

„Digitale Medien und Musikedition“

Kolloquium des Ausschusses für musikwissenschaftliche Editionen der Union der deutschen Akademien der Wissenschaften. Akademie der Wissenschaften Mainz, 16. – 18. November 2006

Johannes Kepper (Detmold)

Codierungsformen von Musik

Die Codierung von Musik unterteilt sich in zwei sehr disparate Bereiche. Da aber bei den Komponisten der klassisch-romantischen Periode nur sehr bedingt Tondokumente mit Quellcharakter erhalten sind, kann für diesen Vortrag die eine Seite der Medaille, die Codierung konkreter klanglicher Ereignisse in unterschiedlichen Audioformaten, getrost übergangen werden. Die andere Seite der Medaille erweist sich bei näherer Betrachtung jedoch als weitaus komplexer: Die Zahl der verschiedenen Notationssysteme scheint keine Begrenzung zu kennen. Dabei ist jede eigenständige Form der Notation ganz konkret bezogen auf die Musik, die sie verschriftlicht. Während sie für diese Musik hochspezialisiert ist, lassen sich andere Formen der Musik bestenfalls nur sehr beschränkt darstellen. Mit Lautentabulaturen etwa lassen sich keine sinfonischen Werke darstellen, während die ja keineswegs über drei Jahrhunderte konstante (im englischen vereinfachend CWN [Common Western Notation] genannte) klassisch-europäische Notation ebenfalls kaum dazu taugt, Lautenmusik in ebenso effizienter Weise wie die darauf spezialisierten Tabulaturen wiederzugeben. Aufgrund der großen Komplexität und Disparität von Musik hat sich also im Laufe der Zeit eine Fülle von jeweils zweckgebundenen Darstellungssystemen entwickelt, die alle gleichberechtigt nebeneinander existieren. Eine sämtliche musikalisch-akustischen Ereignisse beschreibende Universalnotation konnte sich aufgrund dieser Umstände nie etablieren.

Eben diese Umstände gilt es nun bei der Suche nach digitalen Codierungsformen von Musiknotation zu berücksichtigen – auch hier wird es nicht das eine Format, den einen Code geben, der sämtliche Musik gleichberechtigt und in ihr jeweils angemessener Form speichern und wiedergeben kann. Die erste und zugleich wichtigste Aufgabe für den forschenden Musikwissenschaftler ist daher die Formulierung von konkreten Anforderungen an das potentielle Datenformat. Hierbei dienlich sind drei Fragen, deren Beantwortung zu einer angemessenen Auswahl aus der Fülle verschiedener Dateiformate führen kann.

Die erste dieser drei Fragen lautet: Wie detailliert sind die zu speichernden Daten? Bereits 1987 postulierte Llorenç Balsach die 1997 von Eleanor Selfridge-Field in *Beyond MIDI* wieder aufgegriffene Differenzierung von Notationsformaten anhand ihres enthaltenen Detailgrades¹. Auf einer ersten Ebene werden demnach Informationen zu Tondauern und -höhen gespeichert, also lediglich die einfachsten musikalischen Sachverhalte. Die zweite Ebene des Detailreichtums umfaßt zusätzlich weitere musikalische Zeichen wie etwa Bögen, Vorzeichen oder Akzente. Hier wird demnach das eigentliche semantische Umfeld der einzelnen Noten mit einbezogen. Die dritte und letzte Ebene

schließlich speichert zu jedem enthaltenen Element – Noten wie auch zusätzliche Zeichen – konkrete Positionsangaben, ermöglicht also den Druck der gespeicherten Musik. Das Konzept der nach Detailreichtum gestaffelten Notationsformate manifestiert sich etwa im MuseData-Format, das am Center for Computer Assisted Research in the Humanities der Stanford University entwickelt wurde. Dieses Dateiformat kennt in sich zwei Stufen, auf denen die einzelnen Daten angesiedelt sind.

Die zweite Frage, die vor der Auswahl eines bestimmten Formates zu beantworten ist, ist die nach dem zu codierenden Inhalt. Wie auch bei den herkömmlichen Notationssystemen ziehen verschiedene Formen von Musik unterschiedliche digitale Formate nach sich. Mittelalterliche Neumen bedürfen eines sich auf Ligaturen verstehenden Systems, moderne Partituren etwa von Ligeti oder Xenakis hingegen erfordern die Berücksichtigung von in der Romantik unbekanntem Zeichen, Formen und Anordnungen. Auch bei digitalen Notationssystemen bestimmt also der Inhalt die Form.

Die dritte und wichtigste Frage ist aber die: Zu welchem Zweck codiert werden soll? Es gibt verschiedene Anwendungsbereiche für digital gespeicherte Noten, deren Anforderungen an die Art der Speicherung teilweise konträr sind. Die einzelnen Nutzungsprofile stehen dabei gleichberechtigt nebeneinander.

Der vermutlich häufigste Einsatz von digitalem Notenmaterial dient der Aufführung von Musik, etwa durch Synthesizer, Computer oder Mobiltelefone. Ein für diesen Bereich entwickeltes Format speichert über die reinen Töne hinaus Informationen zu eingesetzten Instrumenten, Lautstärken und allen weiteren den Klang beeinflussenden Faktoren. Nicht für den Klang relevante Informationen, etwa zum notierten Schlüssel oder der transponierten Schreibung einer Klarinette werden in diesen Formaten nicht benötigt und daher nicht gespeichert.

Der neben Spielanweisungen zweite große Einsatzbereich digital codierter Musik ist der Notendruck. Ein für den Notensatz entwickeltes Format speichert präzise Informationen zu Form und Position aller enthaltenen musikalischen Zeichen. Die semantische Erschließung derselben hingegen ist für ein derart konzipiertes System nachrangig – in diesem Fall wäre also der von der niedergeschriebenen Note abweichende Klang einer Klarinette vermutlich nicht Bestandteil der zu speichernden Informationen.

Die Analyse von musikalischem Material hingegen legt den Fokus auf die semantische Bedeutung der einzelnen Zeichen, benötigt dafür aber unter der Maßgabe der Eindeutigkeit einzelner Elemente keine Informationen zur exakten Form oder Position derselben.

Für die erfolgreiche Archivierung digitaler Noten wiederum ergeben sich komplett abweichende Anforderungen. Ein für die dauerhafte Datenhaltung geeignetes Format muß robust, also wenig fehleranfällig, für längere Zeiträume stabil, vor allem aber einfach und eindeutig sein. Die gespeicherten Daten müssen in ihrer Bedeutung auch ohne ausführliche Dokumentation verständlich und unabhängig von bestimmten Programmen auch in Jahrzehnten noch leicht rekonstruierbar sein.

Ein für den Austausch zwischen mehreren unterschiedlich akzentuierten Systemen entwickeltes Format sitzt grundsätzlich immer zwischen zwei Stühlen: Einerseits muß es ein möglichst breites Spektrum an verschiedenartigen Daten ohne inhaltliche Verluste speichern können, andererseits

muß es strukturell möglichst einfach und schnell erlernbar aufgebaut sein, damit Anwender wie auch Softwareentwickler das Format nutzen bzw. Schnittstellen dafür entwickeln. Die mit fortschreitenden Möglichkeiten steigende Komplexität verschreckt hier eher die potentiellen Nutzer des Formates, die meist nur an der einfachen Konvertierung zwischen zwei bestimmten Dateiformaten interessiert sind und alle weiteren Funktionen nicht benötigen.

Eine wissenschaftliche Edition der Daten wiederum stellt die vermutlich höchsten Ansprüche an ein Datenformat. Wie bei für den Druck entwickelten Systemen sind Positionsangaben zu den einzelnen musikalischen Zeichen unverzichtbar, wie bei Analyseformaten ist aber auch deren inhaltliche Bedeutung von höchstem Interesse. Analog zu Archivformaten müssen die Daten für eine editorische Auswertung eindeutig sein, aber gleichzeitig müssen Varianten und Unsicherheiten als solche im Format speicherbar sein.

| | | | | | | |
|--------------------|------------------|----------------|---------|----------|-----------|---------|
| Positionen | | SCORE DARMS | | | MUSICXML | MEI |
| Ergänzende Zeichen | | | HUMDRUM | MUSEDATA | | |
| Notenwerte | MIDI | | | | MIDI | |
| | Spielanweisungen | Notendruck | Analyse | Archiv | Austausch | Edition |

Kombiniert man diese Fragen, so lassen sich einige bekannte, aber willkürlich gewählte Notationsformate in die sich aufspannende Matrix eintragen. Oftmals lassen sich die einzelnen Formate nicht abschließend verorten – so wurde MIDI aufgrund seiner Einfachheit über lange Jahre hinweg als das Standard-Austauschformat genutzt, obwohl die damit transferierbaren Daten bestenfalls als kümmerlich bezeichnet werden können.

Lassen Sie mich, bevor ich näher auf die Ansprüche an ein editorisches Format eingehe, noch einige Worte zur Transformation der Daten verlieren. Grundsätzlich droht bei jeder Datenkonvertierung in ein anderes Format ein Informationsverlust durch Details, die im neuen Format inhaltlich nicht

gekennzeichnet werden können. Dies läßt sich am Beispiel des beliebten Kinderspielzeuges, bei dem unterschiedlich geformte Teile in entsprechende Öffnungen einer Kiste gesteckt werden müssen, verdeutlichen. Während die zu codierenden Daten den einzelnen Teilen entsprechen, gleichen die verschiedenen Datenformate den Öffnungen. Solange Teil und Öffnung zusammenpassen, ergeben sich keine Schwierigkeiten. Auch im Vergleich zur gewählten Öffnung kleinere Körper bereiten scheinbar keine Probleme. Hierbei gilt es aber zu bedenken, daß die Öffnung nicht ausgefüllt wurde, daß neue Datenformat also nicht mit allen Informationen, die es speichern könnte, versorgt wurde. Auch der Fall eines zu großen Teiles läßt sich lösen: Man nehme eine Säge und bearbeite die Daten, bis sie in das gewünschte Format passen. Die abgesägten Überreste hingegen sind wertlos, ihr Informationsgehalt durch die unbedachte Konvertierung verloren. Bei der Digitalisierung von Noten wie auch bei jedem späteren Wechsel in andere Formate ist also höchste Vorsicht am Platze, um die investierte Arbeit nicht aus Versehen zunichte zu machen. Grundsätzlich gilt dabei, daß ein informationsreduzierender Wechsel, also das Zurechtschneiden der Daten, technisch leichter vonstatten geht. So kann man etwa MIDI-Daten verhältnismäßig einfach aus anderen Formaten generieren. Solange diese nur für eine bestimmte Aufgabe eingesetzt werden und die Originaldaten in unveränderter Form erhalten bleiben, ist dieses Vorgehen sogar zweckmäßig und sinnvoll. Auf den komplexeren Ebenen der Notation läßt sich grundsätzlich feststellen, daß eine Konvertierung aus XML-basierten Formaten wie MusicXML und MEI ebenfalls sehr einfach ist, da hier einige allgemeine XML-Techniken eingesetzt werden können. Die Qualität der Konversion hängt dabei hauptsächlich von der Dokumentation der beiden beteiligten Dateiformate ab.

Bevor ich Ihnen mit einem Beispiel die Problematik geeigneter Datenformate näher erläutere, möchte ich die Ansprüche an ein für die Edition nutzbares Dateiformat nochmals in Erinnerung rufen. Zunächst müssen in einem derartigen Format ausführliche Metadaten zur Verfügung stehen, die etwa genutzte Quellen identifizieren und so den Inhalt einer Datei umschreiben und kommentieren. Weiterhin muß es möglich sein, divergierende Lesarten und verschiedene Quellen eines ganzen Stückes, aber auch einzelner Abschnitte speichern zu können. Dabei müssen alle Unsicherheiten in der Lesung wie vorgefunden gespeichert werden können; die aufgrund seiner Erfahrung getroffenen Entscheidungen eines Editors zur Auflösung dieser Stellen müssen, als solche kenntlich gemacht, ebenfalls direkt an den entsprechenden Stellen hinterlegt werden können.

Um wissenschaftlichen Ansprüchen zu genügen, müssen die Quellen wie vorgefunden dokumentiert werden; die editorische Aufbereitung stellt erst einen zweiten Schritt dar, der bei der späteren Nutzung der Daten als solcher transparent sein muß. Dies allerdings erfordert auch die Möglichkeit, eben solche Entscheidungen durch inhaltlich mit den entsprechenden Stellen verknüpfte Kommentare begründen zu können.

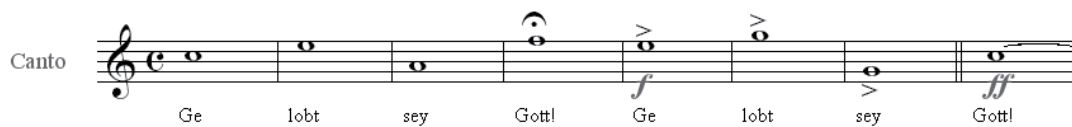
Ferner muß das zum Einsatz kommende Format Bedienfehler und Falscheingaben soweit als möglich schon im Dateiformat durch eine Validierung der Daten technisch unterbinden können. Dies darf allerdings nicht zu einer Einschränkung der Möglichkeiten führen.

Momentan bieten sich aufgrund dieser komplexen Anforderungen lediglich die beiden erwähnten XML-basierten Dateiformate für eine editorische Nutzung an. MusicXML bietet durch seine vielfältigen,

aber wenig speziellen Möglichkeiten eine gute Basis für Weiterentwicklungen. Der größte Vorteil des Formates ist aber seine sehr weite Verbreitung. Wie Michael Good bereits erwähnte, lassen sich MusicXML-Daten mit beinahe allen wichtigen Notationsprogrammen austauschen, so daß auf diesem Wege auch bereits ein großes Repertoire für eine editorische Auswertung zur Verfügung stünde. Das an die Text Encoding Initiative angelehnte MEI-Format von Perry Roland hingegen befindet sich noch in einem früheren Entwicklungsstadium, bietet allerdings bereits jetzt eine Fülle an editorisch notwendigen Funktionalitäten.

Anhand eines musikalisch trivialen Beispiels sollen die beiden Formate exemplarisch miteinander verglichen werden, um Ihnen den Aufbau beider Formate zu verdeutlichen und so eine kritische Bewertung zu ermöglichen.

Zu Beginn des vierten Satzes der (vermutlichen) Stichvorlage seiner Hymne op. 36 änderte Carl Maria von Weber die Takteinteilung der einleitenden sieben Takte. War vom Kopisten (wie auch im Autograph verzeichnet) zunächst im unbegleiteten Chor je eine ganze Note pro Takt notiert, entfernte Weber hier die Taktstriche zwischen den Takten 2 und 3 bzw. 6 und 7 durch Rasur und verlängerte die ohnehin bereits durch Fermate hervorgehobene Ganze in Takt 4 zur Doppelganzen. Als zu codierendes Beispiel möchte ich lediglich den Sopran dieser Stelle herausgreifen, der unten im Faksimile und als Übertragung vor und nach Webers Eingriff dargestellt wird.



CARL MARIA VON WEBER, HYMNE OP. 36, ALLO: MAESTOSO

Im Hinblick auf die spezifischen editorischer Anforderungen an ein Datenformat geht es bei der Codierung vor allem um die Fragestellung der Speicherbarkeit der beiden Varianten dieser Quelle. Trotz oder gerade wegen der inhaltlichen Trivialität dieses Beispiels eignet es sich gut, um die Möglichkeiten der beiden Dateiformate in diesem Bereich auszuloten.

MusicXML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE score-partwise SYSTEM "partwise.dtd">
3 <score-partwise>
4 <work>
5 <work-number>Op. 36</work-number>
6 <work-title>Hymne: In seiner Ordnung schafft der Herr</work-title>
7 </work>
8 <movement-number>4</movement-number>
9 <movement-title>Allo: maestoso</movement-title>
10 <identification>
11 <creator type="composer">Weber, Carl Maria von</creator>
12 <creator type="lyricist">Rochlitz, Friedrich</creator>
13 <rights>Copyright © 2006 Johannes Kepper</rights>
14 <encoding>
15 <encoder>Johannes Kepper</encoder>
16 <encoding-date>November 2006</encoding-date>
17 </encoding>
18 <source> Kieler Abschrift </source>
19 <miscellaneous>
20 <miscellaneous-field name="Quelle"> Erste, unbearbeitete Fassung der Quelle
21 </miscellaneous-field>
22 </miscellaneous>
23 </identification>
24 <part-list>
25 <score-part id="canto">
26 <part-name font-family="handwriting">Canto</part-name>
27 </score-part>
28 </part-list>
29 <part id="canto"> [150 lines]
180 </score-partwise>
    
```



Im obigen Ausschnitt der MusicXML-Datei sehen Sie beinahe sämtliche in diesem Format hinterlegbaren Metadaten. Innerhalb des <identification>-Elements werden Informationen zum codierten Werk, der vorliegenden Datei und auch der zugrundeliegenden Quelle gemeinsam hinterlegt, eine weitere inhaltliche Aufschlüsselung ist nicht vorgesehen und nur durch das semantisch nicht weiter spezifizierte <miscellaneous>-Element möglich.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE score-partwise SYSTEM "partwise.dtd">
3  <score-partwise>
4  <work> [3 lines]
8  <movement-number>4</movement-number>
9  <movement-title>Allo: maestoso</movement-title>
10 <identification> [13 lines]
24 <part-list> [4 lines]
29 <part id="canto">
30 <measure number="1">
31 <attributes>
32 <divisions>1</divisions>
33 <key>
34 <fifths>keine</fifths>
35 <mode>Dur</mode>
36 </key>
37 <time symbol="common">
38 <beats>4</beats>
39 <beat-type>4</beat-type>
40 </time>
41 <clef>
42 <sign>G</sign>
43 <line>2</line>
44 </clef>
45 </attributes>
46 <note>
47 <pitch>
48 <step>C</step>
49 <octave>5</octave>
50 </pitch>
51 <duration>4</duration>
52 <type>whole</type>
53 <lyric>
54 <text>Ge</text>
55 </lyric>
56 </note>
57 </measure>
58 <measure number="2"> [12 lines]
71 <measure number="3"> [12 lines]
84 <measure number="4"> [15 lines]
100 <measure number="5"> [20 lines]
121 <measure number="6"> [17 lines]
139 <measure number="7"> [20 lines]
160 <measure number="8"> [18 lines]
179 </part>
180 </score-partwise>

```



Der nächste Ausschnitt der MusicXML-Datei zeigt den Aufbau eines Taktes. Zunächst werden in einem <attributes>-Element generelle Informationen zu Takt- und Tonart sowie Schlüsselung hinterlegt. Diese Informationen müssen nicht für jeden Takt wiederholt werden, sondern behalten bis zu einer expliziten Änderung für diese Stimme ihre Gültigkeit. Die Codierung einer einzelnen Note im <note>-Element benötigt ohne weitere Informationen etwa zur Dynamik o. ä. bereits elf Zeilen Quelltext, ist dafür aber sehr intuitiv lesbar.

```

71 <measure number="3">
72 <note>
73 <pitch>
74 <step>A</step>
75 <octave>4</octave>
76 </pitch>
77 <duration>4</duration>
78 <type>whole</type>
79 <lyric>
80 <text>sey</text>
81 </lyric>
82 </note>
83 </measure>
84 <measure number="4">
85 <note>
86 <pitch>
87 <step>F</step>
88 <octave>5</octave>
89 </pitch>
90 <duration>4</duration>
91 <type>whole</type>
92 <notations>
93 <fermata/>
94 </notations>
95 <lyric>
96 <text>Gott!</text>
97 </lyric>
98 </note>
99 </measure>
100 <measure number="5">
101 <note>
102 <pitch>
103 <step>E</step>
104 <octave>5</octave>
105 </pitch>
106 <duration>4</duration>
107 <type>whole</type>
108 <notations>
109 <articulations>
110 <accent placement="above"/>
111 </articulations>
112 <dynamics>
113 <ff>
114 </dynamics>
115 </notations>

```



Speziellere Bestandteile des Notensatzes wie Fermaten oder Akzente lassen sich teilweise bereits mit Positionsangaben als Unterelement der übergeordneten Note darstellen. Das `<f/>`-Element der drittletzten Zeile offenbart eine konzeptionelle Schwäche von MusicXML: Es gibt eine abschließende Liste möglicher Dynamikangaben. Diese ist zwar sehr umfänglich und reicht vom sechsfachen piano bis zum sechsfachen forte ohne dabei seltenere Angaben wie Sforzato außer Acht zu lassen. Dennoch finden sich Besonderheiten wie die häufigen orthographischen Varianten (etwa Webers gängige Schreibweise *po*;) nicht in der Liste. Zwar wäre eine Anpassung an Webers Schreibgewohnheit möglich, allerdings wäre es utopisch zu glauben, jemals alle historischen Möglichkeiten in einer abschließenden Liste hinterlegen zu können. Es muß aber gerade in einem editorischen Speicherformat möglich sein, die in der Quelle vorgefundene Schreibung genau zu hinterlegen, in Webers Fall also das *po*;, auch wenn es inhaltlich ohne große Mühen mit einem einfachen *p* gleichzusetzen ist. Durch die Speicherung als *p* aber würde die Quelle bereits interpretiert, schon an dieser Stelle würde ein Datenverlust durch die digitale Aufbereitung entstehen – anhand der Datei ist nicht mehr nachvollziehbar, ob in der Quelle *p* oder vielleicht doch *po*: und damit z.B. der Grund für eine Fehlesung als *pp* in anderen Quellen verzeichnet war. Diese Problematik ließe sich durch einen anderen Aufbau von MusicXML leicht umgehen, dürfte aufgrund der von Michael Good zurecht gewünschten Abwärtskompatibilität seines Formates jedoch schwierig umzusetzen sein.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE score-partwise SYSTEM "partwise.dtd">
3 <score-partwise>
4 <work>
5 <work-number>Op. 36</work-number>
6 <work-title>Hymne: In seiner Ordnung schafft der Herr</work-title>
7 </work>
8 <movement-number>4</movement-number>
9 <movement-title>Allo: maestoso</movement-title>
10 <identification>
11 <creator type="composer">Weber, Carl Maria von</creator>
12 <creator type="lyricist">Rochlitz, Friedrich</creator>
13 <rights>Copyright © 2006 Johannes Kepper</rights>
14 <encoding>
15 <encoder>Johannes Kepper</encoder>
16 <encoding-date>November 2006</encoding-date>
17 </encoding>
18 <source> Kieler Abschrift </source>
19 <miscellaneous>
20 <miscellaneous-field name="Quelle"> Spätere, bearbeitete Fassung der Quelle
21 </miscellaneous-field>
22 </miscellaneous>
23 </identification>
24 <part-list>
25 <score-part id="canto">
26 <part-name font-family="handwriting">Canto</part-name>
27 </score-part>
28 </part-list>
29 <part id="canto"> [160 lines]
190 </score-partwise>

```



Die Problematik der zwei Lesarten des Beispiels läßt sich innerhalb einer MusicXML-Datei nicht darstellen, man muß also für die korrigierte Fassung eine zweite, komplett eigenständige Datei anlegen. Der Hinweis auf die veränderte Fassung läßt sich lediglich im bereits erwähnten freien <miscellaneous>-Element hinterlegen.

```

29 <part id="canto">
30 <measure number="1">
31 <attributes>
32 <divisions>1</divisions>
33 <key>
34 <fifths>keine</fifths>
35 <mode>Dur</mode>
36 </key>
37 <time symbol="common">
38 <beats>8</beats>
39 <beat-type>4</beat-type>
40 </time>
41 <clef>
42 <sign>G</sign>
43 <line>2</line>
44 </clef>
45 </attributes>
46 <note>
47 <rest/>
48 <duration>4</duration>
49 </note>
50 <note>
51 <pitch>
52 <step>C</step>
53 <octave>5</octave>
54 </pitch>
55 <duration>4</duration>
56 <type>whole</type>
57 <lyric>
58 <text>Ge</text>
59 </lyric>
60 </note>
61 </measure>
62 <measure number="2">
63 <note>
64 <pitch>
65 <step>E</step>
66 <octave>5</octave>
67 </pitch>
68 <duration>4</duration>
69 <type>whole</type>
70 <lyric>
71 <text>lobt</text>
72 </lyric>
73 </note>

```



Im oben rot hervorgehobenen Bereich sehen Sie, wie die Taktart auf 8/4 gesetzt wird, als Symbol aber ein C angegeben werden soll - auch Weber änderte die Taktart ohne dies explizit anzuzeigen. Etwas gewöhnungsbedürftig hingegen ist die Codierung von Pausen. Bei der unteren roten Markierung wird deutlich, daß zunächst ein <note>-Element angelegt wird, welches über das <rest/>-Kindelement zur Pause erklärt wird. Auch wenn diese Konstruktion die Verarbeitung von Pausen in verschiedenen Programmen wie Finale oder Sibelius erleichtern dürfte, erscheint sie inhaltlich doch zumindest etwas fragwürdig.

```

144 <note>
145 <pitch>
146 <step>G</step>
147 <octave>4</octave>
148 </pitch>
149 <duration>4</duration>
150 <type>whole</type>
151 <notations>
152 <articulations>
153 <accent placement="above"/>
154 </articulations>
155 </notations>
156 <lyric>
157 <text>sey</text>
158 </lyric>
159 </note>
160 <barline location="right">
161 <bar-style>light-light</bar-style>
162 </barline>
163 </measure>
164 <measure number="8">
165 <attributes>
166 <time>
167 <beats>4</beats>
168 <beat-type>4</beat-type>
169 </time>
170 </attributes>
171 <note>
172 <pitch>
173 <step>C</step>
174 <octave>5</octave>
175 </pitch>
176 <duration>4</duration>
177 <type>whole</type>
178 <notations>
179 <tied type="start"/>
180 <dynamics>
181 <ff/>
182 </dynamics>
183 </notations>
184 <lyric>
185 <text>Gott!</text>
186 </lyric>
187 </note>
188 </measure>

```



Unkompliziert und leicht nachvollziehbar hingegen ist die Codierung des sich zwangsläufig zu Beginn des ursprünglichen Takt 8 ergebenden Taktwechsels – es wird einfach ein neuerliches <attributes>-Element eingefügt, welches die notwendigen Daten enthält.

Abschließend läßt sich sagen, daß MusicXML einzelne, in sich eindeutige Fassungen weitgehend vollständig und sauber wiedergeben kann. Neben einigen Schwächen im Aufbau des Formates fällt dabei vor allem der Umfang auf, den die codierten Daten einnehmen: Addiert man beide Fassungen, so kommt man auf beinahe 400 Zeilen Quelltext für lediglich acht Noten. Für die weitaus meisten Anwendungen eignet sich MusicXML aufgrund seiner Einfachheit damit sehr gut – Einfachheit ist bis heute das Hauptargument für den Einsatz von MIDI als Notationsformat. Am konkreten editorischen Problem der Darstellung von Fassungen aber scheitert MusicXML in der vorliegenden Version.

MEI

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mei SYSTEM "mei17bFull.dtd">
3 <mei version="1.7b">
4 <meihead>
5 <meiid>Variantenbeispiel Mainz</meiid>
6 <filedesc>
7 <titlestmt>
8 <title>In seiner Ordnung schafft der Herr</title>
9 <respstmt>
10 <agent>Weber, Carl Maria von</agent>
11 </respstmt>
12 </titlestmt>
13 <pubstmt/>
14 </filedesc>
15 <editorialdecl>
16 <textnote>
17 Diese Datei soll lediglich einige editorische Anforderungen an
18 Notations-Datenformate verdeutlichen. Aus diesem Grunde wurde nur ein sehr kleiner
19 Ausschnitt des behandelten Werkes codiert.
20 </textnote>
21 </editorialdecl>
22 <sourcecsc> [44 lines]
67 </meihead>
68 <work> [94 lines]
163 </mei>

```



Das an die Text Encoding Initiative angelehnte MEI glänzt mit einem an TEI angelehnten Metadatenbereich. Hier lassen sich detaillierte Informationen zur entsprechenden Datei (inklusive einer Bearbeitungshistorie), einem etwaigen Gesamtprojekt, editorischen Prinzipien, der Entstehungsgeschichte des Werkes und der einzelnen Quellen in dafür vorgesehenen Elementen hinterlegen. Ich habe mich bei der vorliegenden Datei auf einige wenige Metadaten beschränkt, um zwar die Möglichkeiten anzudeuten, aber nicht vom eigentlichen Beispiel abzulenken.

```

22 <sourcecsc>
23 <source id="KA1" label="Kieler Abschrift">
24 <titlestmt>
25 <title>
26 Hymne</lb>
27 = In seiner Ordnung schafft der Herr =</lb>
28 ...
29 </title>
30 </titlestmt>
31 <pubstmt>
32 <date certainty="unsecure" reg="1814-01-01">1814</date>
33 </pubstmt>
34 <physdesc>
35 <extent units="pages">42</extent>
36 <dimensions units="cm">36,5 * 26,7</dimensions>
37 <physmedium>An dieser Stelle lassen sich Informationen zum Papier, Wasserzeichen etc. hinterlegen.</physmedium>
38 <condition>Insgesamt gut, einige kleinere Stockflecken</condition>
39 <inscription type="Widmung">der löbl. Schweizerischen großen Musikgesellschaft</inscription>
40 <inscription type="Bibliotheksstempel">Landesbib. Kiel</inscription>
41 <provenance>
42 <bibref>
43 <persname>Familie Türke, Berlin</persname>
44 <date>1814-02-15</date>
45 </bibref>
46 <bibref>
47 <persname>unknown</persname>
48 </bibref>
49 <bibref>
50 <corpname>Auktionshaus Henrici</corpname>
51 <date>1922</date>
52 </bibref>
53 <bibref>
54 <corpname>Holsteinische Landesbibliothek, Kiel</corpname>
55 <date>1965</date>
56 </bibref>
57 </provenance>
58 </physdesc>
59 <handlist>
60 <hand id="Kiel_Kopist" initial="yes" character="geübter Schreiber" ink="dunkelbraune Tinte, breite Feder" resp="Veit, Kepper"/>
61 <hand id="Kiel_Weber" initial="no" label="Weber, Carl Maria von" character="Autographe Eintragungen von Webers Hand" ink="etwas hellbraunere Tinte mit spitzerer Feder" resp="Veit, Kepper, 2005"/>
62 </handlist>
63 </handlist>
64 <extent>Codiert wurden die ersten 8 bzw. nach Eingriff Webers 6 Takte des Soprans im vierten Satz auf Blatt 8 recto.</extent>
65 </sourcecsc>
66 </sourcecsc>

```



Besonders interessant sind die Möglichkeiten im Bereich der Quellenbeschreibung. Dort kann man neben ausführlichen und gut strukturierten Informationen zur Titelei, Entstehung und ggf. auch Publikation, genutztem Papier, Abmessungen, Zustand und Provenienz der Quelle auch eine Liste der identifizierten Schreiber einfügen, denen im späteren Datenbereich des Formates einzelne Zeichen wie Noten oder auch Bögen etc. zugeordnet werden können. Allerdings gibt es auch hier einzelne Kritikpunkte, etwa die trotz Angabe der Maßeinheit uneindeutige Speicherung der Abmessungen einer Quelle.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE mei SYSTEM "mei17bFull.dtd">
3  <mei version="1.7b">
4  <meihead> [63 lines]
68 <work>
69 <music>
70 <mdiv>
71 <parts>
72 <part>
73 <staffdef key.mode="major" key.pname="c" key.sig="0" label.full="Canto"
74 id="staff1" meter.count="4" meter.unit="4" meter.sym="common"/>
75 <app>
76 <rdg label="Ursprüngliche Fassung"> [36 lines]
113 <rdg label="Korrigierte Fassung"> [43 lines]
157 </app>
158 </part>
159 </parts>
160 </mdiv>
161 </music>
162 </work>
163 </mei>
    
```



Die Definition von Takt- und Tonart, Stimmbezeichnung etc. geschieht innerhalb einer MEI-Datei erheblich kompakter als bei MusicXML. Das rot markierte <app>-Element enthält die beiden Lesarten, also den Apparat des Ausschnitts. Die Speicherung varianter Lesarten ist dabei auf mehreren Ebenen möglich – neben einer vollständigen Speicherung zweier Lesarten wie hier lassen sich auch einzelne Takte oder andere, beliebig wählbare Ausschnitte nebeneinander stellen.

```

73 <staffdef key.mode="major" key.pname="c" key.sig="0" label.full="Canto"
74 id="staff1" meter.count="4" meter.unit="4" meter.sym="common"/>
75 <app>
76 <rdg label="Ursprüngliche Fassung">
77 <staff def="staff1">
78 <measure n="1">
79 <note pname="c" oct="5" dur="1" syl="Ge"/>
80 </measure>
81 <measure n="2">
82 <note pname="e" oct="5" dur="1" syl="lobt"/>
83 </measure>
84 <measure n="3">
85 <note pname="a" oct="4" dur="1" syl="sey"/>
86 </measure>
87 <measure n="4">
88 <note pname="f" oct="5" dur="1" syl="Gottl" fermata="above"
89 />
90 </measure>
91 <measure n="5">
92 <note pname="e" oct="5" dur="1" syl="Ge" >
93 <artic value="acc"/>
94 </note>
95 <dynam staff="1">f</dynam>
96 </measure>
97 <measure n="6">
98 <note pname="g" oct="5" dur="1" syl="lobt">
99 <artic value="acc"/>
100 </note>
101 </measure>
102 <measure n="7">
103 <note pname="g" oct="4" dur="1" syl="sey">
104 <artic value="acc"/>
105 </note>
106 </measure>
107 <measure n="8" left="dbl">
108 <note pname="c" oct="5" dur="1" syl="Gottl" tie="1"/>
109 <dynam staff="1">ff</dynam>
110 </measure>
111 </staff>
112 </rdg>
113 <rdg label="Korrigierte Fassung"> [43 lines]
157 </app>
    
```



Der obige Ausschnitt enthält die vollständige Transkription der ursprünglichen Fassung unseres Beispiels. Dabei wird deutlich, daß einzelne Noten erheblich kompakter, nämlich innerhalb einer einzigen Zeile codiert werden. Dabei lassen sich Fermaten und andere Elemente wahlweise direkt als Attribut innerhalb eines <note/>-Elements integrieren oder explizit als Kindelement desselben mit eigenen Attributen genauer spezifizieren. Mit dem <dynam>-Element bietet MEI einen robusteren Ansatz zur Speicherung dynamischer Angaben als MusicXML, da hier ein beliebiger Text, also auch Webers *po: gespeichert* werden kann.

```

75 <app>
76 <rdg label="Ursprüngliche Fassung"> [36 lines]
113 <rdg label="Korrigierte Fassung">
114 <section>
115 <staffdef id="staff1 a" key.mode="major" key.pname="c" key.sig="0" label.full="Canto"
116 meter.count="2" meter.unit="1" meter.sym="common"/>
117 <staff def="staff1 a">
118 <measure n="1">
119 <rest dur="1" hand="KA1"/>
120 <note pname="c" oct="5" dur="1" syl="Ge"/>
121 </measure>
122 <measure n="2">
123 <note pname="e" oct="5" dur="1" syl="lobt"/>
124 <note pname="a" oct="4" dur="1" syl="sey"/>
125 </measure>
126 <measure n="4">
127 <note pname="f" oct="5" dur="breve" syl="Gott!"
128 fermata="above"/>
129 </measure>
130 <measure n="5">
131 <rest dur="1"/>
132 <note pname="e" oct="5" dur="1" syl="Ge" >
133 <artic value="acc"/>
134 </note>
135 <dynam staff="1">f</dynam>
136 </measure>
137 <measure n="6">
138 <note pname="g" oct="5" dur="1" syl="lobt"/>
139 <artic value="acc"/>
140 </note>
141 <note pname="g" oct="4" dur="1" syl="sey">
142 <artic value="acc"/>
143 </note>
144 </measure>
145 </staff>
146 </section>
147 <section>
148 <staffdef id="staff1 b" meter.count="4" meter.unit="4" meter.rend="invis"/>
149 <staff def="staff1 b">
150 <measure n="8" left="db"/>
151 <note pname="c" oct="5" dur="1" syl="Gott" tie="i"/>
152 <dynam staff="1">ff</dynam>
153 </measure>
154 </staff>
155 </section>
156 </rdg>

```



Dieser Ausschnitt wiederum enthält die zweite, also korrigierte Fassung des Beispiels. Durch den Taktwechsel wird hier die Aufteilung desselben in zwei Sektionen notwendig, die jeweils einen eigenen Abschnitt mit Angaben zur Taktart etc. enthalten. Hier manifestiert sich eine der Schwächen von MEI: Bedingt durch die vielfältigen Möglichkeiten des Formates gibt es auch verschiedene Positionen, an denen ein solches <staffdef/>-Element gespeichert werden kann. Das zwar weniger mächtige, aber dafür unkompliziertere <attributes>-Element aus MusicXML ist erheblich leichter handhabbar, da es sehr intuitiv nutzbar ist, während man bei MEI an dieser Stelle bereits auf die Dokumentation des Formates angewiesen ist.

Die von Weber hinzugefügte Pause im ersten Takt läßt sich über das *hand*-Attribut auf diesen zurückführen. Die Pause als solche wird gegenüber MusicXML erheblich intuitiver direkt als <rest/>-Element gespeichert.

Auch bei MEI finden sich noch etliche Schwierigkeiten, die es bis zu einem produktiven Einsatz des Formates zu beheben gilt. Wie auch MusicXML befindet es sich aber in fortwährender Weiterentwicklung und wird von seinem Entwickler an die Wünsche der Benutzer nach Möglichkeit angepaßt. Allerdings erscheint MEI von seinen bereits enthaltenen Möglichkeiten her erheblich besser auf spezifisch editorische Anforderungen eingestellt zu sein als MusicXML.

Fazit

Lassen Sie mich zum Schluß kommen. Wie sich gezeigt hat, gibt es nicht das eine Dateiformat, welches alle verschiedenen Ansprüche befriedigen kann. Für die erfolgreiche Verwendung der zu codierenden Daten ist also die Auswahl eines angemessenen Formates essentiell. Für unseren Bereich, also die wissenschaftliche Edition von Musik, bieten sich momentan zwei Formate an. MusicXML besticht durch seine hohe Verbreitung, ist von seinen technischen Möglichkeiten aber im aktuellen Zustand wenig auf editorische Belange ausgerichtet. MEI hingegen enthält bereits sehr viele der benötigten Möglichkeiten. Das Format ist allerdings erheblich komplexer und dadurch auch fehleranfälliger als MusicXML. Ein weiterer Nachteil an MEI ist, daß es bislang meines Wissens kein einziges Programm gibt, welches MEI-Dateien direkt generieren oder bearbeiten könnte.

Die Integration von Varianten in MusicXML würde den Rahmen des Formates vermutlich sprengen, zumal steigende Komplexität für ein Austauschformat wie MusicXML eher hinderlich ist. Die Aufteilung der Varianten in mehrere Dateien hingegen ist wenig zweckmäßig; schon bei zwei nicht in Verbindung stehenden varianten Stellen innerhalb eines Werkes entstehen theoretisch vier Kombinationsmöglichkeiten. Die Ausgliederung nur einzelner abweichender Takte in externe Dateien, welche per Link eingebunden werden können, erzwingt wiederum eine enorm komplexe Rekonstruktion der Daten, die aus Gründen der Datenhaltbarkeit wenig praktikabel erscheint.

Insgesamt gesehen erscheint es mir daher am sinnvollsten, MEI als Format der Wahl für editorische Arbeiten weiterzuentwickeln und gemeinsam mit Perry Roland an diese Anforderungen anzupassen. Gleichzeitig aber wäre es wünschenswert, auch MusicXML dahingehend weiterzuentwickeln, daß ein Datenaustausch zwischen MusicXML und einzelnen editorisch begründeten Lesarten innerhalb einer MEI-Datei weitgehend verlustfrei möglich wird. So könnten etwa einzelne Varianten oder komplette Fassungen in Finale oder Sibelius erstellt und über MusicXML an MEI weitergegeben werden.

Neben der Weiterentwicklung beider Dateiformate ist aber gleichzeitig die Entwicklung eines Programmes notwendig, das als Oberfläche für MEI-Daten fungiert und etwa die Lesarten auch als solche visualisieren kann. Der Einsatz eines komplexen Formates wie MEI, aber auch MusicXML ist nur möglich, wenn Sie als Editoren die eben gezeigten Quelltexte nie wieder für Ihre editorische Arbeit sehen müssen. Im Rahmen des Edirom-Projektes ist der Einsatz von ad-hoc-Notensatz für editorische Zwecke als Bestandteil unserer zweiten Projektphase eingeplant, daher wird in diesem Rahmen auch ein solches Programm, welches zumindest eine vereinfachte Bildschirmdarstellung varianter Lesarten ermöglichen sollte, entwickelt werden. In Druckreife werden die mit diesem bearbeiteten Daten sicherlich nicht vorliegen, aber da in gedruckter Form ohnehin jeweils nur eine Fassung dargestellt

werden kann, ist hier wiederum der Export über MusicXML hin zu Score oder anderen Satzsystemen leicht vorstellbar. Ein solches von uns entwickeltes Programm würde wie auch die gesamte Edirom unter einer offenen Lizenz kostenlos zur Verfügung gestellt werden.

Um aber den wirklichen Bedarf besser einschätzen zu können, bedarf es des regelmäßigen öffentlichen Diskurses aller beteiligten Wissenschaftler. Nur so kann sichergestellt werden, daß der Fokus eines solchen Projektes nicht zu eng gewählt wird und man nach Möglichkeit den Ansprüchen verschiedener Editionen gerecht werden kann.