

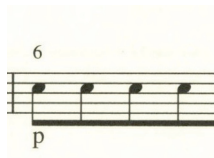
# Proposal for encoding Figured Bass in MEI

by Maja Hartwig and Johannes Kepler, 2010-15-11

In the following we propose a model for encoding figured bass within MEI. Currently, MEI supports figured bass only with its <harm>-element, which is quite unspecific and does not even cover all „standard figures“ of figured bass. Instead of reworking the existing <harm>, we modelled a new element called <fb>, which contains <figure> child elements for each contained number. The names of these elements are only quick choices; if there are better names in the english / american terminology, they can be easily replaced. Regarding attribute names, we tried to reuse existing attributes as much as possible. Whenever we pick an existing attribute name, we intend to include it with all its specs.

We do not provide yet a complete schema for the elements proposed in here, but instead offer a small number of examples that shall demonstrate this concept. We'd like to invite you to bring further examples into the discussion, which will hopefully lead to an improved concept of figured bass in MEI.

## Example 1



```
<measure>
  <staff>
    <layer>
      <beam>
        <note dur="8"/>
        ...
      </beam>
    </layer>
  </staff>
  <dynam staff="1" tstamp="1">p</dynam>
  <fb staff="1" tstamp="1" place="above">
    <figure num="6"/>
  </fb>
</measure>
```

The first example demonstrates the basic principles of our proposed model. <fb> is allowed within <measure> (MEI:CMN), and thereby replaces <harm>. It is assigned to a staff, a timestamp and is placed above or below the referred staff (optional). Other reasonable (but optional) attributes for <fb> are: @layer, @copyof, @sameas, @corresp, @chorddef.

**Discussion:** It is to be discuss, whether <fb> should allow other placing mechanisms like @plist, @startid / @endid. A reasonable alternative could be to introduce @refnote (xs:IDREF), but this depends on the decision made for Example 9. The same applies to the optional @dur on <fb>.

## Example 2



```
<fb staff="1" tstamp="1" place="above">
  <figure num="7"/>
  <figure accid="n"/>
</fb>
```

The second example illustrates the concept for <figure>s included in <fb>. Each symbol contained in a figured bass stack is encoded using a <figure> element. This might contain @num and / or @accid.

**Discussion:** If the encoder wants to clarify the meaning of the natural sign, an additional attribute @num.vis might be used to indicate which note is affected by the accidental:

```
<figure num="3" num.vis="false" accid="n"/>
```

### Example 3



```
<fb place="below">
  <figure num="7" accid="f"/>
  <figure num="5"/>
</fb>
```

This example combines @num and @accid. The @accid.pos introduced in Example 5 defaults to „after“, so it is not encoded here.

### Example 4



```
<fb>
  <figure num="6"/>
  <figure num="4" accid="s" accid.pos="within"/>
  <figure num="3" accid="n" accid.pos="before"/>
</fb>
```

Example 4 deals with accidental positions. The „lowest“ entry has a natural, which is written before the number. As the default position for accidentals is after the number, @accid.pos is used to indicate its real position. The middle entry shows a very comparable situation: The sign written in the source is nothing else than a tightly drawn „4“ with a sharp, which over time established its own symbol. As this symbol is fairly well standardized for the usual numbers (2, 4, 5, 6, 7), encoding it by using @accid.pos=“within“ seems adequate for the default situation. If the source uses a less common symbol, Example 10 offers a concept for encoding it.

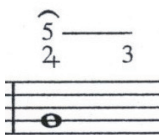
### Example 5



```
<fb>
  <figure num="6" accid="s" accid.pos="within"/>
</fb>
```

In this case, the number is slashed in the source, which is again an accidental mixed into the number, so the encoding uses the same mechanism as seen before.

### Example 6



```
<fb tstamp="1">
  <figure num="5" accid="f" extender="true" dur="0m+3" sym="predef.symbol"/>
  <figure num="2" accid="s" accid.pos="within" next="following3"/>
</fb>
<fb tstamp="3">
  <figure xml:id="following3" num="3"/>
</fb>
```

On the first beat, we see a fairly complex <fb>. The curved 5 is an individual symbol used by C.P.E. Bach which indicates a diminished fifth. This information is captured using @num=“5“ and accid=“f“. As this is an uncommon sign, @sym is used to refer to a predefined symbol which resembles the layout of this entry. (It is to be discussed whether it is necessary to hide the flat by using a then needed @accid.vis)

## Example 7



```
<fb tstamp="1.5">
  <figure extender="true" dur="0m+1.5" prev="previousFB"/>
</fb>
```

Sometimes a `<fb>` contains nothing else than a stroke indicating to hold a chord / harmony. Using only `@extender`, `@dur` (and optionally `@prev` / `@next`) matches this situation.

## Example 8



Solution 1

```
<fb tstamp="1">
  <figure num="7" next="added5"/>
  <figure accid="n" next="addedExtender"/>
</fb>
<add resp="MH">
  <fb xml:id="addedExtender" tstamp="1.5">
    <figure extender="true" dur="0m+3.5"/>
  </fb>
</add>
<add resp="MH">
  <fb xml:id="added5" tstamp="2">
    <figure num="5" accid="s" accid.pos="within" next="existing6"/>
  </fb>
</add>
<fb tstamp="3">
  <figure xml:id="existing6" num="6"/>
</fb>
```

Solution 2


```
<fb tstamp="1">
  <figure num="7" next="added5"/>
  <figure accid="n" next="addedExtender"/>
</fb>
<fb tstamp="1.5" dur="0m+3.5">
  <add resp="MH">
    <figure extender="true" dur="0m+3.5"/>
  </add>
  <add resp="MH">
    <figure xml:id="added5" tstamp="2" num="5" accid="s"
      accid.pos="within" next="existing6"/>
  </add>
  <figure xml:id="existing6" tstamp="3" num="6"/>
</fb>
```

Solution 1 uses multiple `<fb>`s, one for each timestamp that holds new symbols. This leads to three more `<fb>`s for all figures following the first figure. Solution 2 recognizes all these symbols as part of one harmony which is splitted over several notes. This allows to recognize passing or grace notes as part of a greater harmonic construction. At the same time, it is far more analytical than the first solution.

In terms of schema layout, the difference between the two solutions is that in the first solution only the `<fb>` has a `@tstamp` which applies to all included `<figure>`s, whereas in the second solution each `<figure>` can have its own `@tstamp`. In either solution `<figure>` needs a `@dur`, as seen in example 7 before. Likewise, `@next` and `@prev` should be available to allow explicit information about the „layers“ of those figures. This means that the whole question comes down to whether we want to allow individual `@tstamps` on the `<figure>`s contained within one `<fb>`. If we decide so, this does not permit to encode like in the first solution, but it offers greater flexibility (Solution 1 is therefore a subset of Solution 2). Example 6 could be encoded both ways as well.

Another issue raised here is the question whether `<fb>` (or maybe `<figure>`?) should have a `@refnote`. Using `@startid` implies that there is an `@endid`, and `@plist` seems wrong, as there should be only one element on that list.

## Example 9



Example 9 shows a musical staff with a treble clef and a key signature of one flat (B-flat). The staff contains a sequence of notes: a half note G4, a quarter note A4, a quarter note B4, and a half note C5. Below the staff, there are fingerings: 'b3' for the first note, '6' and '5' for the second, and '7' and '#3' for the third. To the right of the staff is XML code for the first measure.

```
<fb tstamp="3.5">
  <figure num="3" accid="f" accid.pos="before"/>
  <figure xml:id="first6" num="6" next="extender6-7"/>
  <figure num="5" next="s3_t4_e3"/>
</fb>
fb tstamp="4">
  <figure xml:id="extender6-7" prev="first6" next="last7" extender="true" dur="0m+4"/>
  <figure xml:id="s3_t4_e3" accid="s" accid.pos="before" extender="true" dur="0m+4.5"/>
</fb>
fb tstamp="4.5">
  <figure xml:id="last7" num="7"/>
</fb>
```

This example might help in the discussion of `@refnote`. The dash between the 6 and 7 implies that the notes indicated by the 6 (in that case an A) should be hold on tstamp 4. Considering the bass note this means it is no longer a 6 but instead a 5 (C-Ef-G-A -> D-Fs-A). Of course this could be followed by determining that „extender6-7“ has no `@num` by its own, looking for its `@prev` (or any `@next` pointing to it), and figuring out the note to which that `<fb>` is related. If we describe this behaviour in the Guidelines explicitly, that’s a perfectly valid way for encoding such situations, and additionally, it doesn’t require much insight from the encoder. Nevertheless, it seems quite cumbersome to get this corresponding note. It might be a better solution – for applications as well as human readers – if the `<figure>` would have a pointer `@refnote` that directly hints to the C on tstamp 3.5. Another possibility would be to encode it like this:

```
<figure xml:id="extender6-7" num="5" num.vis="false" prev="..." next="..." extender="true" dur="0m+4"/>
```

This would be closer to the musical „meaning“ of the symbols (whatever that might be), but it seems to be rather far away from the symbols itself. We suggest therefore to use the existing `@chorddef` (currently available on `<harm>`, should be available for `<fb>` as well), which allows to capture the harmony in the gestural domain (as Perry / MEI calls it).

## Example 10



Example 10 shows three musical staves. The first staff has a treble clef and a key signature of one flat, with notes G4, A4, and B4. Below the staff are fingerings: '8', '6', '4', '2' for the first note, '6', '4', '3' for the second, and '5', '3' for the third. The second staff has a treble clef and a key signature of one flat, with a single note G4. Below the staff is a fingering: '8'. The third staff has a treble clef and a key signature of one flat, with a single note G4. Below the staff is a fingering: '5'.

This example illustrates the issue that the symbols for `@accid.within` are not as standardized as we want them to be (of course they are not...). There are several ways to solve this problem. Solution 1: Ignoring these differences completely seems to be a valid option for an encoder, but not for the schema itself (there might be projects interested in recording these differences). Solution 2: Referring those projects self-defined in the header to `@sym` is a solution, but again it is only a solution for individual projects – you can’t compare encodings from different projects, as they probably will use different identities for symbols that might be the same. Solution 3 is to offer a complete list of all possible symbols for `@accid.within`, which could be referred from a dedicated attribute (`@accid.sym?`). This seems impossible for practical reasons: There are lists with symbols, but we will never match everything that’s out there in prints or, even worse, manuscripts. Solution 4 seems to be a good compromise: The format itself retrieves to the possibilities described in this document, which allow to capture the symbols on a logical level. `@sym` can be used to point to symbol definitions (like in Solution 2). But instead of pointing to individual definitions, MEI offers an extensible list of symbols together with the IDs to be used for them in the Guidelines / Tag Library. This makes perfectly sense if / when MEI switches from IDREFs to XPointers, as has been proposed before. In that case every project can directly refer to a common entity defined at a public spot. If there is no adequate symbol, it needs to be added to the list, which means it’s publicly available from then. The benefit of this solution is that it allows interchange between different projects without complicating things in the model itself.