuvres sont construites sur des répétitions successives d'accord : nous avons donc ajouté la règle 8 de *groupement par similarité*. L'ajout de cette règle nous permet par ailleurs d'uniformiser le comportement du niveau 0 au format audio (répétition d'échantillons similaires jusqu'à une rupture de similarité) avec les niveaux supérieurs.

1. <https://staff.aist.go.jp/m.goto/RWC-MDB/rwc-mdb-p.html>, visité le 19 octobre 2023

6.3.2 Analyse MIREX

Une piste est ensuite de comparer les segmentations obtenues au niveau supérieur avec les segmentations de référence proposées par Bimbot et al. Pour cela, on propose d'utiliser les tâches d'évaluation définies dans le cadre MIREX² (*Music Information Retrieval Evaluation eXchange*) définies par J.S Downie et al. [47].

Cinq mesures spécifiques nous intéressent afin de mesurer précisément la qualité de la segmentation obtenue :

- la *Precision* (mesurée à 0.5 secondes et à 3 secondes) ;
- le *Recall* (mesurée à 0.5 secondes et à 3 secondes) ;
- la *F-mesure* (mesurée à 0.5 secondes et à 3 secondes) ;
- la *Ref-to-est deviation* ;
- la *Est-to-ref deviation*.

Les mesures ont été calculées en utilisant la bibliothèque Python *mir_eval*³ avec la fonction `mir_eval.segment.evaluate(ref_intervals, ref_labels, est_intervals, est_labels)` qui prend en arguments les intervalles de segmentation et les labels associés de référence et estimés.

La précision correspond au nombre de segmentation estimées valides dans l'intervalle de temps considéré autour d'une segmentation de référence, divisé par le nombre total de segmentations. Une précision faible induit donc que de nombreuses segmentations sont éloignées des segmentations de référence, même si celles qui sont proches le sont de toutes les segmentations de référence. Cela peut être le cas par exemple lors d'une segmentation précise mais sur-segmentée.

Le recall correspond au nombre de segmentations estimées valides qui sont dans l'intervalle de temps considéré autour d'une segmentation de référence, divisé par le nombre total de segmentations de références. Un recall faible induit donc que peu de segmentation estimées sont alignées avec les segmentations de référence : si les segmentations estimées sont précises, cela peut avoir lieu par exemple dans le cas d'une sous-segmentation.

La mesure qui permet de combiner les deux est donc la F-mesure, notée *Fm*. Celle-ci se calcule de la manière suivante :

$$Fm = \frac{(precision * recall)}{(precision + recall)}$$

Dans la modélisation que nous proposons, chaque temps dure une seconde. Un

2. <http://www.music-ir.org/mirexwiki>

3. https://craffel.github.io/mir_eval/

fenêtre de 0,5' n'autorise donc aucune erreur, tandis qu'une fenêtre de 3' autorise une erreur de 3 temps.

La Ref-to-est deviation correspond au temps moyen qui sépare les segmentations de référence aux segmentations estimées les plus proches et la Est-to-ref deviation mesure le temps moyen qui sépare les segmentations estimées aux segmentations de référence.

Une difficulté cependant est de savoir quel niveau du diagramme formel hiérarchisé comparer avec les segmentations de référence à l'échelle de la section : nous avons gardé pour chaque diagramme formel hiérarchisé les valeurs du niveau pour lequel la F-mesure à 3.0' était maximale afin d'avoir un juste milieu entre la sur-segmentation et la sous-segmentation.

Nous avons effectué la même analyse sur l'ensemble de la base de donnée, avec les règles suivantes :

- règle de classification : alignement ;
- règle de segmentation : 1, 2, 5, 6 avec une contrainte de taille minimale de 3 éléments et maximale de 16 éléments, et 8.

Les valeurs moyennes obtenues sont les suivantes :

P@0.5	R@0.5	Fm@0.5	P@3.0	R@3.0	Fm@3.0	RE dev.	ER dev.
0.34	0.53	0.35	0.39	0.63	0.41	5.78	5.26

D'une part, on note que les valeurs à 0.5' correspondent aux segmentations qui tombent précisément au bon moment. Nous avons donc en moyenne une précision de 34% et un recall de 50%. Cela monte à une précision de 39% et un recall de 63% dans les 3 temps autour de la segmentation de référence, ce qui reste très peu.

On constate en moyenne que le nombre de temps d'écart entre la segmentation de référence et la segmentation estimée, et inversement, sont entre 5 et 6 temps. Par ailleurs, les fichiers MIDI synthétisés ont un tempo autour de 120bpm, ce qui correspond à deux temps par seconde. Cela signifie que l'état de l'art sur les fichiers audio a une précision d'analyse environ deux fois plus faible : nous proposons donc une deuxième analyse à deux temps par seconde, avec donc des fenêtres de temps deux fois plus grandes, soit une marge d'erreur d'1 temps (0.5') et 6 temps (3.0').

Les résultats obtenus sont présentés dans le tableau suivant :

P@0.5	R@0.5	Fm@0.5	P@3.0	R@3.0	Fm@3.0	RE dev.	ER dev.
0.35	0.43	0.35	0.53	0.63	0.54	2.69	2.59

La déviation moyenne est deux fois plus faible (2,69' et 2,59') pour un morceau

deux fois plus rapide, ce qui reste entre 5 et 6 temps et est cohérent. Elle peut différer légèrement car les niveaux retenus en fonction de la F-mesure ne sont pas toujours les mêmes pour chaque analyse, par rapport à l'étude précédente. On constate une F-mesure à 0.5' similaire, ce qui signifie que nous avons la même précision à 0 ou 1 temps près. En revanche, la F-mesure à 6 temps est de 54% et est considérablement meilleure que la précision à 3 temps de 0.41%.

Plusieurs facteurs peuvent expliquer ces résultats : tout d'abord, l'estimation automatique du niveau correspondant peut être inadaptée et fausser les résultats : cela se retranscrit par un fort déséquilibre entre la mesure de la précision et du recall (Pop 82 par exemple). D'autre part, lorsque l'on regarde le détail des résultats, nous constatons que certains morceaux présentent de très bons résultats (Pop 62 par exemple) , tandis que d'autres morceaux présentent des résultats très faibles (Pop 99 par exemple) : d'une part, les faibles résultats peuvent être dus à la robustesse du logiciel. En effet, un premier décalage de segmentation peut avoir un fort impact dans la suite de la segmentation du niveau actuel, ainsi que pour les niveaux supérieurs. Un autre constat est que les mêmes règles précisément ne sont pas forcément les plus adéquates pour l'ensemble de la base : par exemple, les contraintes choisies pour la taille du mot ne seront pas forcément les mêmes selon que le morceau soit binaire ou ternaire. Une suggestion est donc de subdiviser la base en un ensemble de règles pertinentes, ou de tester plusieurs règles sur la base et de garder les meilleurs résultats de chaque règle.

On présente dans le tableau suivant les résultats obtenus pour les exemples précis 52, 66 et 99 :

morceau	niveau	P@0.5	R@0.5	Fm@0.5	P@3.0	R@3.0	Fm@3.0	RE dev.	ER dev.
62	3	0.78	0.78	0.78	0.78	0.78	0.78	0.0	0.0
82	3	1.0	0.18	0.3	1.0	0.18	0.3	32.0	0.0
99	2	0.15	0.18	0.17	0.46	0.55	0.5	2.0	4.0

6.3.3 Études spécifiques des pièces RWC Pop 11 et RWC Pop 21

Nous choisissons de nous attarder sur deux exemples précis : l'exemple RWC Pop 11, et l'exemple RWC Pop 21.

Pour simplifier la lecture, nous avons ajouté des segments verticaux transversaux entre les niveaux afin d'indiquer la segmentation d'un niveau donné et l'instant correspondant aux niveaux inférieurs.

RWC Pop 11

La figure 6.21 présente le diagramme formel du morceau RWC Pop 11 obtenu avec les paramètres décrits plus haut. En gris sont indiqués les segmentations obtenues, et en bleu les segmentations de référence. Nous avons par ailleurs annoté par les lettres correspondantes les segmentations des sections de références, également présentées dans le tableau suivant :

I/2	16	Dm7	G7	Bb&9	FM7							
J1/C	32	Dm7	G7	Bb&9	FM7	Dm7	G7	Bb&9	FM7			
A	32	Dm7	G7	Bb&9	FM7	Dm7	G7	Bb&9	FM7			
B	32	BbM7	C7	A7d9*	Am7	D7s4	D7	BbM7	Am7	Bbs4	F	F+
C	32	Dm7	G7	Bb&9	FM7	Dm7	G7	Bb&9	FM7			
A'	32	Dm7	G7	Bb&9	FM7	Dm7	G7	Bb&9	FM7			
B	32	BbM7	C7	A7d9*	Am7	D7s4	D7	BbM7	Am7	Bbs4	F	F+
C	32	Dm7	G7	Bb&9	FM7	Dm7	G7	Bb&9	FM7			
I/2	16	Dm7	G7	Bb&9	FM7							
J2/C	32	Dm7	G7	Bb&9	FM7	Dm7	G7	Bb&9	FM7			
C	32	Dm7	G7	Bb&9	FM7	Dm7	G7	Bb&9	FM7			
C	32	Dm7	G7	Bb&9	FM7	Dm7	G7	Bb&9	FM7			
(5/4)J*/C	40	Dm7	G7	Bb&9	FM7	Dm7	G7	Bb&9	FM7	Bb&9	FM7	

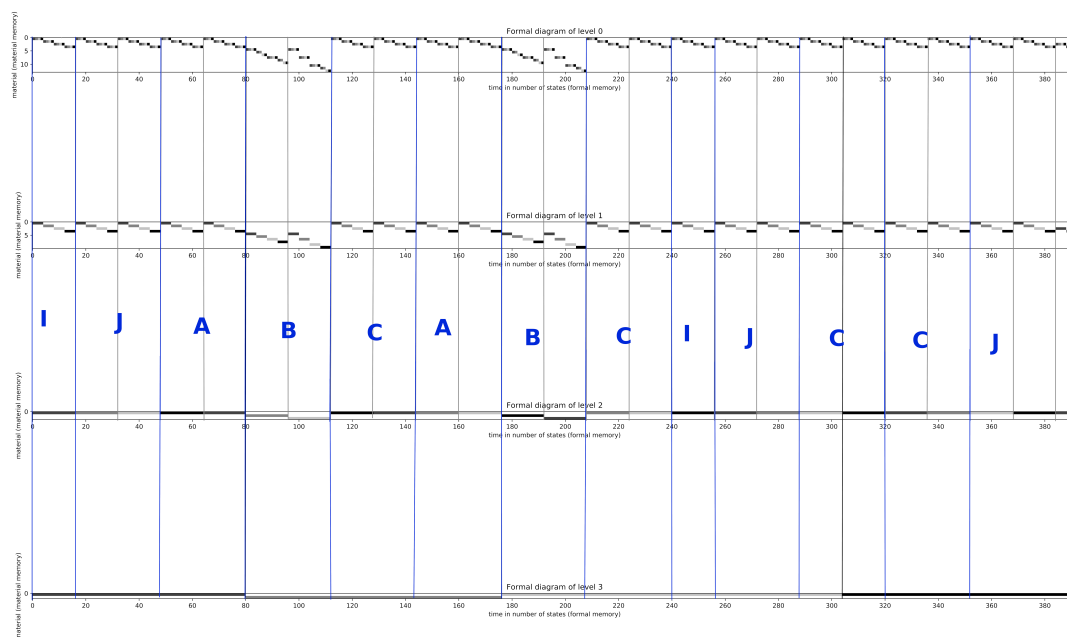


FIGURE 6.21 – Diagramme formel multi-échelle du morceau RWC Pop 11.

Nous constatons effectivement ici une segmentation précise, mais une sur-segmentation par rapport à l'échelle de la section au niveau 2 et une sous-segmentation par rapport à l'échelle de la section au niveau 3. Concernant la labélisation, un constat frappant est que la dimension étudiée pour l'analyse, à savoir l'harmonie, n'est pas suffisante pour distinguer les différentes sections annotées. En effet, la quasi intégralité de l'œuvre est constituée des 4 mêmes accords : c'est donc l'instrumentation, qui n'est pas étudiée ici, qui permet de déterminer la segmentation : ainsi, les segments I, J, a et C sont considérés comme identiques par l'algorithme et constitués des matériaux a, b, c et d des niveaux 0 et 1. Seule la section B, reconnue aux deux occurrences et constituée des matériaux $fghi$ $fhjk$ est distinguée. Ainsi, il est nécessaire ici de mettre ce diagramme formel hiérarchisé en perspective avec un autre diagramme formel sur une autre dimension telle que le rythme ou l'instrumentation.

RWC Pop 21

La figure 6.22 présente le diagramme formel du morceau RWC Pop 21 obtenu avec les mêmes paramètres. La segmentation estimée est reprécisée par des barres verticales grises (jusqu'au temps 208 au niveau et pour tout le morceau aux niveaux supérieurs) et la segmentation de référence est affichée en rouge. La segmentation de référence au niveau 1 correspond à celle de la mesure, et au niveau 2 à celle de la section obtenue par Bimbot et al. Nous avons également indiqué les noms des sections de référence prenant effet entre les deux segmentations rouges précédentes et suivantes.

Nous constatons que la segmentation au niveau 1 est quasiment la même, à deux types d'occurrences près : la première est au temps 19 : 7 objets successifs c sont regroupés entre eux au niveau 0, le premier matériau distinct et alors regroupé selon la règle de segmentation 8 dans le segment suivant. Un autre cas apparaît mesure 47 et est dû au retour de l'objet labélisé g , qui a déjà débuté un objet de niveau supérieur à la mesure 35. Une manière simple de corriger ces imprécisions ici serait d'imposer la métrique comme une contrainte forte et de paramétrer la règle de contrainte sur la taille des objets de telle sorte à ce que ceux-ci contiennent au minimum ou exactement 4 objets de niveau inférieur. Au niveau supérieur, la segmentation est moins précise, avec une sur-segmentation liée au retour du matériau A (temps 16, 408) ou un décalage allant jusqu'à 2 mesures (temps 92, 152, 212, 328).

Si l'on regarde plus spécifiquement les labels des sections, nous constatons cependant que les matériaux correspondants aux différentes sections sont reconnus

à chaque retour des sections. Ainsi, la section I correspond aux objets (AA) (temps 0 à 32), la section A correspond à l'objet (B) (temps 32 à 64 et 172 à 204), les sections B et C1 correspondent aux objets (CDE) (temps 64 à 116 et 204 à 252), et les sections C2 et C correspondent majoritairement à l'objet (F) (temps 116 à 144 et 352 à 392). Le matériau F ressemble en par ailleurs beaucoup aux matériaux (DE) dont la similarité n'est pas visible au niveau 2) avec une variation différente à la fin.

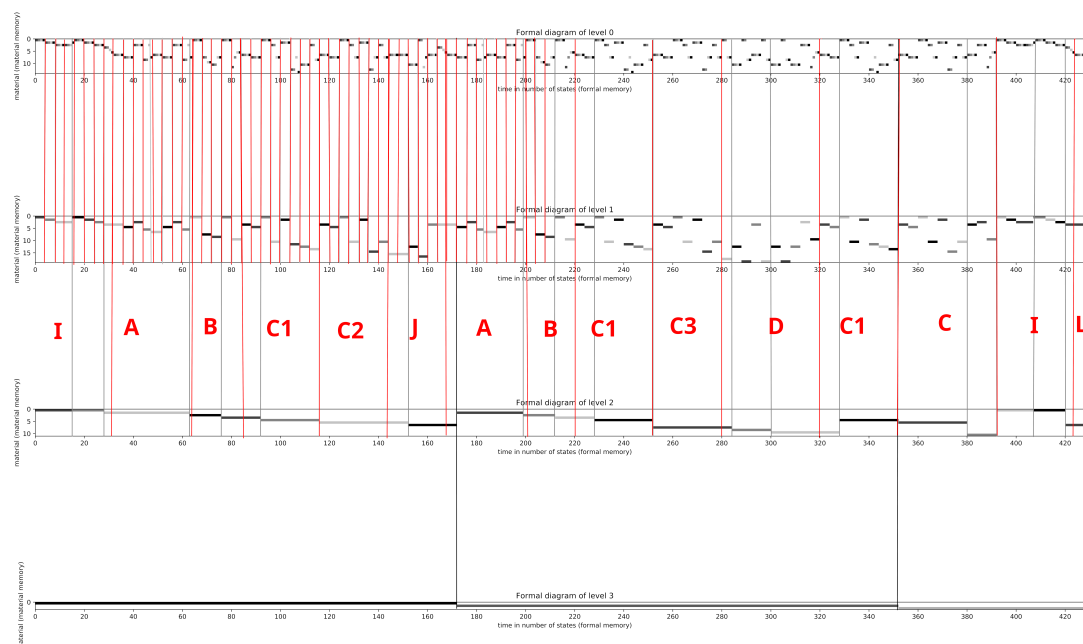


FIGURE 6.22 – Diagramme formel multi-échelle du morceau RWC Pop 21.

6.4 Analyses dans le cadre du cours *Outils numériques pour l'analyse musicale*

Dans le cadre d'un cours de Musicologie pour les enseignements de Master 1 à Sorbonne Université, trois étudiants ont eu l'occasion d'utiliser le logiciel afin de procéder à des analyses. Les trois œuvres étudiées sont :

- le Solo de cor anglais de l'acte III de *Tristan et Isolde*, Richard Wagner ;
- *Musik to heare*, Igor Stravinsky ;
- *L'Esperanza*, Sven Väth.

Pour cela, un premier prototype d'interface en tant que service web a été implémenté avec le framework Django⁴ et est hébergé sur les serveurs du labri⁵. Cette interface précède l'implémentation présentée au chapitre 5.

6.4.1 Présentation de l'interface prototypique

Cette interface est divisée en 3 pages web principales. L'interface, ainsi que l'ensemble des paramètres et l'algorithme en amont, a nécessité une présentation lors d'un cours magistral auprès des étudiants de musicologie. Une première remarque que nous pouvons faire est donc que l'utilisation de cette interface nécessite dans un premier temps une explication approfondie de la théorie sous-jacente.

Page de présentation du logiciel

Une première page (figure 6.23) présente un bref résumé de l'interface ainsi que son intérêt. On y rappelle que MORFOS propose une représentation dynamique de la structure musicale que l'on appelle forme, qui cherche à appréhender autrement les flux de données, notamment numérique. On précise par ailleurs qu'il s'agit ici d'un modèle de l'écoute, qui ne se veut pas réaliste, il s'agit bien d'un modèle, et nous constatons nos observations à partir de ce modèle donné.

Ensuite, nous précisons que cette interface web n'est qu'un prototype en attendant l'interface qui précède l'interface présentée au chapitre 5. Cette interface web se contente de la superposition des représentations en 2 dimensions, et l'interface n'est pas spécialement ni ergonomique ni optimisée pour l'utilisateur. Enfin, la définition des diagrammes formels est présentée, ainsi que celle des objets et des matériaux.

Page de paramétrisation

Une deuxième page présentée en figure 6.24 est la page de paramétrisation de l'algorithme.

Nous proposons à gauche un formulaire à remplir par l'utilisateur et à droite le détail des explications pour chacun des paramètres, avec, idéalement, l'influence en terme d'analyse

4. <https://www.djangoproject.com/> visité le 02/10/23

5. <http://algocog.labri.prod.iaas.u-bordeaux.fr/algocognitif-web-interface/> visité le 02/10/23

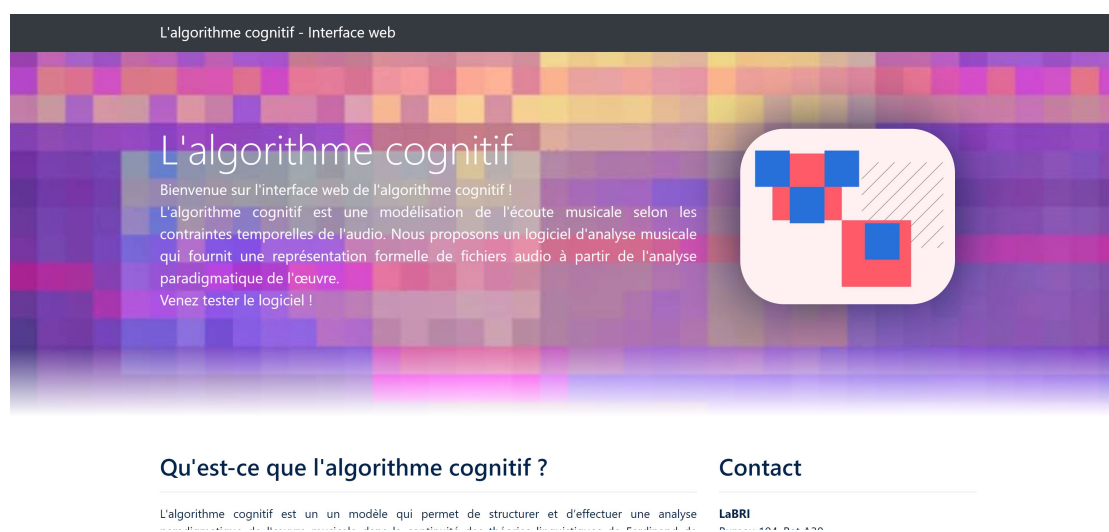


FIGURE 6.23 – Page de présentation de l'interface prototypique.

Les paramètres proposés sont les suivants :

- le type de descripteur pour l'analyse : *fft*, *cqt* ou *mfcc*
- le seuil de similarité et le seuil de segmentation
- les règles de similarité pour le signal (concordance différentielle ou distance euclidienne)
- les règles de similarité pour le symbolique (égalité stricte ou alignement)
- les règles de segmentation pour le signal (transformée de fourier différentielle ou transformée de fourier dynamique)
- les règles de segmentation pour le symbolique (les règles ont été renommées et correspondent aux règles 1,2,3, 4 et 6, seulement la règle 1 validait la segmentation au retour de n'importe quel élément similaire même s'il ne correspondait pas au premier élément d'un facteur précédent)

Un autre paramètre expérimental a été ajouté ici : il s'agit d'un paramètre d'affichage de la polyphonie tel que présenté dans la première section de ce chapitre.

Des paramètres par défaut sont proposés pour faciliter l'intuition de l'utilisateur.

Une fois les paramètres sélectionnés par l'utilisateur, une page de confirmation résume les paramètres sélectionnés et demande confirmation à l'utilisateur.

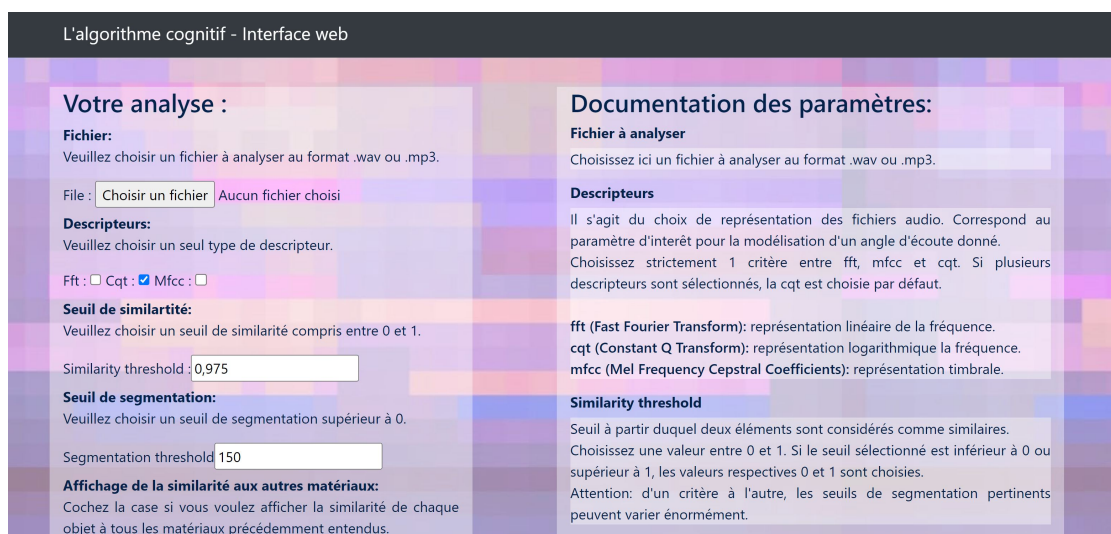


FIGURE 6.24 – Page de paramétrisation de l'interface prototypique.

Résultats

Une dernière page affiche ensuite les résultats en superposant les diagrammes formels des différents niveaux obtenus.

Le temps d'analyse sur l'interface web est malheureusement très long. Le logiciel à l'époque n'était d'une part pas optimisé pour les œuvres de quelques minutes, l'utilisation en concurrentiel de l'interface et la potentielle bridation des serveurs alloués sont un frein pour l'utilisation de cette interface. Nous avons également pu nous rendre compte de l'importance d'une barre de chargement pour tenir au courant l'utilisateur de l'état de l'algorithme. Une fois les résultats chargés, les différents diagrammes formels s'affichent à l'écran en deux dimensions.

6.4.2 Solo de cor anglais de l'acte III de *Tristan et Isolde*, Richard Wagner

Rémi Lacombe, en M1 Musicologie à Sorbonne Université propose une analyse du solo de cor anglais de l'acte III de *Tristan et Isolde* composé par Richard Wagner. Ses commentaires sont disponibles également sous forme de blog⁶.

6. voir <https://outilsnumeriquesmusicologie.wordpress.com/2022/01/13/lalgorithme-cognitif-a-lepreuve-de-la-melodie-du-berger-application-au-solo-de-cor-anglais-de-lacte-iii-de>

Le choix de l'œuvre est motivé par les nombreuses analyses préexistantes (voir J.J Nattiez [113]). Par ailleurs, un autre des intérêts de l'œuvre est que les paradigmes de cet extrait se basent sur des hauteurs relatives et des spécificités rythmiques, qui permettent alors d'étudier les limites du logiciel.

Ainsi, R. Lacombe présente dans un premier temps l'analyse effectuée par F. Lerdhal et présentée par J.J Nattiez dans [113], que l'on retrouve en figure 6.25.



Exemple 40. Jean-Jacques Nattiez, analyse paradigmatique des longues unités du solo de cor anglais, d'après Fred Lerdahl*
* Lerdahl, « Prolongational Structure », figure 10, p. 37.

FIGURE 6.25 – Analyse paradigmatique du solo par F. Lerdhal, présentée par J.J. Nattiez dans [113]).

R. Lacombe souligne 3 paradigmes identifiés : un premier paradigme correspondant à une phrase mélodique et deux paradigmes constitués d'un triolet et d'une note tenue pour le premier et d'une série de triolets pour le second.

Comparaison entre le fichier audio et le fichier midi

Tout d'abord, R. Lacombe propose une comparaison de l'analyse à partir d'un fichier MIDI synthétisé, et un enregistrement audio d'une performance⁷. Les paramètres de pré-traitement et de tests de similarité et segmentations ont été laissés par défaut, et des tests ont été effectués avec 7 couples de seuils de similarité et de segmentation différents.

On présente en figure 6.26 les résultats pour une analyse audio (haut) et MIDI (bas) avec un seuil de similarité de 0.975 et un seuil de segmentation de 150.

tristan-et-isolde/, visite du 02/10/23

7. ref de la performance

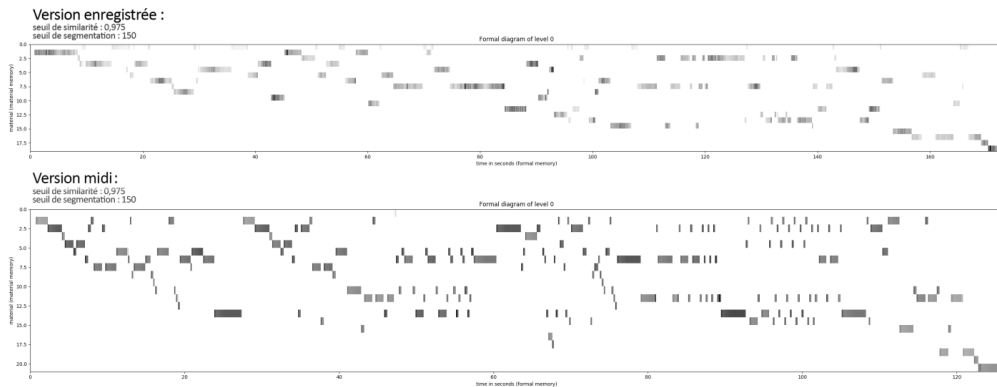


FIGURE 6.26 – Comparaison des diagrammes de niveau 0 du solo de cor obtenus pour la version enregistrée (haut) et la version midi (bas).

Bien que la forme soit reconnue, on constate la création d'un plus grand nombre de matériaux avec la version MIDI. En effet R. Lacombe commente : « Cette différence s'explique naturellement par la complexité des paramètres qui interviennent dans un enregistrement : variations du timbre, inflexions agogiques et dynamiques de l'instrumentiste, variabilité des attaques, etc. Ceux-ci rendent à l'évidence la tâche de l'algorithme moins aisée quand il s'agit d'identifier des hauteurs individuelles. » En effet, une œuvre performée induit de nombreux paramètres supplémentaires liés à la physique de l'instrument ou encore à la performance musicale, et qui sont lissés dans un format MIDI. Néanmoins, il ajoute que bien que le format MIDI permette de lever certaines ambiguïtés, il est toujours préférable de proposer une analyse sur des enregistrements de performances réelles dans le cadre notamment d'une modélisation de l'écoute, afin de prendre en compte tous les facteurs de l'audio.

Comparaison des résultats MIDI avec l'analyse paradigmatique de F. Lerdahl

Le résultat obtenu avec les paramètres précédents au niveau 4 de l'analyse est comparé avec l'analyse de F.Lerdahl en figure 6.27.

Les 4 phrases du solo sont à peu près retrouvées, strictement pour la première et approximativement pour les phrases trois et quatre. R. Lacombe interprète alors que *"ces résultats traduisent l'impression diffuse d'une organisation en quatre temps, et que le caractère quelque peu arbitraire des délimitations formelles reflète une*

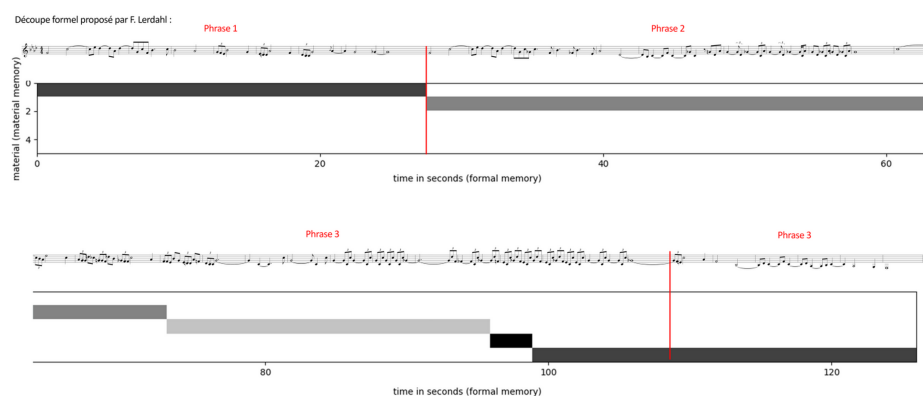


FIGURE 6.27 – Comparaison entre le diagramme formel de niveau 4 du solo de cor obtenu pour la version midi (seuil de similarité de 0,97) et le découpage formel proposé par Lerdahl.

ambiguïté effectivement présente au niveau perceptif"

Au niveau inférieur 3 (figure 6.28) avec ces mêmes paramètres, seuls deux motifs ont été retrouvés de manière récurrente : les répétitions étant pour la plupart transposées, cela démontre ainsi la limite du choix du descripteur pour l'analyse effectuée, qui ne prend pas en compte les transpositions, ni la répétition des cellules rythmiques.

En diminuant le seuil de similarité à 0,965, trois paradigmes sont retrouvés et sont recensés en figure 6.29. On constate alors par ailleurs la forte sensibilité du seuil de similarité.

6.4.3 *Musik to heare*, Igor Stravinsky

Elisa Constable, également étudiante en M1 à Sorbonne Université propose une analyse de l'œuvre *Musik to heare* composée par Igor Stravinsky⁸. Il s'agit d'une œuvre sérielle composée autour des quatre notes *si, sol, la, sib*. Ici spécifiquement, E. Constable s'est intéressée à la partie vocale de l'œuvre. L'analyse a été effectuée avec les paramètres par défaut du logiciel. Au premier niveau, représenté en figure 6.30, tous les matériaux sont considérés comme similaire. E. Constable aurait pu alors augmenter le seuil de similarité pour découvrir des matériaux distincts.

8. <https://outilsnumeriquesmusicologie.wordpress.com/2022/01/09/de-linteret-dun-logiciel-de-modelisation-de-lalgorithme-cognitif-pour-lanalyse-musicale/>, visité le 02/10/23

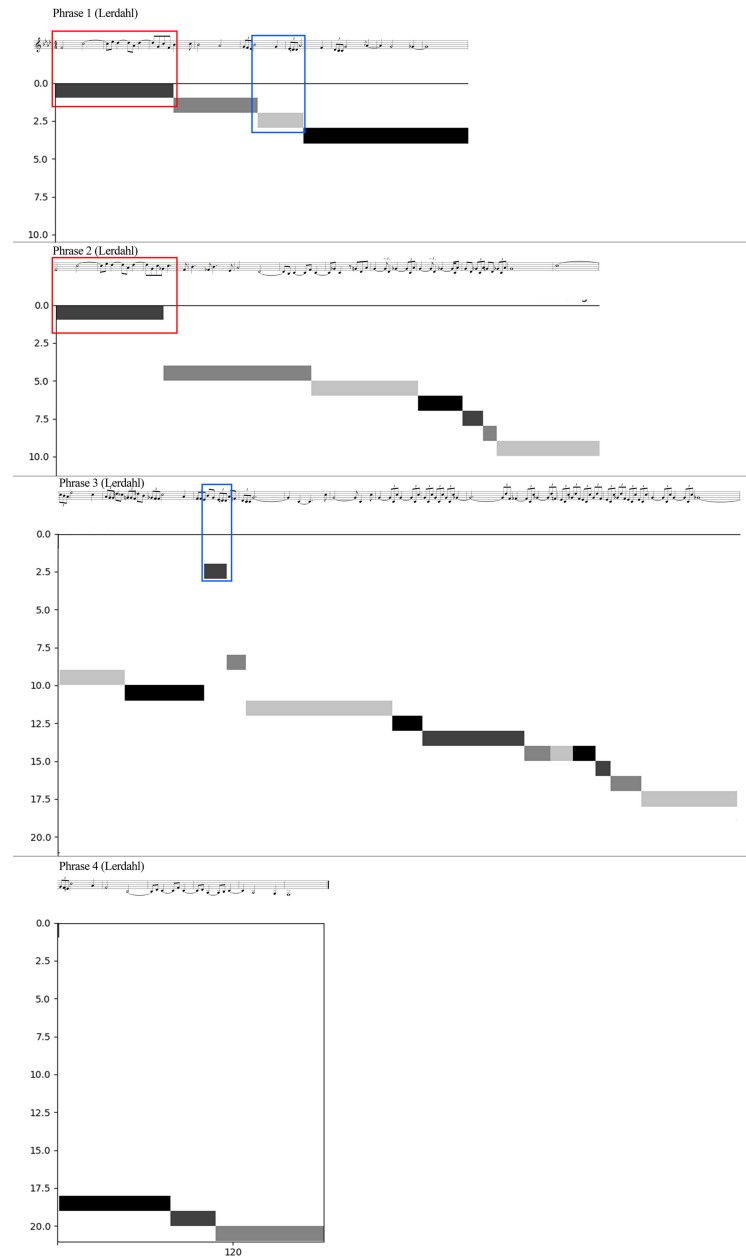


FIGURE 6.28 – Comparaison entre le diagramme formel de niveau 3 du solo de cor obtenu pour la version midi (seuil de similarité de 0,97) et le découpage formel proposé par Lerdahl. Les deux principales récurrences paradigmatiques identifiées par l'algorithme ont été encadrées en rouge et en bleu.

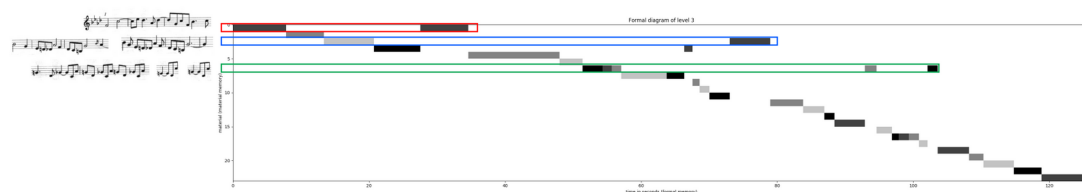


FIGURE 6.29 – diagramme de niveau 3 obtenu pour la version MIDI (seuil de similarité de 0,965) et fragments musicaux correspondant aux différentes occurrences des trois principaux paradigmes.

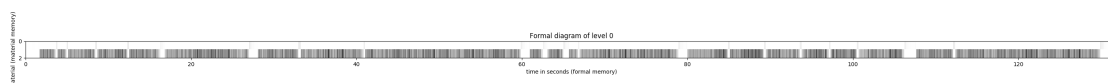


FIGURE 6.30 – Diagramme formel de *Musik to heare* de niveau 0 obtenu avec le logiciel.

E. Constable propose en figure 6.31 une analyse de ce premier niveau bien plus développée. Elle commente "*Ce tableau que nous avons réalisé à partir d'une analyse poétique de la partition et de sa composition est purement théorique et ne suffit pas à une analyse complète. Aussi satisfaisant et pertinent que cela soit pour l'analyste, cela ne correspond certainement pas à la réalité de l'écoute, l'auditeur n'étant certainement pas capable de reconnaître instantanément tel rétrograde ou telle transposition. Et c'est bien ce que ce premier diagramme produit par le logiciel nous démontre. L'analyse purement sérielle risque de n'être qu'une vue de l'esprit (certes, celui du compositeur), si on ne la met pas en perspective avec des questions d'ordres temporel et esthétique*". En effet, le diagramme formel obtenu au niveau zero pourrait être considéré comme une incapacité, d'un point de vue auditif, à distinguer les différents matériaux qui constituent l'œuvre.

Au niveau 1, présenté en figure 6.32, le logiciel détermine 4 occurrences du matériau 0, qui correspondent en réalité au début de chaque nouvelle strophe mesures 9, 22 (40s.), 25 (80s.), et 44 (105s.). On constate cependant une mécompréhension par E. Constable concernant la couleur des objets dans le diagramme, qu'elle assimile à une valeur de similarité alors qu'il ne s'agit que d'une rotation de couleurs par défaut afin de distinguer les objets successifs. C'est cette ambiguïté qui nous a poussé à utiliser finalement une représentation avec des objets ayant toujours la même couleur (quitte à effectivement utiliser les niveaux de gris pour représenter la

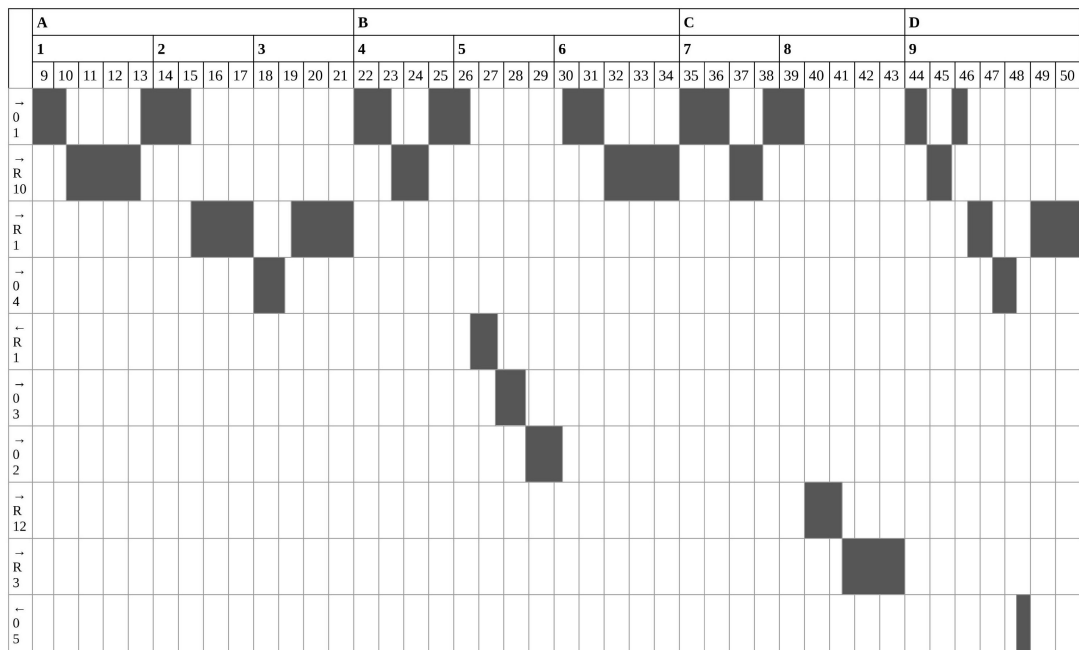


FIGURE 6.31 – Diagramme formel de *Musik to heare* de niveau 0 obtenu manuellement par E. Constable.

valeur de similarité à l'avenir), et des frontières séparant les objets. Le diagramme formel obtenu est cette fois-ci relativement similaire à l'analyse formelle proposée par E. Constable en figure 6.33. Cependant, si chaque occurrence du matériau 0 est à peu près retrouvée, les matériaux qui les succèdent ne sont pas distingués dans le diagramme formel figure 6.32, contrairement à l'analyse formelle figure 6.33.

Le niveau 2 du logiciel en figure 6.34 permet de décrire un découpage global en 2 de l'œuvre avec le retour au matériau 0 : les matériaux intermédiaires ne sont quant à eux pas vraiment précis mais indiquent la répétition par deux fois des mêmes matériaux, et le niveau 3 présenté en figure 6.35 enfin décrit les deux parties répétées et similaires et un matériau final différencié.

Ainsi, la forme générale obtenue par le logiciel est résumée par E. Constable dans la figure 6.36.

Le niveau 1 décrit ainsi une œuvre prenant « *une forme rondeau ou une forme*

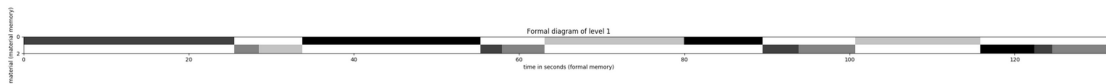


FIGURE 6.32 – Diagramme formel de *Musik to heare* de niveau 1 obtenu avec le logiciel.

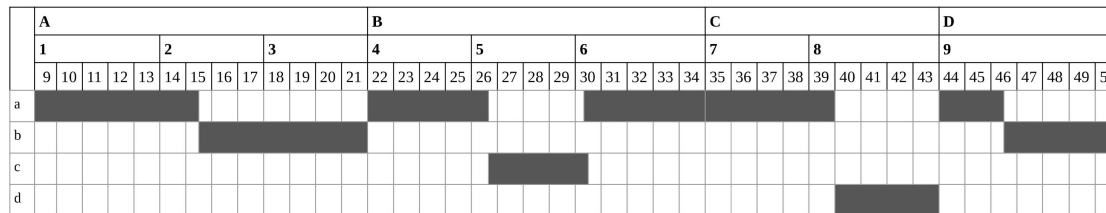


FIGURE 6.33 – Diagramme formel de *Musik to heare* de niveau 1 obtenu manuellement par E. Constable.

à refrain, correspondant à l'exploitation de chaque strophe du poème de Shakespeare ». Cependant, E. Constable souligne que l'analyse logique du logiciel n'est pas à confondre avec la construction sociale et donc la connaissance culturelle d'un musicologue : cette forme nécessite ainsi l'interprétation par un spécialiste dans un second temps. Dans cette œuvre par exemple, les différentes occurrences de A sont initiées par ce que E. Constable nomme en reprenant le terme de M. Imberty des « vecteurs dynamiques », qui sont « de saillance dynamique, des déclencheurs de la construction cognitive et émotionnelle de la forme temporelle de l'œuvre musicale dans son déploiement linéaire » [72] et qui sont déterminés par la règle 1 de retour à l'objet de niveau actuel présentée au chapitre 3. Mais les différentes occurrences de 1 relèvent en réalité de différentes versions noyées dans un jeu rythmique assez peu reconnaissable.

6.4.4 L'Esperanza, Sven Väth

La dernière étude effectuée dans le cadre du M1 musicologie de Sorbonne Université est effectuée par R. Dardel⁹ sur la musique électronique polyphonique *L'esperanza*(1'20) du DJ et producteur Sven Väth.

9. <https://outilsnumeriquesmusicologie.wordpress.com/2021/12/13/interet-de-lanalyse-algorithmique-dans-la-musique-electronique/> visité le 02/10/23

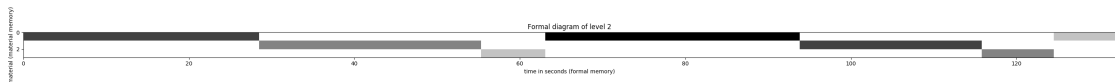


FIGURE 6.34 – Diagramme formel de *Musik to heare* de niveau 2 obtenu avec le logiciel.

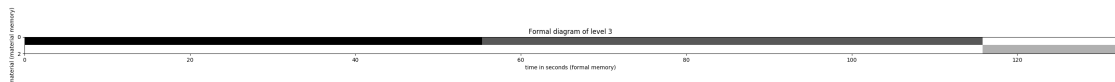


FIGURE 6.35 – Diagramme formel de *Musik to heare* de niveau 3 obtenu avec le logiciel.

level 1	A	B	A	C	A	D	A	E
level 2	A	B	C	A'	B'	C'	C	
level 3		A			A'		B	

FIGURE 6.36 – Résumé de la structure obtenue avec le logiciel.

Pour cette analyse, l'étudiant s'est confronté à une difficulté à trouver les paramètres pertinents pour permettre une analyse adéquate ainsi qu'au temps de calcul pour l'analyse de l'œuvre.

On trouve ainsi en figure 6.37 les diagrammes obtenus.

De manière générale, on constate ici qu'une œuvre telle que celle-ci composée en strates musicales n'est pas un format bien géré par l'algorithme. Néanmoins, en augmentant le seuil de similarité, nous devrions trouver de nouveaux matériaux lors de l'apparition de nouvelles strates musicales.

6.4.5 Bilan

Les retours des étudiants sont donc les suivants. Tout d'abord, ils soulignent le fort potentiel du logiciel lié à la mise à disposition de nombreux descripteurs (R. La-

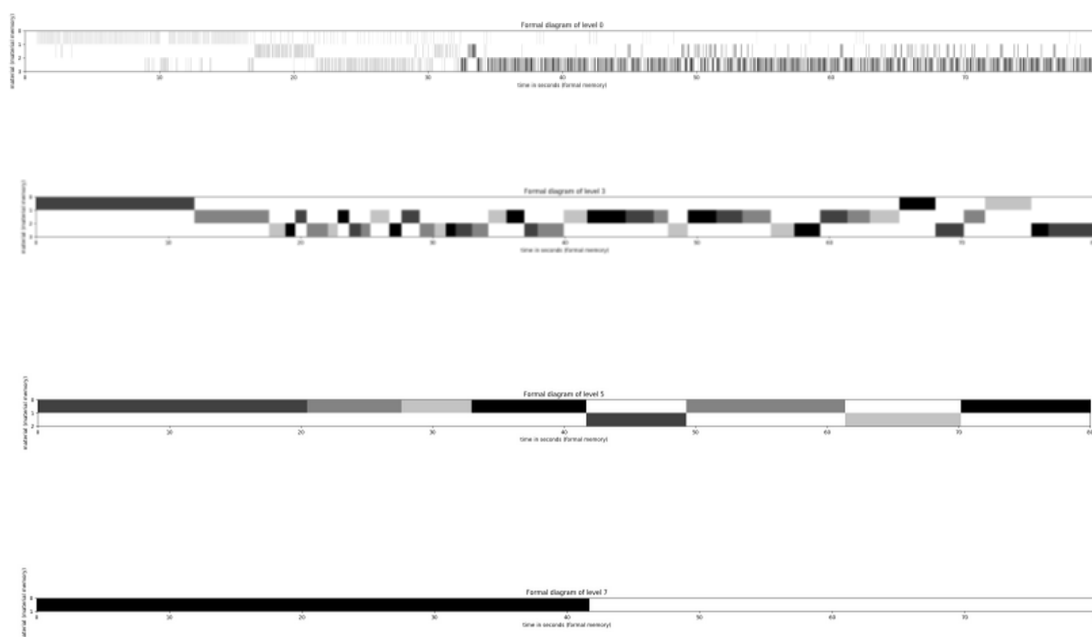


FIGURE 6.37 – Diagrammes formels de l'*Esperanza* obtenu avec le logiciel. De haut en bas : niveau 0, 3, 5 et 7.

combe), et la potentielle utilité concernant la vérification d'analyse (E. Constable). Néanmoins, un retour négatif principal a porté sur le temps de calcul de l'interface, aggravé par l'hébergement de l'interface et la gestion non parallélisée des calculs, augmentant donc drastiquement le temps de calcul lorsque plusieurs analyses sont lancées en simultané. Par ailleurs, il manque la prise en compte de la polyphonie (Lacombe, Constable) et il est également souligné le problème du choix de la mesure absolue de la hauteur et non pas intervallique. R. Lacombe explique D. Huron en citant : « la nature des représentations mentales de la mélodie reste incertaine, il ne fait nul doute que les paramètres du contour mélodique, des intervalles et du rythme sont d'un point de vue perceptif plus pertinents qu'une représentation absolue de la hauteur » [69]. Enfin, les valeurs et les choix de certains paramètres ne semblaient pas intuitifs pour les étudiants (Lacombe, Dardel) et de manière générale nous constatons qu'il y a une difficulté à les maîtriser, et que cela nécessite actuellement un usage empirique important de l'interface.

Conclusion

Dans ce chapitre, nous avons présenté différentes applications de MORFOS à travers des études d'œuvres diverses.

Des premières analyses sur les exemples simples du *Geisslerlied Maria Muoter Reinu Mait* nous ont permis d'étudier l'influence des différents paramètres de pré-traitement du signal ainsi que du seuil de similarité. Nous avons ainsi pu constater que ces paramètres sont très sensibles et qu'une légère variation peut fournir des représentations très différentes mais cependant représentatives de différents types d'écoute. C'est pourquoi nous avons décidé de laisser le choix de ces paramètres à l'utilisateur de l'interface. Nous avons relevé par ailleurs l'importance d'une analyse automatisée à partir d'une représentation audio avec l'étude du *Rondo du Menuet K.545* de W.A. Mozart au niveau 0, selon deux fichiers distincts : nous avons pu constater que l'interprétation et le nombre de paramètres impliqués dans la qualité du son permet de mettre en avant des informations très distinctes pour une même œuvre. Ensuite, une analyse à partir d'une représentation symbolique de ce même menuet nous a permis de déterminer un ensemble de règles grammaticales pour la segmentation et qui nous permettent de retrouver la forme attendue du *Rondo*. D'autres exemples, avec d'autres sélections de règles de segmentation sont présentés aux chapitres 3 et 4.

Nous avons également calculé les diagrammes formels multi-échelles sur l'ensemble de la base de données RWC Pop avec un agenda unifié pour toutes les analyses. Nous avons ensuite comparé les segmentations obtenues avec les segmentations de références proposées par Bimbot et al. [16]. Nous avons constaté, de part l'hétérogénéité de résultats obtenus, d'une part qu'un même Agenda s'applique difficilement à l'ensemble de la base de données. Par ailleurs, les niveaux de segmentations avec MORFOS sont construits automatiquement en fonction de la concaténation des objets de niveau inférieur et ne sont pas inférés par l'utilisateur, ce qui fait que les niveaux construits ne correspondent pas nécessairement à ceux de référence : il y a effectivement soit sous-segmentation, soit sur-segmentation dans les résultats obtenus avec MORFOS. Cependant, il semblerait pertinent de retrouver le niveau de segmentation proposé dans les données de référence.

Par ailleurs, plusieurs études ont été effectuées avec des étudiants en Musicologie : tout d'abord, des analyses ont été en collaboration avec avec Elia Bichoy, dont nous présentons deux analyses sur la *Passion selon Saint Matthieu BWV.244*, J.S. Bach ainsi que le *Concerto pour violon en la mineur RV356* de A. Vivaldi. Nous avons dans le cadre de ces études pu échanger sur les effets et les choix des différents paramètres de l'Agenda. Trois autres études ont été effectuées par des étudiants

en Master 1 de Musicologie de Sorbonne-Université qui ont utilisé une interface développée pour l'occasion. Les trois œuvres étudiées sont le solo de cor anglais de l'acte III de *Tristan et Isolde* de Richard Wagner, *Musik to heare* de Igor Stravinsky ainsi que *L'Esperanza* de Sven Väth. Sur l'ensemble de ces analyses, nous constatons des résultats variés qui dépendent notamment des contraintes de similarité de l'œuvre analysée et des règles de segmentation. La gestion des transpositions par exemple n'est pas bien gérée, et un décalage de segmentation peut induire un décalage sur toute la suite de l'œuvre à cause des règles de segmentation 1 et 2 qui segmentent lors de la reconnaissance d'un objet de niveau supérieur ou au retour du début d'un objet de niveau supérieur. Par ailleurs, nous avons pu constater une difficulté à manier les paramètres, qui demandent une étude empirique importante par l'utilisateur. Néanmoins, après une paramétrisation pertinente, nous parvenons sur beaucoup d'œuvres à retrouver les formes attendues et à mettre en lumière des phénomènes morphologiques, ou l'absence d'explicitation de ces phénomènes.

Conclusion et Perspectives

Cette thèse interdisciplinaire est au croisement de la musicologie et de l'informatique et implémente une méthodologie qui puise ses sources dans la sémiologie et les sciences cognitives. On y présente l'automatisation de l'Algorithme Cognitif présenté par Jean-Marc Chouvel dans [33] par l'implémentation d'un logiciel intitulé MORFOS, pour *Multiscale Oracle Representations For Organized Sounds*. Ainsi, nous posons de nombreuses bases pour la formalisation, l'implémentation, la visualisation, l'analyse informatique et musicologique de ce logiciel.

Résumé

Nous présentons un modèle informatique pour le logiciel, basé sur l'Algorithme MORFOS qui est un algorithme récursif inspiré des modèles cognitifs de l'écoute musicale. Nous formalisons les notions de *matériau*, *objet* et *diagramme formel* présentés par Chouvel et nous introduisons la notion de *Diagramme Formel Multi-échelles*, qui est l'ensemble organisé des différents diagrammes formels obtenus à chacun des niveaux d'analyse de l'œuvre analysée. De cette notion émerge une organisation en trois plans que sont les plans de la *forme*, de la *structure* et de l'*organisation*, qui correspondent à trois sous-représentations distinctes de l'information contenue dans le Diagramme Formel Multi-échelles.

MORFOS prend en entrée des représentations audio, symboliques ou vectorielles et est implémenté de manière modulaire. Il présente quatre modules principaux : l'algorithme MORFOS, les structures de données, la visualisation du diagramme formel multi-échelle et l'interface des paramètres. Trois autres modules indépendants, le pré-traitement des données, les règles de segmentation et les règles de similarité, sont également présentés. L'ensemble des mesures de similarité et de segmentation sont présentées exhaustivement dans ce manuscrit : les règles de segmentation sont divisées en trois types de règles obligatoires (qui forcent la seg-

mentation), interdites (qui interdisent la segmentation) ou bivalente (qui possèdent des contraintes forçant la segmentation et d'autres contraintes l'interdisant). Une étude particulière a été effectuée concernant la détermination de règles de grammaire pour la segmentation symbolique, et l'implémentation d'un gestionnaire de règles est présenté pour aller vers une flexibilité de l'ajout des règles dans le logiciel.

Nous proposons ensuite de comparer le comportement du logiciel MORFOS avec les fonctions cognitives impliquées lors de l'écoute musicale active. Trois principaux phénomènes cognitifs sont distingués : Les opérations de classification et de segmentation sont comparées aux actions d'écoute et à la représentation en mémoire de l'œuvre au niveau cognitif. Au niveau du logiciel, elles se manifestent par l'obtention du diagramme formel multi-échelle. L'analyse du comportement et les phénomènes d'attention correspondent aux opérations cognitives telles que l'acquisition, l'attente, et la formation d'hypothèses. Au niveau logiciel, ces opérations sont modélisées par l'étude du comportement algorithmique. Le calcul des coûts liés aux phénomènes d'attention, représente la charge cognitive associée à l'écoute musicale et à la charge computationnelle des opérations de MORFOS. Nous proposons ensuite de mesurer la pertinence de la segmentation en fonction de la complexité de Kolmogorov en utilisant deux méthodes distinctes : La première méthode compte le nombre d'opérations nécessaires pour décrire un objet acquis par rapport à son contexte. La deuxième méthode utilise la similarité entre l'objet acquis et son contexte pour mesurer la complexité. Nous proposons également d'automatiser l'étude du comportement de l'Algorithme Cognitif avec l'implémentation du calcul automatique des phases cognitives présentées par J.M Chouvel, que nous pouvons mettre en perspective avec les calculs de complexité, et nous proposons une comparaison de MORFOS avec le modèle d'écoute par EEG proposé par S. Koelsch et W. A. Siebel. Nous décrivons par ailleurs une expérience à double tâche pour évaluer le comportement des individus lors de la tâche de segmentation musicale. Les résultats préliminaires sont difficiles à interpréter, mais des analyses plus détaillées sont prévues pour comparer les comportements des individus à ceux de MORFOS.

Nous présentons également trois visualisations des diagrammes formels multi-échelles pour une meilleure compréhension des plans de la forme, de la structure et de l'organisation. La première est en 2D et aligne les diagrammes formels des niveaux sur la hauteur pour représenter le temps. Cette visualisation est claire pour le plan de la forme mais moins efficace pour les plans de la structure et de l'organisation. La deuxième visualisation est en 3D et les objets sont représentés par des pavés, leur hauteur dépendant du niveau, leur longueur de la durée, et leur profondeur est constante. Cette vue offre une représentation de la structure, bien que la vue matériaux/niveaux soit moins utile. La troisième visualisation est

en 3D, représentant les objets par des sphères de diamètre constant. Les coordonnées des sphères dépendent du temps, de la durée du matériau, et de la date d'apparition du matériau. Des liens connectent les objets de niveau supérieur à ceux de niveau inférieur, explicitant les relations entre les matériaux des différents niveaux. On peut alors imaginer représenter la mémoire et les phénomènes d'attention en ajustant par exemple l'opacité et la couleur des matériaux. Les hypothèses sur les matériaux futurs pourraient alors être représentées par une couleur spécifique, changeant selon que l'hypothèse est validée ou invalidée. Pour afficher ces visualisations, une interface en Electron, encore incomplète à ce jour (notamment sur la précision de l'agenda) a été développée, offrant également la possibilité de sauvegarder et de partager des projets. L'objectif futur de cette interface est de permettre par ailleurs une visualisation dynamique de la représentation.

Enfin, un certain nombre d'analyses effectuées avec MORFOS sont présentées, d'une part pour étudier le comportement du logiciel, mais également dans un cadre applicatif avec des étudiants de musicologie. Ces analyses montrent des résultats variés, nécessitant une paramétrisation minutieuse et on constate que certaines d'œuvres ne rentrent pas encore dans le cadre de l'analyse par MORFOS. Cependant, avec les bons paramètres, MORFOS révèle sur un bon nombre d'œuvres à minima des prémices, si ce n'est les structures attendues et met en avant des phénomènes morphologiques.

Perspectives

De nombreuses pistes ont été abordées au cours des 6 chapitres de ce manuscrit et méritent d'être individuellement approfondies et validées. Les différentes visualisations proposées restent à être validées auprès d'utilisateurs et l'interface doit être complétée car elle ne propose que la visualisation sphérique et un ensemble de paramètres limité pour l'Agenda. Les analyses des différentes études pour le calcul de la complexité et de la charge cognitive doivent être également approfondies. Un grand changement a été opéré avec l'ajout du signal à tous les niveaux de hiérarchie avec la mise à niveau 2.0 de MORFOS et des tests et analyses doivent être effectués en conséquence : nous espérons ainsi pouvoir améliorer la segmentation aux niveaux supérieurs mais également travailler sur la classification aux niveaux supérieurs avec les descripteurs audio. Néanmoins, les aspects temporel pour le rythme et transitionnel pour la transposition manquent encore et doivent être modélisés dans le logiciel, ainsi que la polyphonie dont nous apportons une piste de recherche à partir de fichiers multi-pistes dans la section 6.1.

Nous nous sommes limité dans ce manuscrit principalement à une utilisation du logiciel pour l'analyse de la forme, mais la découverte des 3 plans de la représentation ouvre de nouvelles perspectives à explorer pour l'analyse. Néanmoins nous pourrions imaginer que les représentations obtenues servent de socle à la composition. En effet, il est possible de jouer entre la forme et le fond, en gardant les représentations obtenues et en changeant les matériaux associés par exemple, ou bien en gardant un plan fixe et en jouant avec la modification des deux autres plans de la représentation. Peut-être que d'autres processus compositionnels peuvent être expérimentés en jouant sur l'agenda et les niveaux d'analyse par exemple.

Une autre perspective intéressante est l'apport d'une représentation de référence pour la synthèse sonore, lors de performances d'improvisation par exemple. Avec l'implémentation de l'oracle dans le logiciel, il devient possible d'inférer la structure d'une œuvre analysée par exemple, et la table des liens, qui permet la communication entre les oracles des différents niveaux permettrait d'induire les changement d'états autorisés.

Ainsi, cette thèse, qui représente les premiers pas de l'automatisation de l'Algorithme Cognitif avec le développement du logiciel MORFOS, ouvre nous l'espérons de nombreuses portes qui sont encore à explorer sur l'aspect pédagogique, analytique, compositionnel et performatif de la forme musicale.

Annexe A

Formalisation temps réel : exemple du *Rondo* de W.A. Mozart

Nous présentons dans cette annexe l'exemple complet de la formalisation temps réelle du *Rondo* de W.A. Mozart.

Sur l'exemple du *Rondo* de W.A. Mozart, nous avons les changements d'états suivants :

Le diagramme formel multi-échelle est initialement représenté par le mot vide ϵ . On souhaite acquérir un nouvel élément au niveau 0. Pour cela, on doit d'abord créer le niveau 0 avec l'opération *cnn* :

$$\epsilon \xrightarrow{cnn} X_0$$

On acquiert ensuite un nouvel élément a au niveau 0 :

$$(r^{(0)}, s^{(0)}, k^{(0)}) = (\epsilon, \epsilon, 0) \xrightarrow[a]{atr} (r^{(0)}.a, s^{(0)}, k^{(0)}) = (a, \epsilon, 0)$$

Il n'y a pas lieu de segmenter, on acquiert donc l'élément suivant, b :

$$(a, \epsilon, 0) \xrightarrow[b]{atr} (ab, \epsilon, 0)$$

De la même manière, on continue d'acquérir l'élément suivant, a :

$$(ab, \epsilon, 0) \xrightarrow[a]{atr} (aba, \epsilon, 0)$$

Ici, on procède à une opération de segmentation entre b et a . On obtient donc :

$$(aba, \epsilon, 0) \xrightarrow{sgm(0)} (aba, (ab), 3)$$

On souhaite procéder à la classification de (ab) au niveau supérieur. Il faut pour cela avant tout initialiser le niveau supérieur :

$$(aba, (ab), 3) \xrightarrow{enn} (aba, (ab), 3).(\epsilon, \epsilon, 0)$$

On effectue ensuite l'opération de classification sur (ab) que l'on concatène à $r^{(1)}$:

$$(aba, (ab), 3).(\epsilon, \epsilon, 0) \xrightarrow{cls(0)} (aba, (ab), 3).(A, \epsilon, 0)$$

Il n'y a aucune information supplémentaire à traiter au niveau 1, donc on continue d'acquérir de l'information au niveau 0 :

$$(aba, (ab), 3).(A, \epsilon, 0) \xrightarrow[c]{atr} (abac, (ab), 3).(A, \epsilon, 0)$$

$$(abac, (ab), 3).(A, \epsilon, 0) \xrightarrow[a]{atr} (abaca, (ab), 3).(A, \epsilon, 0)$$

On segmente entre le c et le a :

$$(abaca, (ab), 3).(A, \epsilon, 0) \xrightarrow{str(0)} (abaca, (ab)(ac), 5).(A, \epsilon, 0)$$

On classe pour le niveau 1 :

$$(abaca, (ab)(ac), 5).(A, \epsilon, 0) \xrightarrow{cls(0)} (abaca, (ab)(ac), 5).(AB, \epsilon, 0)$$

Il n'y a toujours pas de segmentation au niveau 1 et on acquiert l'information au niveau 0 :

$$(abaca, (ab)(ac), 5).(AB, \epsilon, 0) \xrightarrow[b]{atr} (abacab, (ab)(ac), 5).(AB, \epsilon, 0)$$

$$(abacab, (ab)(ac), 5).(AB, \epsilon, 0) \xrightarrow[a]{atr} (abacaba, (ab)(ac), 5).(AB, \epsilon, 0)$$

On segmente à nouveau

$$(abacaba, (ab)(ac), 5).(AB, \epsilon, 0) \xrightarrow[a]{str(0)} (abacaba, (ab)(ac)(ab), 7).(AB, \epsilon, 0)$$

On classe pour le niveau 1 :

$$(abacaba, (ab)(ac)(ab), 7).(AB, \epsilon, 0) \xrightarrow{cls(0)} (abacaba, (ab)(ac)(ab), 7).(ABA, \epsilon, 0)$$

On a ici segmentation au niveau 1 :

$$(abacaba, (ab)(ac)(ab), 7).(ABA, \epsilon, 0)$$

$$\xrightarrow{sgm(1)} (abacaba, (ab)(ac)(ab), 7).(ABA, (AB), 2)$$

On doit donc initialiser le niveau 2 pour la classification :

$$(abacaba, (ab)(ac)(ab), 7).(ABA, (AB), 2)$$

$$\xrightarrow{cnn} (abacaba, (ab)(ac)(ab), 7).(ABA, (AB), 2).(\epsilon, \epsilon, 0)$$

et on effectue la classification pour le niveau 2 :

$$(abacaba, (ab)(ac)(ab), 7).(ABA, (AB), 2).(\epsilon, \epsilon, 0)$$

$$\xrightarrow{cls(1)} (abacaba, (ab)(ac)(ab), 7).(ABA, (AB), 2).(A, \epsilon, 0)$$

Nous n'avons plus d'information à traiter au niveau 2, ni d'information à traiter au niveau 1, on continue ainsi à acquérir l'information au niveau 0...

On obtient ainsi les diagrammes selon les étapes présentées en figure [A.1](#).



FIGURE A.1 – Succession des états 1 à 11 des diagrammes formels équivalents aux représentations de niveaux 1 à 3.

Annexe B

Algorithme de compression du diagramme formel multi-échelle

Nous proposons de représenter de manière compressée le diagramme formel multi-échelle obtenu avec MORFOS selon le procédé suivant :

On commence par le diagramme de niveau le plus élevé. Si le matériau n'a jamais été vu, on inscrit le nouveau label et on ouvre une parenthèse, puis on analyse le niveau inférieur à l'instant t correspondant. Si le matériau a déjà été vu, on inscrit le label correspondant et on analyse l'objet suivant

Le code suivant présente l'algorithme de compression et de décompression de la représentation des diagrammes formels multi-échelles obtenus.

```
# 'nb_materials' is a list that corresponds to the number  
of material already encoded for each formal diagram.  
# 'encoded_char' corresponds to the final encoded char  
# 'dictionary' is a dictionary that list for each level  
# the couples that correspond to an actual-level material  
# and the corresponding constituent materials of lower  
# level  
# 'formal_diagrams' are the final decoded formal diagrams.  
global nb_materials  
global encoded_char  
global dictionary  
global formal_diagrams
```

```

def convert(fd_str, lvl, mso, t_start, t_end):
    """ Takes as entry the multi-scale oracle data
    structure 'module_mso' and a level 'lvl' and time
    boundaries 't_start' and 't_end'. Returns a character
    string corresponding to the adequate sub-formal
    diagram. """
    oracle = mso[lvl][0]
    for i in range(t_start, t_end):
        fd_str.append(oracle.data[i] - 1)
    return fd_str

def parse(fd_str):
    """ Takes as entry the character strings corresponding
    to the materials of the formal diagram at a specific
    level 'lvl_str'. Return the first character of this
    string and modify it by deleting this first character.
    """
    if len(fd_str) == 0:
        return None, fd_str
    else:
        c = fd_str[0]
        fd_str = fd_str[1:] # copy the reference, use
        # numpy references if you want to work with views
        # on references
    return c, fd_str

def already_seen(lvl, m_object):
    """ Takes 'object' as input and returns 0 if this
    object has already been seen in this level. Otherwise
    returns 1. """
    global nb_materials
    if m_object > nb_materials[lvl]:
        nb_materials[lvl] = m_object
        return 0
    return 1

def get_boundaries(lvl, mso, t_start):

```

```
    """ Return the lower-level boundaries corresponding to
    the character at stat 't_start' and level 'lvl'."""
    link = mso[lvl - 1][1]
    link_r = link.copy()
    link_r.reverse()
    k_init = link.index(t_start)
    k_end = len(link) - link_r.index(t_start)
    return k_init, k_end

def write(char_object):
    """ Add 'char_object' to the end of the compressed
    char 'encoded_char'."""
    global encoded_char
    encoded_char.append(char_object)
    return 0

def add_dict(level, char):
    """ Add new materials labeled 'char' in the adequate
    level 'level' and in the corresponding couple of upper
    level. """
    global dictionary
    if char == '(' or (level == 0 and len(dictionary[0])
    < char + 1):
        char = len(dictionary[level])
        dictionary[level].append([char, []])
    if level < len(dictionary) - 1:
        dictionary[level + 1][len(dictionary[level + 1])
        - 1][1].append(char)
    return char

def add_df(level, char):
    """ Write a new material labeled 'char' in the formal
    diagram of level 'level'."""
    global formal_diagrams
    if char is not None:
        formal_diagrams[level].append(char)
    return 0
```

```

def add_recursive_df(level, char):
    """ Write a material 'char' in the formal diagram of
    level 'level' and all the adequate material in the
    lower levels """
    global formal_diagrams
    char = min(char, len(dictionary[level]) - 1)
    add_df(level, char)
    if level > 0:
        for i in range(len(dictionary[level][char][1])):
            add_recursive_df(level - 1, dictionary[level]
                [char][1][i])
        level -= 1
    return 0

```

```

def encode(lvl, mso, fd_str, t_start=1, t_end=1):
    fd_str = convert(fd_str, lvl, mso, t_start, t_end)
    c, fd_str = parse(fd_str)
    while c is not None:
        if already_seen(lvl, c) or lvl == 0:
            t_start += 1
            write(c)
            c, fd_str = parse(fd_str)
        else:
            write('(')
            t_start_inf, t_end_inf = get_boundaries(lvl,
                mso, t_start)
            encode(lvl - 1, mso, fd_str=[],
                t_start=t_start_inf, t_end=t_end_inf)
            c, fd_str = parse(fd_str)
            t_start += 1
            write(')')

```

```

def decode(level):
    global encoded_char
    char, encoded_char = parse(encoded_char)
    while char != ')':

```

```

if len(encoded_char) == 0:
    new_char = add_dict(level, char)
    add_df(level, new_char)
    break
if char == '(':
    new_char = add_dict(level, char)
    add_df(level, new_char)
    decode(level - 1)
else :
    add_dict(level, char)
    add_recursive_df(level, char)
char, encoded_char = parse(encoded_char)
parse(encoded_char)

```

Pour rappel, si l'on procède à une analyse sur *Rondo* de Mozart avec l'ensemble des règles B présentées dans le chapitre 4, on obtient les représentations et les représentations segmentées de niveau 0 à 3 suivantes :

$r^{(0)}$: *abacabacdeabfgabachijklmhinopqabr sr sttu*
 $s^{(0)}$: *(ab)(ac)(ab)(ac)(de)(ab)(fg)(ab)(ac)(hi)(jklm)(hi)(nopq)(ab)(ac)(rs)(rs)(ttu)*
 $r^{(1)}$: ABABCADABEFEGABHHI
 $s^{(1)}$: (AB)(AB)(CAD)(AB)(EF)(EG)(AB)(HHI)
 $r^{(2)}$: AABACDAE
 $s^{(2)}$: (AAB)(ACD)(AE)
 $r^{(3)}$: ABC
 $s^{(3)}$: (ABC)

En appliquant l'algorithme, on obtient le résultat suivant :

$A3(A2(A1(A0, B0), B1(A0, C0)), A2, B2(C1(D0, E0), A1, D1(F0, G0)))$
 $B3(A2, C2(E1(H0, I0), F1(J0, K0, L0, M0)), D2(E1, G1(N0, O0, P0, Q0)))$
 $C3(A2, E2(H1(R0, S0), H1, I1(T0, T0, U0)))$

Les indices des lettres correspondent au niveau de diagramme formel correspondant. Comme on peut les retrouver grâce au parenthésage, on peut donc retirer cette numérotation. Par ailleurs, les labels choisis sont par ordre croissant par défaut, on propose donc de ne pas les expliciter devant une parenthèse lorsqu'un nouveau matériau est créé car on sait par défaut de quel label il s'agit. On obtient alors la représentation suivante :

$((((AB)(AC))A((DE)A(FG)))$
 $(A((HI)(JKLM))(E(NOPQ)))$
 $(A((RS)H(TTU)))$

Afin d'éviter ou du moins minimiser les pertes concernant les variations entre deux objets similaires mais non identique, on pourrait exprimer les objets en tant que transformation de l'objet considéré similaire.

Par exemple, si le troisième A ici est subtilement différent de sa première occurrence nous pourrions inscrire $f(A, p)$ avec f la fonction de transformation et p les paramètres correspondants à l'indice et le label de la variation. Il resterait cependant à estimer dans quelle mesure cela compresse effectivement l'information exprimée.

Un réel apport pour MORFOS par ailleurs serait l'implémentation dynamique d'un tel algorithme.

Annexe C

Complexité : calcul du nombre de liens de l'oracle

On présente en figure C.1 le nombre de liens créés en fonction de la qualité du nouvel objet ajouté dans l'oracle et de son préfixe.

Il s'agit d'un arbre de décision en fonction de l'objet dernièrement acquis (selon qu'il s'agisse d'un objet existant en mémoire ou nouveau), et en fonction de son préfixe, au sens défini ici par l'oracle c'est-à-dire la chaîne de caractères précédent l'objet acquis la plus longue déjà vue.

Une répétition interne au préfixe signifie que le préfixe est constitué d'un motif donné : le préfixe *ababab* est par exemple constitué de 3 répétitions du motif *ab*.

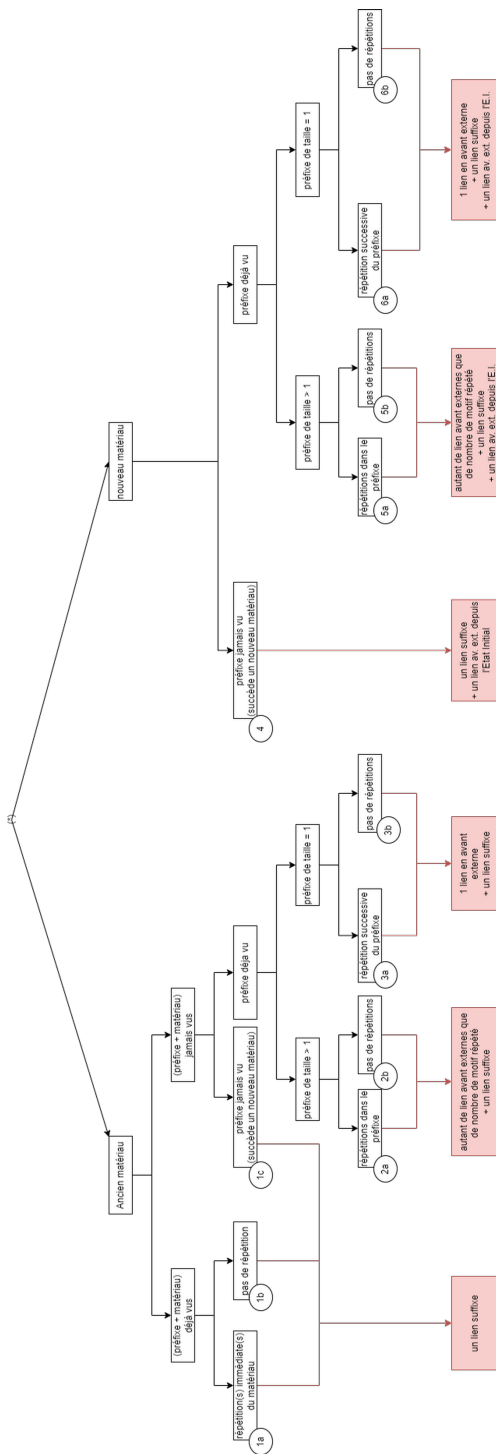


FIGURE C.1 – Arbre des coûts théoriques de l'oracle.

Annexe D

Corpus musical de l'expérience cognitive

Nous présentons ici la liste des œuvres proposées dans le cadre de l'expérience cognitive décrite au chapitre 4 :

Phase	Niveau	Oeuvre
Entraînement	0	L.V. Beethoven : <i>Bagatelle pour piano op.119 no.9</i> (durée 1 :10 min.) – mouvement entier
	0	T.Takemitsu : Valse issu d'une musique de film <i>The Face of Another</i>
Phase 1	1	G.Ligeti : <i>Études pour piano no.16 : Pour Irina</i>
	2	R.Schumann : <i>Kanonische Etude for Pedalfügel op.56 no.2</i>
	3	J.N.Hummel : <i>6 Bagatelles pour piano op.107 no.4 : Rondo all'Ungherese</i>
Phase 2	1	Adam de Saint Victor : Séquence <i>Jubilemus Salvatori</i> XIIe siècle. (Durée 3 :11 min.)
		L.V.Beethoven : <i>7 Bagatelles pour piano op.33 no.1 : Andante grazioso quasi allegretto</i>
		A.Vivaldi : <i>Concerto pour deux violons en la majeur RV :552, 2ém mouvement : Largetto</i>
	2	J.Brahms : <i>4 Klavierstücke pour piano op.119 no.1 : Intermezzo en si mineur</i>

		G.Ligeti : <i>Sonate pour violoncelle, 1er mouvement : Dialogo - Adagio, rubato, cantabile</i>
		H.Partch : <i>Two Studies in Ancient Greek Sclaes</i>
	3	I.Stravinsky : <i>Trois pièces pour quatuor à cordes no.3</i>
		M.Levinas : <i>Froissements d'ailes pour flûte solo</i>
		A.Clyne : <i>Danse II : ...If you've torn the bandage off</i>

Lexique

agenda ensemble de contraintes auditives permettant de passer de la représentation de l'objet tel qu'il est exprimé à la représentation de l'objet tel qu'il est entendu. Dans le cadre de MORFOS, interface de paramétrage.. 65

diagramme formel représentation des matériaux d'une œuvre en fonction du temps.. 30

diagramme formel multi-échelle organisation hiérarchisée et assemblée des diagrammes formels d'une analyse donnée.. 50

facteur concaténation d'objets de niveau inférieur qui constituent un objet de niveau supérieur.. 50

matériau paradigme au sens linguistique du terme, c'est-à-dire une classe d'éléments musicaux substituables.. 30

objet occurrences des matériaux dans une œuvre donnée. . 30

Références

- [1] A. Adamczyk. An encounter : Cognition, computing and music. Master's thesis, Cognitive Science and Ergonomics University of Bordeaux, 2022.
- [2] T. W. Adorno. *Philosophie de la nouvelle musique*. Gallimard, Paris, 1962.
- [3] C. Allauzen, M. Crochemore, and M. Raffinot. Factor oracle : a new structure for pattern matching ; ; theory and practice of informatics. In *Proceedings of SOFSEM'99*, Paris, France, 1999.
- [4] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3) :403–410, 1990.
- [5] Gérard Assayag and Shlomo Dubnov. Using factor oracles for machine improvisation. *Soft Computing*, 8 :604–610, 01 2004.
- [6] Cédric Bach and Dominique Scapin. Critères ergonomiques pour les interactions homme-environnements virtuels : définitions, justifications et exemples. 01 2005.
- [7] R. Bellman and R. Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2) :1–9, 1959.
- [8] I. Bent and W. Drabkin. *L'Analyse musicale. Histoire et Méthodes (trad. A. Cœurdevey et J. Tabouret)*. Éditions Main d'Œuvre, 1998.
- [9] L. Berio. *Entretiens avec Rossana Dalmonte*. Paris, Lattès, 1983.
- [10] Mireille Besson and Frédérique Faïta. An event-related potential (erp) study of musical expectancy : Comparison of musicians with nonmusicians. *Journal of Experimental Psychology : Human Perception and Performance*, 21 :1278–1296, 12 1995.
- [11] Roberta Bianco, Peter MC Harrison, Mingyue Hu, Cora Bolger, Samantha Picken, Marcus T Pearce, and Maria Chait. Long-term implicit memory for sequential auditory patterns in humans. *eLife*, 9, 2020.
- [12] E. Bigand. Les sciences de la cognition musicale. *Bulletin de psychologie*, 46-411 :449–458, 1993.

-
- [13] E. Bigand and S. McAdams. *Contributions de la musique aux recherches sur la cognition auditive humaine*, chapter 8, pages 261–262. 2005.
- [14] L. Bigo and M. Andreatta. Filtration of pitch-class sets complexes. In *7th International Conference, MCM 2019*, pages 213–226, 2019.
- [15] F. Bimbot, E. Deruty, G. Sargent, and E. Vincent. System & contrast : a polymorphous model of the inner organization of structural segments within music pieces. *Music Perception, University Of California Press*, 33 :631–661, 2016.
- [16] Frédéric Bimbot, Gabriel Sargent, Emmanuel Deruty, Corentin Guichaoua, and Emmanuel Vincent. Semiotic description of music structure : an introduction to the quaero/metiss structural annotations. *Proceedings of the AES International Conference*, 01 2014.
- [17] R. Blard, L. Dujardin, L. Héreau, L. Jouhault, G. Perron, and N. Sanson. Interface pour algorithme cognitif d’analyse musicale. Master’s thesis, Ecole Nationale Supérieure d’Electronique, Informatique, Télécommunications, Mathématique et Mécanique, 2022.
- [18] Sebastian Böck, Matthew E. P. Davies, and Peter Knees. Multi-task learning of tempo and beat : Learning one to improve the other. In *International Society for Music Information Retrieval Conference*, 2019.
- [19] Paul Boersma and Vincent Van Heuven. Speak and unspeak with praat. *Glott Int*, 5 :341–347, 01 2001.
- [20] Dibya Bora. A novel approach for color image edge detection using multidirectional sobel filter on hsv color space. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*, 5 :154–159, 02 2017.
- [21] A. Boucourechliev. *Le langage musical*. Fayard, 1993.
- [22] R. Bracewell. *The Fourier transform and its applications*. McGraw-Hill New York, 1965.
- [23] G. Brelet. *le temps musical : essai d’une esthétique nouvelle de la musique, 2 Volumes*. Presses Universitaires de France, Paris, 1949.
- [24] Antoine Brionne. *L’Amusie : un type d’agnosie auditive singulier*. PhD thesis, 11 2010.
- [25] James C. Byers, Alvah C. Bittner, and Susan G. Hill. Traditional and raw task load index (tlx) correlations : Are paired comparisons necessary ? in a. 1989.
- [26] Nicole Calma-Roddin and John Drury. Music, language, and the n400 : Erp interference patterns across cognitive domains. *Scientific Reports*, 10 :1–14, 07 2020.

- [27] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8 :679 – 698, 12 1986.
- [28] Julien Cegarra and Nicolas Morgado. Étude des propriétés de la version francophone du nasa-tlx. *EPIQUE 2009 : 5ème Colloque de Psychologie Ergonomique*, pages 233–239, 01 2009.
- [29] J-M. Celerier, P. Baltazar, C. Bossut, N. Vuaille, J-M. Couturier, and al. Ossia : Towards a unified interface for scoring time and interaction. In *Proc. Int. Conf. on Technologies for Music Notation and Representation - TENOR 2015*, Paris, France, 2015.
- [30] C.E. Cella. On computational creativity for music and the arts. SCRIME, juin 2022.
- [31] J-M. Chouvel. Musical form, from a model of hearing to an analytic procedure. *Interface*, 22 :99–117, 1993.
- [32] J-M. Chouvel. *Esquisses pour une pensée musicale, Les métamorphoses d’Orphée*. L’Harmattan, Paris, 1998.
- [33] J-M. Chouvel. *Analyse Musicale, sémiologie et cognition des formes temporelles*. L’Harmattan, Paris, 2006.
- [34] Blender Community. Blender - a 3d modelling and rendering package. In *Blender Foundation*, <http://www.blender.org>, 2018.
- [35] A. Cont. Antescofo : Anticipatory synchronization and control of interactive parameters in computer music. In *International Computer Music Conference (ICMC)*, pages 33–40, Belfast, Ireland, 2008.
- [36] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19 :297–301, 1965.
- [37] G. W. Cooper and L. Meyer. *The Rhythmic Structure of Music*. Chicago, 1960.
- [38] P. Couprie. Eanalysis : Developing a sound based music analytical tool. *S. Emmerson, L. Landy (éd.), Expanding The Horizon Of Electroacoustic Music Analysis, Cambridge, Cambridge University Press*, 27 :170–194, 2016.
- [39] W. de Haas, Matthias Robine, Pierre Hanna, Remco Veltkamp, and Frans Wiering. Comparing approaches to the similarity of musical chord sequences. volume 6684, pages 242–258, 06 2010.
- [40] B. de Schloezer. *Introduction à J.-S. Bach. Essai d’esthétique musicale*. éditions Gallimard, 1947.
- [41] Sofie Degeest, Katrien Kestens, and Hannah Keppler. Listening effort measured using a dual-task paradigm in adults with different amounts of noise exposure. *Ear & Hearing*, Publish Ahead of Print, 10 2021.

-
- [42] F. Delalande. *La musique au-delà des notes*. Presse Universitaires de Rennes, 2019.
- [43] F. Delalande, M. Formosa, M. Frémiot, P.Gobin, P. Malbosc, J. Mandelbrojt, and E. Pedler. *Les Unités sémiotiques temporelles : éléments nouveaux d'analyse musicale*. Marseille : MIM/Documents Musurgia, 1996.
- [44] I. Deliège. *L'organisation psychologique de l'écoute de la musique*. Thèse de doctorat, Université de Liège, Liège, 1990.
- [45] Daniel Dennett and Marcel Kinsbourne. Time and the observer : The where and when of consciousness in the brain. *Behavioral and Brain Sciences*, 15 :183–247, 06 1992.
- [46] J. Dewey. *Art as experience*. Minton, Balch, 1934.
- [47] J. Downie, Mert Bay, Andreas Ehmann, and Malo Jones. Audio cover song identification : Mirex 2006-2007 results and analyses. pages 468–474, 01 2008.
- [48] Shlomo Dubnov, Gérard Assayag, and Arshia Cont. Audio oracle analysis of musical information rate. pages 567 – 571, 10 2011.
- [49] Tor Endestad, Rolf Inge Godøy, Markus H. Sneve, Thomas Hagen, Agata Bochynska, and Bruno Laeng. Mental effort when playing, listening, and imagining music in one pianist's eyes and brain. *Frontiers in Human Neuroscience*, 14, 2020.
- [50] Michael W. Eysenck. *Fundamentals of Cognition*. Hove, UK : Psychology Press., 2006.
- [51] W.T. Fitch and M.D. Martins. Hierarchical processing in music, language, and action : Lashley revisited. pages 2014 May ;1316(1) :87–104, 2014.
- [52] E. Fix and J.L. Hodges. *Discriminatory Analysis : Nonparametric Discrimination : Consistency Properties*. USAF School of Aviation Medicine, 1951.
- [53] J. Foote. Visualizing music and audio using self-similarity. In *Proc. Of ACM Multimedia*, pages 77–80, 1999.
- [54] J. Foote and S. Uchihashi. The beat spectrum : a new approach to rhythm analysis. In *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.*, pages 881–884, 2001.
- [55] A. Forte. *The Structure of Atonal Music*. New Haven and London : Yale University Press, 1973.
- [56] G. Gabbolini and D. Bridge. An interpretable music similarity measure based on path interestingness. In *in Proc. of the 22nd Int. Society for Music Information Retrieval Conf., Online*, 2021.
- [57] Y. Geslin and A. Lefevre. Sound and musical representation : the acoustographe software. In *In Proceedings Of The 2004 International Computer Music Conference (ICMC 2004)*, 2004.

- [58] V. Gillot and F. Bimbot. Polytopic reconfiguration : a graph-based scheme for the multiscale transformation of music segments and its perceptual assessment. In *SMC 2019-16th Sound & Music Computing Conference*, pages 1–8, 2019.
- [59] Valentin Gillot. *Approches algorithmiques et perceptives pour la recombinaison polytopique de mélodies par critère de complexité multi-échelle*. PhD thesis, 2021. Thèse de doctorat dirigée par Bimbot, Frédéric Informatique Rennes, INSA 2021.
- [60] Chelsea Gordon, Patrice Cobb, and Ramesh Balasubramaniam. Recruitment of the motor system during music listening : An ale meta-analysis of fmri data. *PLoS ONE*, 13, 10 2018.
- [61] G.Shibata, R.Nishikimi, E.Nakamura, and K.Yoshii. Statistical music structure analysis based on a homogeneity-, repetitiveness-, and regularity- aware hierarchical hidden semi-markov model. In *in Proc. of the 20st Int. Society for Music Information Retrieval Conf., Delft, The Netherlands*, 2019.
- [62] G.Shibata, R.Nishikimi, and K.Yoshii. Music structure analysis based on an lstm-hsmm hybrid model. In *in Proc. of the 20st Int. Society for Music Information Retrieval Conf., Delft, The Netherlands*, 2019.
- [63] Corentin Guichaoua. *Modèles de compression et critères de complexité pour la description et l'inférence de structure musicale*. PhD thesis, 2017. Thèse de doctorat dirigée par Bimbot, Frédéric Informatique Rennes 1 2017.
- [64] C. Guillotel-Nothmann. Les signes musicaux et leur étude par l'informatique. le statut épistémologique du numérique dans l'appréhension du sens et de la signification en musique. *Revue musicale OICRM*, 46, 88, 2018.
- [65] S. G. Hart and Lowell E. Staveland. Development of nasa-tlx (task load index) : Results of empirical and theoretical research. *Advances in psychology*, 52 :139–183, 1988.
- [66] N. Herold. *Analyser le timbre : objets, méthodes, représentations*, pages 15–33. dir. Philippe Lalitte, éditions universitaires de dijon edition, 11 2019.
- [67] Dorien Herremans, David Martens, and Kenneth Sörensen. *Composer Classification Models for Music-Theory Building*, pages 369–392. 10 2015.
- [68] David Huffman. A method for the construction of minimum-redundancy codes. In *Proceedings of the I.R.E*, pages 1098–1102, 1952.
- [69] David Huron. *Sweet Anticipation : Music and the Psychology of Expectation*, volume 1. 01 2006.
- [70] E. Husserl. *Méditations cartésiennes, chapitre 2*. Vrin, 1931.
- [71] M. Imberty. Structure perceptive du style musical. *Bulletin de psychologie*, 30.329 :781–791, 1977.

-
- [72] M. Imberty. *La musique creuse le temps, de Wagner à Boulez : musique, psychologie, psychanalyse*. Paris : L'Harmattan, 2005.
- [73] Beatty J. Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychol Bull.*, Mar ;91(2) :276–92, 1982.
- [74] Daniel Kahneman and Jackson Beatty. Pupil diameter and load on memory. *Science*, 154 :1583 – 1585, 1966.
- [75] Michael Klingbeil. Software for spectral analysis, editing, and synthesis. *International Computer Music Conference, ICMC 2005*, 01 2005.
- [76] S Koelsch, Thomas Gunter, Erich Schröger, Mari Tervaniemi, Daniela Sammler, and A Friederici. Differentiating eran and mmn : An erp study. neuroreport, 12, 1385-1389. *Neuroreport*, 12 :1385–9, 06 2001.
- [77] Stefan Koelsch. Towards a neural basis of processing musical semantics. *Physics of life reviews*, 8 :89–105, 04 2011.
- [78] Stefan Koelsch, Elisabeth Kasper, Daniela Sammler, Katrin Schulze, Thomas Gunter, and Angela Friederici. Music, language and meaning : Brain signatures of semantic processing. *Nature neuroscience*, 7 :302–7, 04 2004.
- [79] Siebel WA. Koelsch S. Towards a neural basis of music perception. Dec.9(12) :97–114, 2005.
- [80] A. N. Kolmogorov. Three approaches to the quantitative definition of information. In : *Prob Info Trans*, 1.1 :3–11, 1965.
- [81] Solomon Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22 :79–86, 1951.
- [82] M. Kutas and S. A. Hillyard. Reading senseless sentences : Brain potentials reflect semantic incongruity. *Science*, 207(4427) :203–205, 1980.
- [83] P. Lalitte. *Musique et cognition : perspectives pour l'analyse et la performance musicales*. éditions universitaires de Dijon, 2019.
- [84] Jörg Langner, Reinhard Kopiez, Christian Stoffel, and Martin Wilz. Realtime analysis of dynamic shaping. 08 2000.
- [85] O. Laske. *Music, Memory and Thought, explorations in Cognitive Musicology*. University Microfilms International, Ann Arbor (MI), 1977.
- [86] O. Laske. *A Rule System for Music-Analytical Discoveries*. Modena, 1982.
- [87] E. Leipp. *La Machine à Ecouter, Essai de psycho-acoustique*. Masson, Paris, 1977.
- [88] F. Lerdhal and R. Jackendoff. *A generative theory of tonal music*. Cambridge, Mass. : MIT Press, 1983.
- [89] D. Levitin. *This Is Your Brain on Music*. USA : Dutton Penguin, 2006.
- [90] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2) :129–137, 1982.

- [91] Corentin Louboutin and Frédéric Bimbot. Modeling the multiscale structure of chord sequences using polytopic graphs. 2017, hal-01653455.
- [92] Roisin Loughran, Jacqueline Walker, Michael O’Neill, and Marion O’Farrell. The use of mel-frequency cepstral coefficients in musical instrument identification. 08 2008.
- [93] Gaulhiac Manuel. *Les descripteurs harmoniques : étude théorique et applications musicologiques*. PhD thesis, 06 2021.
- [94] A. Marmoret, J. Cohen, N. Bertin, and F. Bimbot. Uncovering audio patterns in music with nonnegative tucker decomposition for structural segmentation. *ISMIR 2020 - 21st International Society For Music Information Retrieval*, 33 :1–7, 2020.
- [95] A. Marmoret, J. Cohen, and F. Bimbot. Convolutional block-matching segmentation algorithm with application to music structure analysis. Saint-Etienne, France, June 2022.
- [96] B. Martin. *Analyse de structures répétitives dans les séquences musicales*. Université Sciences et Technologies - Bordeaux I, 2012 fftel-01086136f.
- [97] Benjamin Martin, D.G. Brown, Pierre Hanna, and Pascal Ferraro. Blast for audio sequences alignment : a fast scalable cover identification. 01 2012.
- [98] M. Mathews. *The Thechnology of Computer Music*. The MIT Press, 1969.
- [99] B. McFee and K. Kinnaird. Improving structure evaluation through automatic hierarchy expansion. In *20th International Society for Music Information Retrieval Conference, Delft, The Netherlands*, 2019.
- [100] D. L. Medin and A. Ortony. Psychological essentialism. In *S. Vosniadou & A. Ortony (Eds.), Similarity and analogical reasoning*, 22, Issue 2 :179–195, 1989.
- [101] N Meeùs. Épistémologie d’une musicologie analytique. volume XXII :97–114, 2002.
- [102] M. Mesnage. Morphoscope, a computer system for music analysis. *Interface*, 22, Issue 2 :119–131, 1993.
- [103] L. B. Meyer. *Emotion and Meaning in Music*. University of Chicago Press : Chicago, 1956.
- [104] E. Michel. Rapport de stage sur la conception d’une ihm pour un algorithme d’analyse musicale. Master’s thesis, Université Toulouse Jean-Jaurès, 2021.
- [105] J. Molino. Fait musical et sémiologie de la musique. *Musique enjeu*, 17 :37–62, 1975.
- [106] N. Méeus. De la forme musicale et de sa segmentation. *Musurgia, Eska, Paris*, I, n°1, 1994.

-
- [107] Daniel Müllensiefen, B. Gingras, J. Musil, and L. Stewart. Measuring the facets of musicality : The goldsmiths musical sophistication index (gold-msi). *Personality and Individual Differences*, 60 :S35, 04 2014.
- [108] Meinard Müller, Henning Mattes, and Frank Kurth. An efficient multiscale approach to audio synchronization. 10 2006.
- [109] Eugene Narmour. *The Analysis and Cognition of Basic Melodic Structures : The Implication-Realization Model*. University of Chicago Press, Chicago,, 1990.
- [110] Eugene Narmour. *The Analysis and Cognition of Melodic Complexity : The Implication-Realization Model*. University of Chicago Press, Chicago,, 1992.
- [111] J. Nattiez. *Musicologie générale et sémiologie*. Christian Bourgois, 1987.
- [112] Jean-Jacques Nattiez and Jonathan M. Dunsby. Fondements d’une sémiologie de la musique. *Perspectives of New Music*, 15(2) :226–233, 1977.
- [113] J.J. Nattiez. *Analyses et interprétations de la musique : La mélodie du berger dans le Tristan et Isolde de Richard Wagner*. Paris, Vrin, 2013.
- [114] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3) :443–453, 1970.
- [115] Christiane Neuhaus, Thomas R. Knösche, and Angela D. Friederici. Effects of musical expertise and boundary markers on phrase perception in music. *Journal of Cognitive Neuroscience*, 18 :472–493, 2006.
- [116] K. A. Paller, M. Kutas, and H. K. McIsaac. Monitoring conscious recollection via the electrical activity of the brain. *Journal of cognitive neuroscience*, 6(2) :107–111, 1995.
- [117] Aniruddh Patel. Patel, a. d. language, music, syntax and the brain. *nature neurosci. Nature neuroscience*, 6 :674–81, 08 2003.
- [118] Aniruddh Patel, Edward Gibson, Jennifer Ratner, Mireille Besson, and Philip Holcomb. Processing syntactic relations in language and music : An event-related potential study. *Journal of cognitive neuroscience*, 10 :717–33, 12 1998.
- [119] William B. Pennebaker and Joan L. Mitchell. Jpeg : Still image data compression standard. 1992.
- [120] Yurika Permanasari, Erwin Harahap, and Erwin Prayoga. Speech recognition using dynamic time warping (dtw). *Journal of Physics : Conference Series*, 1366 :012091, 11 2019.
- [121] Jeremy Pickens and Tim Crawford. Harmonic models for polyphonic music retrieval. page 430, 01 2002.

- [122] Justine Pomerleau Turcotte, Maria Teresa (Maite) Moreno Sala, Francis Dubé, and François Vachon. Does cognitive load differ among sight-singers? an exploratory study using pupillometry and interviews. 12 2021.
- [123] Maxime Poret, Myriam Desainte-Catherine, and Catherine Semal. Perceptual study of sound ecology in peripheral sonification. In *Proceedings of SMC 2022 - 19th Sound and Music Computing Conference*, Saint-Étienne, France, June 2022.
- [124] L. Postman. Perception, motivation, and behavior. *Journal of Personality*, 22, Issue 1 :17–31, 1953.
- [125] R.K. Potter. Visible patterns of sound. *Science*, 1945.
- [126] L. Pottier. *L'analyse musicale et sonore De la recherche à la recommandation*. Cnmlab edition, 2023.
- [127] Kunert R., Willems R. M., Casasanto D., and Hagoort P. Patel A. D. Music and language syntax interact in broca's area : An fmri study. *PLoS One*, 10(11) :e0141069 :674–81, 2015.
- [128] Colin Raffel, Brian Mcfee, Eric Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel Ellis. mir_eval : A transparent implementation of common mir metrics. 10 2014.
- [129] H. Riemann. « *Formes musicales* », *Dictionnaire de musique*,. Paris, 1931.
- [130] A. Riotte. *From Traditional to Formalized Analysis; in memoriam Jean Barraqué : some Examples Drawn from his Unpublished Analysis of Anton Webern's Piano Variations op.27*, pages 131–153. Modena, 1982.
- [131] J.C. Risset. Pitch control and pitch paradoxes demonstrated with computer-synthesized sounds. *J. Acoust. Soc. Am.*, 46, 88, 1969.
- [132] M. Rohrmeier. Towards a formalization of musical rhythm. In *in Proc. of the 21st Int. Society for Music Information Retrieval Conf., Montréal, Canada*, 2020.
- [133] C. Rosen. *The classical style : Haydn, Mozart, Beethoven Hardcover*. Viking Press, 1971.
- [134] N. Ruwet. Méthodes d'analyse en musicologie. *Revue Belge De Musicologie / Belgisch Tijdschrift Voor Muziekwetenschap*, 20 No. 1/4 :65–90, 1966.
- [135] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1) :43–49, 1978.
- [136] S. Samson. *Perception des timbres musicaux*, pages 123–146. Bernard Lechevalier éd., de boeck supérieur edition, 2010.
- [137] J. Di Santo. A sign to write acousmatic scores. In *Int. Conf. On Technologies For Music Notation And Representation - TENOR 2015*, 2015.

-
- [138] Marc Sarfati, Anthony Hu, and Jonathan Donier. Ensemble-based cover song detection. 05 2019.
- [139] G. Sargent. *Estimation de la structure de morceaux de musique par analyse multi-critères et contrainte de régularité*. PhD thesis, PhD Thesis, Rennes University, 2013.
- [140] F. De Saussure. *Cours de linguistique générale*. Payot, 1906.
- [141] P. Schaeffer. *Traité des objets musicaux : essai interdisciplines*. Éditions du seuil, 1966.
- [142] H. Schenker. *Das Meisterwerk in der Musik : Ein Jahrbuch*, volume 3 vols, Vol. 1. Munich, Vienna and Berlin : Drei Masken Verlag, 1925-30, 1925.
- [143] Hendrik Schreiber, Frank Zalkow, and Meinard Müller. Modeling and estimating local tempo : a case study on chopin’s mazurkas. 10 2020.
- [144] C. Schörkhuber and A. Klapuri. Constant-q transform toolbox for music processing. 2010.
- [145] Ayush Shah, Manasi Kattel, Araj Nepal, and Dichha Shrestha. Chroma feature extraction. 01 2019.
- [146] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 :379–423, 1948.
- [147] C.E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1) :10–21, 1949.
- [148] Roger N. Shepard. Circularity in judgments of relative pitch. *Journal of the Acoustical Society of America*, 36 :2346–2353, 1964.
- [149] L. R. Slevc and A. D. Patel. Meaning in music and language : Three key differences. comment on “towards a neural basis of processing musical semantics” by stefan koelsch. *Physics of Life Reviews*, 8(2) :110–111, 2011.
- [150] J. Sloboda. Music structure and emotional response : Some empirical findings. *Psychology of Music*, 19 :110–120, 1991.
- [151] R. Snyder. *Music and Memory*. The MIT Press, 2001.
- [152] Lijun Sun, Fang Liu, Linshu Zhou, and Cunmei Jiang. Musical training modulates the early but not the late stage of rhythmic syntactic processing. *Psychophysiology*, 55(2).e12983., 2018.
- [153] Xiaoheng Sun, Qiqi He, Yongwei Gao, and Wei Li. Musical tempo estimation using a multi-scale network. In *in Proc. of the 22nd Int. Society for Music Information Retrieval Conf., Online*, 2021.
- [154] John Sweller, Paul Ayres, and Slava Kalyuga. Cognitive load theory. 01 2011.
- [155] John Sweller and Paul Chandler. Why some material is difficult to learn. *Paul Chandler*, 12, 09 1994.

- [156] J.P. Thibaut. Similarité et catégorisation. *L'année psychologique*, 97, n°4 :701–736, 1997.
- [157] Knösche TR, Neuhaus C, Haueisen J, Alter K, Maess B, Witte OW, and Friederici AD. Perception of phrase structure in music. *Hum Brain Mapp.*, 24(4) :259–73, Apr 2005.
- [158] S. Troche. Structure et méthode dans la musique de john cage : une discipline d'attention. *Nouvelle revue d'esthétique*, 9 :91–104, 2012.
- [159] M. Vignal. *Dictionnaire de la Musique*. Larousse, 2017.
- [160] C. Wang and S. Dubnov. The variable markov oracle : Algorithms for human gesture applications. In *IEEE Multimedia*, vol. 22, no. 4, pages 52–67, 2015, doi : 10.1109/MMUL.2015.76.
- [161] Cheng-i Wang and Shlomo Dubnov. Pattern discovery from audio recordings by variable markov oracle : A music information dynamics approach. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 683–687, 2015.
- [162] Paul Werner, Alan Swope, and Fred Heide. The music experience questionnaire : Development and correlates. *The Journal of psychology*, 140 :329–45, 07 2006.
- [163] Clemens Wöllner and David Hammerschmidt. Tapping to hip-hop : Effects of cognitive load, arousal, and musical meter on time experiences. *Attention, Perception, & Psychophysics*, 83, 01 2021.
- [164] Xide yu, Tao Liu, and Dingguo Gao. The mismatch negativity (mmn) : An indicator of perception of regularities in music. *Behavioural neurology*, 10 2015.
- [165] Chao Zhen and Jieping xu. Multi-modal music genre classification approach. pages 398 – 402, 08 2010.
- [166] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5) :530–536, 1978.
- [167] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory*, 23 :337–343, 1977.

Publications

Articles dans des conférences évaluées par les pairs

- [168] J. Calandra, J.M. Chouvel, and M. Desainte-Catherine. Multi-scale oracle and automated representation of formal diagrams based on the cognitive algorithm. In *Int. Conf. On Technologies For Music Notation And Representation - TENOR 2021*, 2021.
- [169] J. Calandra, J.M. Chouvel, M. Desainte-Catherine, and al. Génération automatique de diagrammes formels par un algorithme cognitif modulaire : étude préliminaire. In *Journées d'Informatique Musicales (JIM)*, Strasbourg, France, 2020.
- [170] J. Calandra, J.M. Chouvel, M. Desainte-Catherine, and E. Michel. Visualisation of multi-scale formal diagrams for music analysis. In *2023 International Workshop on Multilayer Music Representation and Processing (MMRP)*, 2023.

Posters

- [171] J. Calandra, P. Hanna, P. Legrand, and M. Desainte-Catherine. Pict2audio : Sound generation by hand-drawn images using convolutional neural networks. In *Linux Audio Conference (LAC)*, 2020.

L'algorithmie Cognitive et ses applications musicales

Résumé

Cette thèse présente la formalisation et le développement du logiciel d'analyse musicale MORFOS (*Multiscale Oracle Representations For Organized Sounds*). Il s'agit d'une automatisation de l'Algorithme Cognitif présenté par Jean-Marc Chouvel, qui s'inscrit dans la continuité de l'analyse sémiologique et de la musicologie cognitive et permet la représentation multi-échelle d'un modèle de la cognition lors de l'écoute musicale. En automatisant cet algorithme, nous cherchons à mieux comprendre les phénomènes de l'écoute musicale et plus particulièrement les phénomènes de classification et de segmentation au cours du temps, ainsi que les phénomènes d'anticipation et de surprise. Nous nous intéressons également à la visualisation de la représentation obtenue, appelée diagramme formel multi-échelle. On cherche ainsi à analyser l'apport de la hiérarchisation pour les phénomènes musicaux étudiés et les rapports qui émergent entre les éléments étudiés aux différentes échelles de temps.

Mots-clés analyse musicale, algorithme cognitif, musicologie cognitive, diagramme formel

The Cognitive Algorithm and its musical applications

Abstract

This thesis presents the formalization and development of the music analysis software MORFOS (*Multiscale Oracle Representations For Organized Sounds*). It is an automation of the Cognitive Algorithm presented by Jean-Marc Chouvel, which is in line with the continuity of semiological analysis and cognitive musicology, and enables the multi-scale representation of a model of cognition during musical listening. By automating this algorithm, we aim to gain a better understanding of the phenomena of musical listening, and more specifically of the phenomena of classification and segmentation over time, as well as the phenomena of anticipation and surprise. We are also interested in the visualization of the resulting representation, known as a multi-scale formal diagram. We aim to analyze the contribution of hierarchization to the musical phenomena studied, and the relationships that emerge between the elements studied at different time scales.

Keywords : musical analysis, cognitive algorithm, cognitive musicology, formal diagram

UNIVERSITÉ SORBONNE UNIVERSITÉ

ÉCOLE DOCTORALE :

ED V – Concepts et langages

Maison de la Recherche, 28 rue Serpente, 75006 Paris, FRANCE

DISCIPLINE : Musique et musicologie