



## **Algorithmische Automatisierung komplexer Notationsregeln in MEI-XML am Beispiel von Versetzungszeichen**

Oleksii Sapov, Salzburg

DOI: 10.25366/2020.98

**Zitation:** Oleksii Sapov, „Algorithmische Automatisierung komplexer Notationsregeln in MEI-XML am Beispiel von Versetzungszeichen“, in: *Brückenschläge zwischen Musikwissenschaft und Informatik. Theoretische und praktische Aspekte der Kooperation*, in Verbindung mit der Fachgruppe Digitale Musikwissenschaft hrsg. von Stefanie Acquavella-Rauch, Andreas Münzmay und Joachim Veit (= Musikwissenschaft: Aktuelle Perspektiven. Bericht über die Jahrestagung der Gesellschaft für Musikforschung 2019 in Paderborn und Detmold, Bd. 3), Detmold, Musikwissenschaftliches Seminar der Universität Paderborn und der Hochschule für Musik Detmold, 2020, S. 91–96, DOI: 10.25366/2020.98



# Algorithmische Automatisierung komplexer Notationsregeln in MEI-XML am Beispiel von Versetzungszeichen

OLEKSII SAPOV, SALZBURG

Mehrere aktuelle musikwissenschaftliche Projekte verwenden für die Codierung von Notentexten das XML-basierte Format MEI<sup>1</sup>. Die Bearbeitung des Codes kann zwar manuell erfolgen, erfordert jedoch oft einen erheblichen Arbeitsaufwand. Daher ist es hilfreich, diesen Prozess dort zu automatisieren, wo es möglich und sinnvoll ist. An einem Fallbeispiel aus der Praxis des Projekts *Digital-interaktive Mozart-Edition*<sup>2</sup> wird gezeigt, wie Notationsregeln mit jenen der Informatik interagieren können.

Im Laufe der Konvertierung der Werke von W. A. Mozart aus dem proprietären DOX- in das MEI-Format<sup>3</sup> wurden die Versetzungszeichen auch bei jenen Noten sichtbar, für welche diese zwar gelten, aber nach den Notensatzregeln nicht sichtbar sein sollten (siehe Abbildung 1). Zum Zweck der Autokorrektur derartiger Fälle wurde das XSLT-Stylesheet<sup>4</sup> *checkAccidentals-Visibility*<sup>5</sup> entwickelt. Dem Algorithmus liegt die Annahme zugrunde, dass die Versetzungszeichen im Code grundsätzlich richtig vergeben sind und das Programm lediglich entscheiden soll, welche ggf. unsichtbar gemacht werden müssen. Diese Entscheidung ist kontextabhängig und wird durch eine Reihe von Notensatzregeln gesteuert.



Abbildung 1: Beispiel für falsch angezeigte Versetzungszeichen

Betrachten wir zunächst die folgende Regel, die als Grundregel fungiert: „Ein Versetzungszeichen gilt bis zum Ende des Takts, allerdings nur für die betreffende Tonlage“.<sup>6</sup>

1 Vgl. <<https://music-encoding.org>> (05.02.2020).

2 Vgl. <<https://dme.mozarteum.at/musik/edition>> (05.02.2020).

3 Vgl. <<https://dme.mozarteum.at/movi/de>> >> Dashboard >> Texte (05.02.2020).

4 XSLT ist eine Programmiersprache, die explizit für die Bearbeitung von XML entwickelt wurde, vgl. <<https://www.w3.org/Style/XSL>> (05.02.2020).

5 Das Tool kann hier heruntergeladen werden: <<https://github.com/ism-dme/DIME-tools>>.

6 Elaine Gould, *Hals über Kopf. Das Handbuch des Notensatzes*, deutsche Fassung von Arne Muus und Jens Berger, London 2014, S. 85.

Der Algorithmus<sup>7</sup> für diese Regel kann folgendermaßen aufgebaut werden: Jede Note, die ein Akzidens hat,<sup>8</sup> wird mit den Noten, die ihr im gleichen Takt voranstehen, verglichen.<sup>9</sup> Beispielsweise im ersten Takt des Notenbeispiels sind es die beiden Töne *fis*, welche die Akzidentien haben und überprüft werden. Zuerst wird das erste *fis* (siehe Abbildung 1: erster Takt, zweite Zählzeit, Abbildung 2: @xml:id="note\_3") geprüft. Dafür wird eine Sequenz aus den vorangehenden Noten erstellt: *a-d*.<sup>10</sup> Jede Vergleichsnote aus der Sequenz wird erst daraufhin überprüft, ob sich ihre Tonhöhe<sup>11</sup> mit jener der Kontextnote gleicht. Da *a* einen anderen @pname-Wert hat, wird die nächste Vergleichsnote geprüft – das *d*. Hier ist dieser Wert ebenso nicht gleich mit jenem Wert der Kontextnote und da es keine weiteren Noten in der Sequenz gibt, entscheidet der Algorithmus, den @accid zu belassen und wechselt zur nächsten Kontextnote (siehe Abbildung 1: das *fis* auf der vierten Zählzeit, Abbildung 2: @xml:id="note\_5"). Hier besteht die Sequenz der vorangehenden Noten aus *e-fis-a-d*. Da die erste Vergleichsnote – das *e* – wiederum einen anderen @pname-Wert hat, wird folglich mit der nächsten Vergleichsnote – mit dem *fis* – verglichen. Bei dieser Note stimmen die beiden Werte von @pname und @oct mit denen der Kontextnote überein und es erfolgt eine weitere Prüfung: Es sollte festgestellt werden, ob diese Vergleichsnote ein @accid oder @accid.ges hat und folgend, ob der Wert dieses Attributs mit dem Wert von @accid der Kontextnote übereinkommt. Hier trifft auch diese Bedingung zu; somit wird die Anweisung ausgelöst, das Attribut @accid der Kontextnote zu @accid.ges zu ändern. Die Vergleichsnote *a* und *d* werden nicht weiter überprüft.

```

<layer n="1" xml:id="layer_1">
  <beam xml:id="beam_1">
    <note dur="8" oct="4" pname="d" xml:id="note_1"/>
    <note dur="8" oct="4" pname="a" xml:id="note_2"/>
  </beam>
  <note accid="s" dur="4" oct="4" pname="f" xml:id="note_3"/>
  <note dur="4" oct="4" pname="e" xml:id="note_4"/>
  <note accid="s" dur="4" oct="4" pname="f" xml:id="note_5"/>
</layer>

```

Abbildung 2: XML-Code des ersten Takts aus dem Fallbeispiel

- 
- 7 Die Abbildung 4 zeigt ein Diagramm mit der kompletten Umsetzung des Algorithmus. Der Artikel ist jedoch aus didaktischen Gründen so aufgebaut, dass er erst mit der Beschreibung des Grundalgorithmus beginnt, weitere Regeln werden sukzessive hinzugefügt.
- 8 In MEI werden die sichtbaren Akzidentien mit dem Attribut @accid kodiert, die unsichtbaren mit @accid.ges. Vgl. <<https://music-encoding.org/guidelines/v4/elements/note.html>> (24.02.2020).
- 9 Die Note, bei der das Akzidens überprüft wird, wird künftig als *Kontextnote*, die ihr voranstehenden Noten werden als *Vergleichsnote* bezeichnet.
- 10 Die Sequenz wird in Bezug zu der Kontextnote erstellt, was einer umgekehrten Reihenfolge der Noten entspricht.
- 11 In MEI sind es die Attribute @pname und @oct.

Folgende Notensatzregel verursacht zwei weitere Anpassungen: „Wenn eine Note über den Takt hinaus gebunden wird, gilt das Versetzungszeichen nur für die Dauer der übergebundenen Note. Erscheint derselbe Ton im nächsten Takt wieder, muss er erneut ein Versetzungszeichen erhalten“.<sup>12</sup> Diese Regel würde die beiden Töne *fis* im zweiten Takt des Notenbeispiels betreffen. Um sie in dem Programm zu implementieren, wird eine Liste erstellt, die alle über den Taktstrich hinweg gebundenen Noten im ganzen Werk enthält. Somit prüft der Algorithmus zuallererst, ob die Kontextnote eine solche (übergebundene) Note ist. Wenn dies der Fall ist, wird die (oben beschriebene) Prüfung der Vergleichsnoten nicht nötig und es wird die Anweisung ausgelöst, das Akzidens unsichtbar zu machen.<sup>13</sup> In dem Notenbeispiel beträfe dies das erste *fis* im zweiten Takt. Die gleiche Liste kommt ggf. später zur Anwendung, wenn die Kontextnote selbst keine übergebundene Note ist, aber es sollte geprüft werden, ob ihr eine solche Note voransteht (siehe Abbildung 1, zweiter Takt).

```
<xsl:when test="map:get($paramsCurrent, 'accid') = $precedingAccid">
  <xsl:choose>
    <xsl:when test="@xml:id = $allTiedNotes">
      <xsl:value-of select="false()" />
      <xsl:break/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="true()" />
      <xsl:break/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:when>
```

Abbildung 3: Auszug aus dem XSLT-Stylesheet *checkAccidentalsVisibility*.

Ein weiterer Aspekt, der berücksichtigt werden muss, ist das Auftreten von Auflösungszeichen. Wenn die Kontextnote beispielsweise ein Kreuz hat und man bei der Prüfung der Vergleichsnoten auf eine Note gleicher Tonhöhe trifft, die ein Auflösungszeichen hat, dann bricht die Prüfung ab, das Akzidens bleibt sichtbar und es wird die nächste Kontextnote geprüft. In ähnlicher Weise, wenn die Kontextnote selbst ein Auflösungszeichen hat, und der Algorithmus auf

<sup>12</sup> Gould, *Hals über Kopf* (wie Fn. 6), S. 86.

<sup>13</sup> Es wird allerdings empfohlen „[a]m Beginn eines neuen Systems [...], das Versetzungszeichen einer übergebundenen Note zu wiederholen [...]“ (ebd., S. 87). Jedoch, angesichts dessen, dass die Notenzeichen aus dem MEI-Code dynamisch gerendert werden (vgl. <<https://verovio.org>> (24.02.2020)), bedeutet dies, dass die Zeilenumbrüche je nach der Notentext- beziehungsweise je nach der Bildschirmgröße stets neu generiert werden. Für die DIME-Codierungsrichtlinien wurde daher festgelegt, dass in solchen Fällen die Akzidentien als unsichtbar codiert werden. Möglicherweise könnte in Zukunft ein entsprechendes Rendering-Programm entscheiden, ob in diesen Fällen die Versetzungszeichen sichtbar gemacht werden sollten.

eine Vergleichsnote mit einem anderen Akzidens als einem Auflösungszeichen trifft, wird das Akzidens der Kontextnote sichtbar belassen.

Eine Note kann auch zwei Akzidentien haben, beispielsweise, wenn ein *f* erst ein Doppelkreuz hätte und das nächste *f* ein Auflösungszeichen und anschließend ein Kreuz. Für eine Kontextnote wird diese Prüfung ganz an den Anfang gesetzt, noch vor die Überprüfung, ob es sich um eine übergebundene Note handelt. Wenn eine Note mit zwei Akzidentien als Vergleichsnote fungiert, wird der Vergleichswert nur von dem letzten Akzidens berücksichtigt.

Schließlich ermöglicht MEI, Varianten zu kodieren. Um eine zu hohe Komplexität zu vermeiden, wird jedoch für die Prüfung nur die Hauptlesart herangezogen: Für das Element `<app>` wird daher der Inhalt von seinem Kindelement `<lem>`, für `<choice>` von dem Kindelement `<corr>` definiert. Sollten diese Kindelemente nicht vorhanden sein (beispielsweise bei zwei `<rdg>` in einem `<app>`), wird das jeweils erste Kindelement prozessiert.

Das Stylesheet *checkAccidentalsVisibility* kann grundsätzlich auf eine beliebige MEI-Datei<sup>14</sup> angewendet werden. Weitere denkbare Erweiterungen der Funktionalität sind beispielsweise das Sichtbarmachen von Akzidentien, die Prüfung der Vorzeichen, die Berücksichtigung von Mehrstimmigkeit<sup>15</sup> etc. Das Stylesheet stellt kein umfassendes Korrekturtool dar, es kann jedoch, unter Berücksichtigung der erwähnten Beschaffenheit der Daten, den Arbeitsaufwand von Editorinnen und Editoren erheblich minimieren (vgl. zum Ablauf auch das Entscheidungsbaum-Diagramm in Abbildung 4 auf der folgenden Seite).

---

14 Getestet mit MEI-Version 3.0.0.

15 Vgl. Gould, *Hals über Kopf* (wie Fn. 6), S. 86.

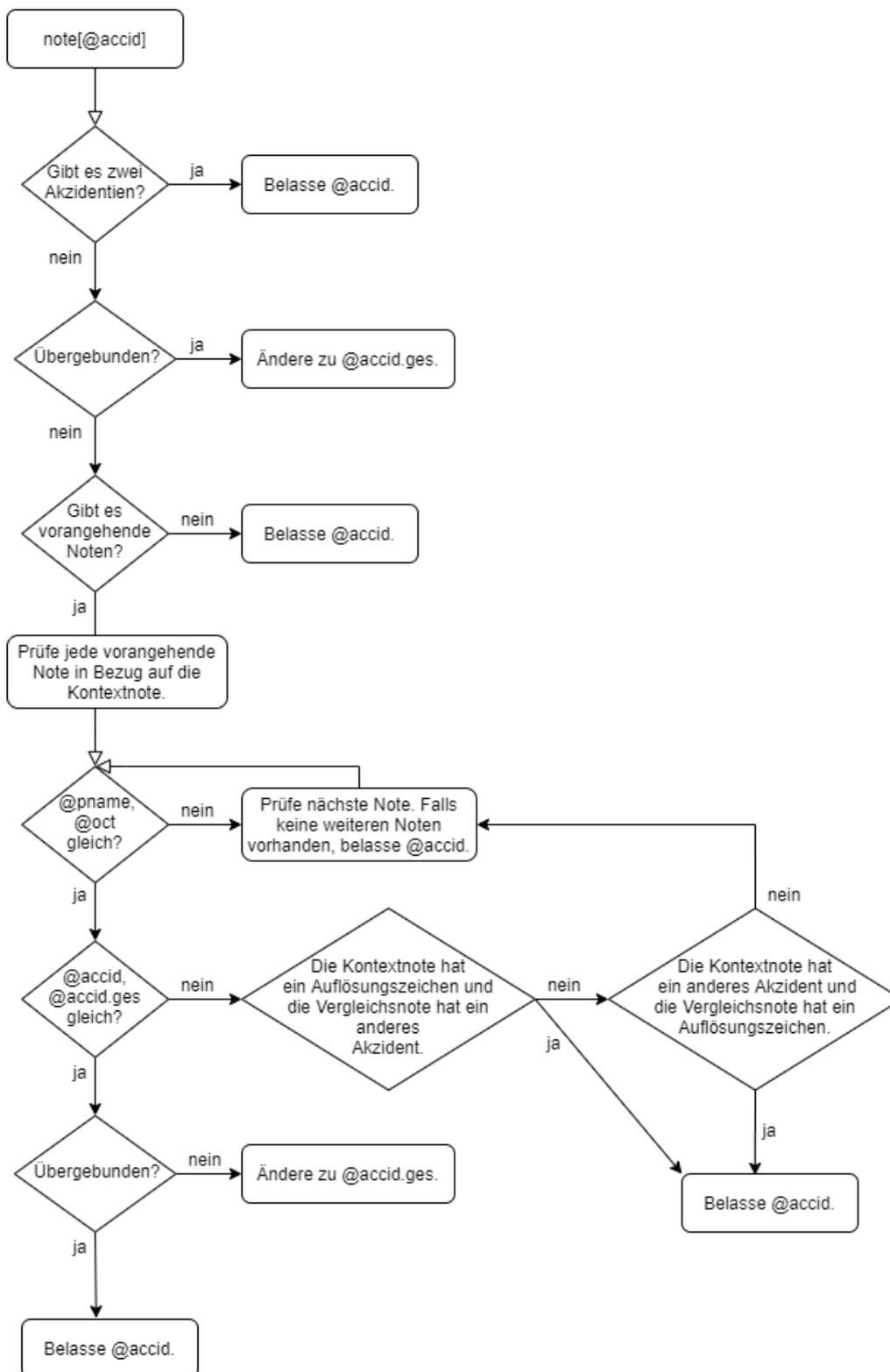


Abbildung 4: Ein Entscheidungsbaum-Diagramm [erstellt mit draw.io].  
 Es wird vorausgesetzt, dass die jeweilige Note keine Vorschlagsnote ist, dass das Akzidents der Kontextnote kein Warnakzidents ist und dass es sich um die Hauptlesart handelt.

**Zitation:** Oleksii Sapov, „Algorithmische Automatisierung komplexer Notationsregeln in MEI-XML am Beispiel von Versetzungszeichen“, in: *Brückenschläge zwischen Musikwissenschaft und Informatik. Theoretische und praktische Aspekte der Kooperation*, in Verbindung mit der Fachgruppe Digitale Musikwissenschaft hrsg. von Stefanie Acquavella-Rauch, Andreas Münzmay und Joachim Veit (= Musikwissenschaft: Aktuelle Perspektiven. Bericht über die Jahrestagung der Gesellschaft für Musikforschung 2019 in Paderborn und Detmold, Bd. 3), Detmold, Musikwissenschaftliches Seminar der Universität Paderborn und der Hochschule für Musik Detmold, 2020, S. 91–96, DOI: 10.25366/2020.98

## Abstract

Algorithmic automation of the complex music notation rules in MEI-XML through the example of the accidentals: The article demonstrates how music notation rules can be formalized into a computer algorithm. In particular, it handles the rule, whether the accidentals should be rendered or not when they repeat on the notes of the same pitch in the same measure. The decision process is described step-by-step while referring to the music notation rules. The algorithm was implemented in the XSLT-language for the needs of the Digital Interactive Mozart Edition and can be applied to MEI data. The tool can be downloaded from <<https://github.com/ism-dme/DIME-tools>>.

## Kurzvita

Oleksii Sapov (geb. 05.01.1986 in Iwano-Frankiwsk, Ukraine) schloss seine Klarinettenstudien in Kyiw und Graz mit einem Master of Arts an der Universität für Musik und darstellende Kunst Graz ab (2014). Anschließend studierte er Musikwissenschaft und Digitale Geisteswissenschaften in Graz. Seit Januar 2018 ist er wissenschaftlicher Mitarbeiter bei dem Projekt *Digital-interactive Mozart-Edition* an der Internationalen Stiftung Mozarteum in Salzburg, wo er für Lektorat und XML-Technologien zuständig ist. Für weitere Informationen siehe <<https://o-sapov.github.io/portfolio/>> (24.02.2020).



# **Brückenschläge zwischen Musikwissenschaft und Informatik**

**Theoretische und praktische Aspekte  
der Kooperation**

**Herausgegeben von Stefanie Acquavella-Rauch,  
Andreas Münzmay und Joachim Veit**

**Musikwissenschaft:  
Aktuelle Perspektiven 3**

**musiconn**  
für vernetzte Musikwissenschaft

# Brückenschläge zwischen Musikwissenschaft und Informatik

## **Musikwissenschaft: Aktuelle Perspektiven**

Bericht über die Jahrestagung der Gesellschaft für Musikforschung 2019 in Paderborn und Detmold

Herausgegeben von Rebecca Grotjahn und Nina Jaeschke

**Band 3**

# Brückenschläge

zwischen Musikwissenschaft und Informatik

Theoretische und praktische Aspekte der Kooperation

Beiträge der Symposien zur Digitalen Musikwissenschaft

Osnabrück 2018 und Paderborn 2019

im Rahmen der Jahrestagungen der Gesellschaft für Musikforschung

In Verbindung mit der Fachgruppe Digitale Musikwissenschaft

herausgegeben von

Stefanie Acquavella-Rauch, Andreas Münzmay und Joachim Veit

Detmold: Musikwissenschaftliches Seminar der Universität Paderborn

und der Hochschule für Musik Detmold

2020



DOI: 10.25366/2020.87

Online-Version verfügbar unter der Lizenz: Urheberrecht 1.0,  
<<https://rightsstatements.org/page/InC/1.0/?language=de>>

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

## **Impressum**

Redaktion: Stefanie Acquavella-Rauch, Andreas Münzmay und Joachim Veit

Satz: Nina Jaeschke und Joachim Veit

© Musikwissenschaftliches Seminar der Universität Paderborn und der  
Hochschule für Musik Detmold 2020

# INHALT

<b>Rebecca Grotjahn, Nina Jaeschke</b> Vorwort zu Band 1–3	IX
<b>Stefanie Acquavella-Rauch, Andreas Münzmay, Joachim Veit</b> Brückenschläge zwischen Musikwissenschaft und Informatik – Vorbemerkung	XI
<b>KOLLABORATIONEN – KO-LABORATORIEN</b>	
<b>Reinhard Keil</b> Der Computer als Denkzeug für hermeneutische Arbeit	3
<b>Ulrich Konrad</b> Philologie und Digitalität. Perspektiven für die Musikwissenschaft im Kontext fächerübergreifender Institutionen	27
<b>Gudrun Oevel</b> Infrastruktureinrichtungen in Forschungsprojekten – Spagat oder Chance?	35
<b>Dennis Ried</b> Erhebung, Transformation und Präsentation digitaler Forschungsdaten	41
<b>Anna Neovesky, Frederic von Vlahovits</b> IncipitSearch – Leitfaden zur Zusammenarbeit	47
<b>Elisabeth Treydte</b> Clara Schumann #digital. 40 Jahre Archiv Frau und Musik und der Start in die Digitalisierung	53
<b>TEXT/DATEN/PROZESSE</b>	
<b>Christine Siegert</b> Komponisten-Gesamtausgaben im digitalen Zeitalter: Perspektiven und Reflexionen am Beispiel Ludwig van Beethovens	61
<b>Markus Neuwirth, Johannes Hentschel, Martin Rohrmeier</b> Perspectives of Musical Corpus Studies: The Annotated Mozart Sonatas	77
<b>Agnes Amminger, Franz Kelnreiter</b> Leopold Mozarts „Gründliche Violinschule“. Zur Textcodierung und -präsentation einer digitalen Edition	83
<b>Oleksii Sapov</b> Algorithmische Automatisierung komplexer Notationsregeln in MEI-XML am Beispiel von Versetzungszeichen	91

<b>Susanne Cox, Richard Sanger</b>	
Digitale Fassungsvergleiche am Beispiel von Beethovens Eigenbearbeitungen	97
<b>Agnes Seipelt</b>	
Digitale Edition und Harmonische Analyse mit MEI von Anton Bruckners Studienbuch	105
<b>Stefanie Acquavella-Rauch</b>	
Musikalische Schaffensprozesse 2.0 – Inkorporation audiovisueller Medien der popularen Musik in Methoden der digitalen Edition	115
<b>DIGITAL(ISIERT)E MATERIALITATEN</b>	
<b>Miriam Akkermann</b>	
(Musik)Instrument (im) Computer	125
<b>Daniel Futterer</b>	
Herausforderungen bei der Kodierung von Paratext am Beispiel Neuer Musik mit Live-Elektronik	141
<b>Matthias Pasdzierny</b>	
How much is the glitch? Das digitale Paradigma als Herausforderung und Chance fur die historische Musikwissenschaft	149
<b>Shintaro Miyazaki</b>	
Musik fur Maschinen?! – Wo sich die Wissenschaft der Medien, des Computers und der Musik treffen und wie sie zusammenarbeiten konnten	173
<b>MUSIKGESCHICHTE(N) IM NETZ</b>	
<b>Matthias Tischer</b>	
Musikgeschichte der DDR: Ein Pilotprojekt zur digitalen Musikvermittlung	181
<b>Annette van Dyck-Hemming, Jan Eberhardt, Melanie Wald-Fuhrmann</b>	
Ansatze zur Analyse historischer Netzwerke mit Neo4j® – Aus der Projekt-Werkstatt der Datenbank zur Fachgeschichte der Musikwissenschaft	191
<b>Axel Beer, Martin Bierwisch, Kristina Kramer</b>	
Das MMM2 – Ein regionalgeschichtliches Onlinelexikon der Arbeitsgemeinschaft fur mittelrheinische Musikgeschichte	199
<b>Matej Santi</b>	
Was erzahlt Fritz Kreislers Geige?	207
<b>Elias Berner</b>	
Alle Menschen werden Bruder?! Ein historisches Dokument aus dem Nationalsozialismus in den sozialen Medien	211

Gabriele Buschmeier in memoriam