



HAL
open science

Un modèle relationnel de la gravure musicale.

Mike Solomon, Dominique Fober

► **To cite this version:**

Mike Solomon, Dominique Fober. Un modèle relationnel de la gravure musicale.. Journées d'Informatique Musicale, 2015, Montréal, Canada. hal-02159021

HAL Id: hal-02159021

<https://hal.science/hal-02159021>

Submitted on 18 Jun 2019

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.

2.1. Modèles de représentation musicale

Il existe actuellement trois modèles communément utilisés de représentation musicale : les langages de balisage exploitables dans plusieurs applications, les syntaxes d'entrée textuelles pour des éditeurs et les représentations internes de la musique associées principalement à des éditeurs graphiques. Les langages génériques, dont abc [7], MusicXML [10] et Music Encoding Initiative (MEI) [28] proposent des conteneurs hiérarchiques (instrument, portée, accord, etc.) pour la modélisation de la musique. Les syntaxes liées à des éditeurs comme ceux qui embarquent libGUIDOEngine [17], MusicTeX [34] et GNU LilyPond [23] se basent sur un langage se rapprochant de celui de $\text{T}_\text{E}\text{X}$ alors que des projets comme Common Music Notation (CMN) [31] et Abjad [35] se servent largement des langages d'extension LISP et Python pour définir un format d'entrée musicale.

Plusieurs systèmes de représentation existent à l'intérieur de ces logiciels afin de regrouper, ordonner et traiter les objets musicaux avant d'effectuer la mise en page. La plupart des logiciels utilisent un système orienté objet (MuseScore, GUIDO) qui attribue des propriétés à des objets graphiques (altérations, clefs, hampes etc.) tandis qu'une minorité utilise la programmation fonctionnelle (CMN, LilyPond) pour représenter la musique comme un flot d'événements à traiter par une série de fonctions [30].

2.2. Algorithmes de mise en page

Les algorithmes de mise en page musicale peuvent être classés dans deux groupes principaux : les algorithmes coûteux et affinés pour les langages musicaux compilés et les algorithmes plus rapides et moins précis pour la mise en page en temps réel. La plupart de langages compilés s'efforcent de suivre les recommandations d'espacement musical soulignées par Read [25], Ross [29] Gould [12] et d'autres graveurs reconnus. Les travaux de Gourlay [13], Haken et Blostein [15] et Renz [26] traduisent ces textes en algorithmes d'espacement qui s'inspirent largement de l'algorithme *boîte et colle* proposé par Knuth dans $\text{T}_\text{E}\text{X}$ [20]. Les logiciels proposant une mise en page en temps réel, comme MuseScore et NoteFlight, utilisent une variation de l'algorithme coûteux de sauts de ligne utilisé par la plupart d'éditeurs de texte graphique et les engins de rendu de texte dans les navigateurs web.

2.3. Visualisation de la partition

Dans la plupart d'éditeurs survolés ci-dessus (MuseScore, LilyPond, GUIDO, CMN), la visualisation de la partition sur un écran d'ordinateur est étroitement liée à l'impression éventuelle de la partition sur un support papier. Plusieurs technologies existent également pour la visualisation des partitions musicales sur des appareils d'affichage numériques,

notamment des logiciels qui font défiler une partition pendant un concert de musique [9] [24] [8]. Il existe également des éditeurs qui calque l'espacement horizontal d'une partition sur la forme d'onde d'un enregistrement (Apple Logic, Finale, Antescofo). Des widgets interactifs de gravure musicale existent également dans des langages de programmation visuelle utilisées pour la génération du son et de la vidéo (Pd, Max/MSP) et pour la composition musicale (OpenMusic, PWGL).

Les outils de gravure disponibles dans ces environnements de programmation visuelle s'adaptent à la logique des langages de programmation, s'intégrant souvent dans un flot de données pour pouvoir visualiser un processus sonore ou le résultat d'un algorithme génératif. Ils doivent faire face à de lourdes contraintes de taille et de vitesse [19] pour pouvoir calculer et afficher rapidement des partitions. Dans l'Expressive Notation Package (ENP) [21], cette idée est élaborée dans un espace musical multidimensionnel où chaque visualisation affichée sur l'écran correspond à des instances différentes de la partition (esquisse de compositeur, partition d'interprète, etc.).

3. UN MODÈLE RELATIONNEL DE LA GRAVURE MUSICALE

Bien que l'état de l'art propose un riche ensemble de logiciels de gravure, aucun logiciel n'est adapté ni à l'affichage des partitions sur Internet en temps réel, ni à l'édition à plusieurs. Pour combler cette lacune, le présent article propose un système de gravure musicale assistée par ordinateur basé sur le *modèle relationnel*. Dans ce modèle, les données liées aux objets graphiques de la partition sont stockées dans une base de données relationnelle et modifiées par le biais d'opérations d'algèbre relationnelle. Des applications externes récupèrent les résultats de ces calculs afin d'effectuer la mise en page des objets musicaux.

Cette section avancera l'hypothèse que les éléments graphiques d'une partition musicale peuvent être encodés, manipulés et partagés de façon expressive et efficace dans le cadre ce modèle.

3.1. Définition du modèle relationnel

Le modèle relationnel [5], élaboré par IBM pendant les années 1960, répondait à l'époque à un besoin de séparer les représentations internes des données et les opérations sur ces données comme insérer, supprimer, modifier et interroger. Les deux buts principaux du modèle relationnel sont la gestion de l'« *indépendance des données*—l'indépendance des logiciels visuels et des utilités de ligne de commande par rapport à la représentation interne des données—et certaines formes d'*incohérence des données* qui nuisent au fonctionnement de plusieurs systèmes de calcul, y compris les systèmes non inférentiaux » [5, p.377]. Si l'indépendance des

données permet à différents clients de travailler de façon asynchrone à partir d'un noyau central de données, l'incohérence des données se produit quand les mêmes données sont exploitées de manière différente selon le client.

Le modèle relationnel résout ces deux problèmes en joignant de façon dynamique un ensemble de données à un ensemble de relations. Le mot « relation » est utilisé dans le sens mathématique classique du terme : il s'agit d'un ensemble de n -uplets où, pour chaque indice i d'un n -uplet, la valeur à cette indice existe dans un ensemble S_i . Plus intuitivement, un n -uplet est un rang dans une base données. Les rangs, superposés, créent des colonnes qui correspondent aux ensembles S_i . Bien que l'analogie de colonnes et de rangs soit utile, le modèle relationnel ne préconise aucun système de stockage interne des données. Grâce à cette définition abstraite de l'organisation des données, les clients qui interrogent le système n'ont pas besoin de connaître le schéma de représentation à l'intérieure de la base de données. Autrement dit, les données et les applications sont *indépendantes*. Quant à l'*incohérence* des données, s'il existe une relation R_1 entre Employée et Prénom et une deuxième relation R_2 entre Employée et Nom de famille, une application n'est pas obligée de définir le prénom et le nom de famille afin de représenter la personne dans le système. Le sens des relations n'est pas imposé par le modèle relationnel mais plutôt par les applications qui interrogent la base de données. Ça permet aux logiciels travaillant avec la base de données de ne stocker que les données dont ils ont besoin et d'utiliser ces données pour des fins différentes.

3.2. La partition musicale comme ensemble de relations

Tous les textes de référence de la gravure musicale définissent un ensemble de relations géométriques plus ou moins strictes entre des symboles musicaux. Par exemple, Gould dans *Behind Bars* [12] indique que :

Pour éviter les collisions entre les liaisons et les articulations ou les liaisons de phrasé, il faut que les bouts de la liaison soient alignés aux bords des têtes de note. Sinon, il faut que la liaison commence juste après la première tête de note et qu'elle se termine juste avant la deuxième. [12, p.62]



A l'intérieur de ce constat se trouve de nombreuses relations – tête de note et articulation, tête de note et liaison de phrasé, tête de note et liaison, liaison et articulation etc. Un modèle relationnel permet d'englober plusieurs règles du même genre sans hiérarchiser les rapports entre les objets. Par exemple, on ne précise pas qu'une articulation fait



Figure 2. Une liaison de phrasé qui arrive au même moment qu'une indication de nuance et un symbole d'articulation.

partie d'une note ou vice versa – il faut tout simplement établir un ensemble de relations et la gravure musicale est le résultat graphique qui découle de ces contraintes. Il ne faut pas non plus établir un ensemble d'objets de notation musicale « essentiels » que tous les clients doivent savoir trier. Un logiciel de gravure qui ne propose pas de liaisons de phrasé, par exemple, peut omettre les relations concernant cet élément graphique sans compromettre les autres relations.

3.3. Communication dans le modèle relationnel

Le modèle relationnel de la gravure musicale assistée par ordinateur permet d'effectuer des opérations sur les données par le biais de l'algèbre relationnelle [5]. Cette algèbre comprenait au départ cinq opérations (permutation, projection, jointure, composition et restriction) auxquelles s'ajoutent trois opérations supplémentaires (jointure externe, projection étendue, et fermeture transitive) dans Structured Query Language (SQL) [6] – un format d'échange mondialement répandu pour manipuler et interroger des bases de données relationnelles.

L'un des avantages majeurs de l'algèbre relationnelle est sa capacité d'exprimer des effets d'interférence qui sont difficiles à préciser dans une organisation hiérarchique des connaissances. Par exemple, dans la gravure traditionnelle, une liaison de phrasé peut influencer sur l'espacement vertical d'une articulation et une indication de nuance. (Figure 2). Or, plusieurs textes sur la gravure musicale ([25] [29] [12]) traitent les courbes et les symboles d'expressivité dans différents chapitres, n'exposant que superficiellement les multiples relations qui peuvent survenir entre ces éléments graphiques. En général, les supports qui obligent à hiérarchiser les connaissances, comme les livres et les pages web, ne sont pas bien adaptés aux domaines qui doivent prendre en compte beaucoup d'effets d'interférence. Avec l'algèbre relationnelle (Figure 3), les relations peuvent encadrer des effets transversaux entre plusieurs éléments, permettant de modéliser facilement des effets d'interférence dans l'espacement des objets graphiques que représentent les données.

```

SELECT y_position.val
FROM y_position
JOIN x_position
  ON y_position.id = x_position.id
JOIN name
  ON y_position.id = name.id
WHERE name = 'articulation'
OR name = 'dynamic'
OR name = 'slur';

```

Figure 3. Pseudo-code en SQL qui récupère les coordonnées Y des liaisons, nuances et articulations simultanées.

4. PURCELL : UNE LIBRAIRIE RELATIONNELLE DE GRAVURE MUSICALE ASSISTÉE PAR ORDINATEUR

Le projet `purcell` propose une mise en œuvre du modèle relationnel de la gravure assistée par ordinateur. `purcell` est basé sur un ensemble de tables liées par le biais de relations auxquelles on peut accéder en passant par des websockets, HTTP ou OSC.

Les relations SQL sont des *relations déclencheuses* qui induisent d'autres relations si certaines conditions sont réunies.

Par exemple, la modification de la table « hauteur » déclenche une modification de la position verticale d'un objet sur une portée. Les déclenchements successifs conduisent au remplissage des tables « finales » qui indiquent les informations graphiques associées aux objets (points d'ancrage, coordonnées des sommets etc.). Les contenus de ces tables sont envoyés au client qui effectue la dernière étape de dessin des objets à partir des données fournies par la base de données. D'éventuelles modifications de la partition sont effectuées uniquement par le biais d'opérations d'algèbre relationnelle (Figure 6).

Le projet `purcell` a pour but de proposer des widgets interactifs intégrables dans les logiciels musicaux indiqués dans la Table 1 d'ici 2017. Pour visualiser une preuve de

OpenMusic
PWGL
INScore
Faust WebAudio
MediaWiki

Table 1. Plate-formes de composition et édition musicales dans lesquelles la librairie sera embarqué.

concept de `purcell` écrite en HTML5, veuillez consulter :

<http://www.mikesolomon.org/purcell>

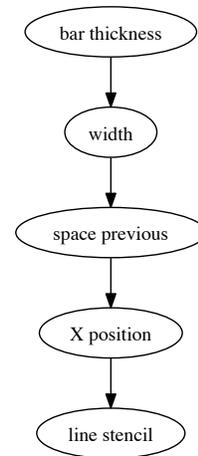


Figure 4. Le cheminement du déclenchement des tables dans une base de données relationnelle qui modélise la gravure musicale. La chaîne commence avec l'épaisseur d'une barre de mesure (`bar thickness`) et procède jusqu'à sa représentation graphique (`line stencil`) en passant par la largeur totale de l'objet (`width`), l'espace entre cet objet et celui à gauche sur la portée (`space previous`) et le positionnement sur l'abscisse (`X position`).

```

CREATE TRIGGER bar_thickness_to_width
AFTER INSERT ON bar_thickness
WHEN (EXISTS (SELECT bar_thickness.id
              FROM bar_thickness))
BEGIN
  DELETE FROM width
  WHERE width.id = new.id;
  INSERT INTO width (id, val)
  WITH bar_lines_to_widths AS
  (SELECT bar_thickness.id AS id,
         bar_thickness.val AS val
   FROM bar_thickness
   WHERE bar_thickness.id = new.id)
  SELECT bar_lines_to_widths.id,
         bar_lines_to_widths.val
  FROM bar_lines_to_widths;
END;

```

Figure 5. Le code SQLite qui correspond au déclenchement d'un changement de largeur d'une barre (`width`) issue d'un changement d'épaisseur de barre, soit la première flèche de la Figure 4.



```
UPDATE accidental SET val = -1 WHERE val = 1;
```



Figure 6. Une partition purcell générée en SVG suivie par une requête SQL et la modification qui en découle.

5. CONCLUSION : LA TOILE ET L'ENCODAGE MUSICAL

Dans la mesure où `purcell` peut fonctionner en mode serveur-client, il est bien adapté aux exigences des plateformes d'édition musicale sur Internet. Il est conçu notamment pour l'utilisation décentralisée à plusieurs comme les éditeurs textuels de Google Doc et de Framapad. Grâce à la librairie `sql.js`¹, `purcell` peut être utilisé comme application cliente dans un navigateur sans communiquer avec un serveur.

Le modèle relationnel de la gravure musicale fait partie de l'écosystème général des langages d'encodage des partitions musicales. Grâce à ses aspects non hiérarchiques et déstructurés, il peut schématiser les données provenant d'autres systèmes d'encodage musical sans dénaturer l'information. Les divergences entre plusieurs systèmes de représentation musicale (MusicXML, MEI etc.) sont une forme d'*incohérence de données* que le modèle relationnel peut encapsuler, permettant de gérer plusieurs systèmes d'encodage simultanément dans la même base de données. Le modèle relationnel peut donc constituer une passerelle entre plusieurs formats d'encodage musical, voire générer des données de méta-encodage qui révèlent les tendances d'encodage de plusieurs clients travaillant avec différents formats d'entrée des données.

6. REFERENCES

- [1] Carlos Agon and Gérard Assayag. Programmation visuelle et éditeurs musicaux pour la composition assistée par ordinateur. In *Proceedings of the 14th French-speaking conference on Human-computer interaction (Conférence Francophone sur l'Interaction Homme-Machine)*, pages 205–206. ACM, 2002.
- [2] David Bainbridge, Craig G Nevill-Manning, Ian H Witten, Lloyd A Smith, and Rodger J McNab. Towards a digital library of popular music. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 161–169. ACM, 1999.
- [3] Dorothea Blostein and Lippold Haken. Justification of printed music. *Communications of the ACM*, 34(3) :88–99, 1991.
- [4] Don Byrd. *Music Notation by Computer*. PhD thesis, Indiana University, 1984.
- [5] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6) :377–387, 1970.
- [6] C Date. *SQL and relational theory : how to write accurate SQL code*. " O'Reilly Media, Inc.", 2011.
- [7] Bryan Duggan, Brendan O'Shea, and Padraig Cunningham. A system for automatically annotating traditional irish music field recordings. In *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pages 25–32. IEEE, 2008.
- [8] Jules Egyud, Gregory J Winksy, and Michael Woolf. Hand held electronic music encyclopedia with text and note structure search, April 14 1998. US Patent 5,739,451.
- [9] Gary Fitzpatrick and Ernest Romero. Networked electronic music display stands, June 2 1998. US Patent 5,760,323.
- [10] Michael Good. Musicxml for notation and analysis. *The virtual score : representation, retrieval, restoration*, 12 :113–124, 2001.
- [11] Michael Good and Geri Actor. Using musicxml for file interchange. In *Web Delivering of Music, International Conference on*, pages 153–153. IEEE Computer Society, 2003.
- [12] Elaine Gould. *Behind Bars, the Definitve Guide to Music Notation*. Faber Music, London, 2011.
- [13] John S. Gourlay. Spacing a Line of Music. Technical Report OSU-CISRC-10/87-TR35, Department of Computer and Information Science, The Ohio State University, 1987.
- [14] Fabrice Guédy, Frédéric Bevilacqua, Dominique Fober, and Loïc Robert. Etudier et jouer une oeuvre musicale en manipulant des objets. *Actes des Journées d'Informatique Musicale JIM2013, Paris, France*, 2013.
- [15] Lippold Haken and Dorothea Blostein. A new algorithm for horizontal spacing of printed music. pages 118–119. International Computer Music Association, 1995.
- [16] W Hegazy and J Gourlay. Optimal line breaking in music. In *Document Manipulation and Typography : Proceedings of the International Conference on Electronic Publishing, Document Manipulation, and Typography*, pages 157–169, 1988.
- [17] Holger H Hoos, Keith A Hamel, Kai Renz, and Jürgen Kilian. The guido notation format—a novel approach for adequately representing score-level music. 1998.

1. <http://github.com/kripken/sql.js/>

- [18] Sergi Jordà and O Wust. A system for collaborative music composition over the web. In *Database and Expert Systems Applications, 2001. Proceedings. 12th International Workshop on*, pages 537–542. IEEE, 2001.
- [19] Edward Kelly. Gemnotes : a realtime music notation system for pure data. In *Proceedings of IV International Conference of Pure data–Weimar*, 2011.
- [20] Donald E. Kunth and Michael F. Plass. Breaking Paragraphs into Lines. *Software – Practice and Experience*, 11 :1119–1184, November 1981.
- [21] Mika Kuuskankare and Mikael Laurson. Expressive notation package. *Computer Music Journal*, 30(4) :67–79, 2006.
- [22] Mikael Laurson and Mika Kuuskankare. Pwgl : a novel visual language based on common lisp, clos and opengl. In *Proceedings of International Computer Music Conference*, pages 142–145, 2002.
- [23] HW Nienhuys, J Nieuwenhuizen, J Reuter, and R Zedeler. Gnu lilypond–the music typesetter. *Free Software Foundation*, 2002.
- [24] Ziqiang Qian, Randy J Cavaiani, Aimee Volanski, and Timothy M Johnson. Portable digital music player with synchronized recording and display, December 19 2002. US Patent 20,020,189,429.
- [25] Gardner Read. *Music Notation, A Manual of Modern Practice*. Allyn and Bacon, Boston, 1964.
- [26] Kai Renz. An improved algorithm for spacing a line of music. pages 475–481. International Computer Music Association, 2002.
- [27] Kai Renz and Holger H Hoos. A web-based approach to music notation using guido. In *Proc. of*, volume 1001, page 98, 1998.
- [28] Perry Roland. The music encoding initiative (mei). In *Proceedings of the First International Conference on Musical Applications Using XML*, pages 55–59, 2002.
- [29] Ted Ross. *The Art of Music Engraving and Processing : A Complete Manual, Reference and Text Book on Preparing Music for Reproduction and Print*. Hansen Books, Miami, 1970.
- [30] Erik Sandberg. Separating input language and format in gnu lilypond. *Uppsala University. Department of Information Technology*, 30, 2006.
- [31] Bill Schottstaedt. Common music notation. In *Beyond MIDI*, pages 217–221. MIT Press, 1997.
- [32] Leland Smith. Score-a musician’s approach to computer music. *Journal of the Audio Engineering Society*, 20(1) :7–14, 1972.
- [33] Mike Solomon, Dominique Fober, Yann Orlarey, Stéphane Letz, et al. Déploiement des services du moteur de rendu de partitions guido sur internet. *Actes des Journées d’Informatique Musicale*, pages 42–46, 2014.
- [34] Daniel Taupin. Musictex : Using tex to write polyphonic or instrumental music. *TUGboat*, 14(3) :212–220, 1993.
- [35] Jeffrey Treviño and Craig Sapp. Automated notation of piano recordings for historic performance practice study. *Journal on Computing and Cultural Heritage (JOCCH)*, 7(3) :17, 2014.
- [36] Vladimir Viro. Peachnote : Music score search and analysis platform. In *ISMIR*, pages 359–362, 2011.