Method XML

Version 1.0, 15 March 2008

Editor:

Peter Niblett

Abstract:

This specification defines an XML format (Method XML) for conveying information about methods. It defines XML elements that can be used to carry information about a single method, to carry information about a set of methods sharing some common properties, and also to represent a collection of methods.

The specification describes the structure and meaning of these XML elements. It includes an XML Schema, which provides a set of constraints on the syntax of the XML elements. As it is not practical to provide a complete set of constraints in XML schema, this specification contains a set of further validation constraints expressed in English.

A working knowledge of XML and XML Schema is assumed.

Status:

This is the final draft of this version

Table of Contents

1 Introduction

Electronic information about ringing methods is currently communicated using a number of different formats. This specification defines a standard format for conveying information about an individual method, and also allows for information about multiple methods to be grouped together into method collections.

1.1 Goals and Requirements

The main purpose of this specification is to provide a consistent data format that can be used by any application that is storing or exchanging computer-readable information about methods.

Typical uses might include:

- A machine-readable method collection
- A web service message that is passing information about a method (such as its name and place notation) to or from a service

This specification aims to meet the following requirements:

- The format must be able to represent all information in the Central Council method collections, but not be limited to the information that is present in today's collections.
- The format must be extensible, so that arbitrary application-specific classification or other data can be added in a straightforward manner.
- The format must support incomplete data. When two particular applications are exchanging information about a method, they might only be interested in some specific items of information (for example the method's name and the location of the first peal in that method), rather than all known information at that method. Users of the format should be free to choose how much or how little information they include.
- The structure of the XML elements must be such that it is relatively straightforward to search a document using XPath [XPATH]. This makes it easier for someone to write an XSLT program (stylesheet) that selects information from a Method XML document and converts it into some other format, such as HTML.

1.2 Overall Structure

This specification declares three primary elements: `method`, `methodSet` and `collection`. These are defined as Global Element Definitions in the XML Schema and are intended for use by applications either as the roots of freestanding XML documents, or as elements inside other XML documents or messages. In addition the schema defines complex types corresponding to each of these three elements. This means that applications can use application-specific names for these elements.

The `method` global element contains data concerning a single method. Its associated type is called `methodType`.

The `methodSet` global element (or its type `methodSetType`) can be used to group together a number of `method` elements whose methods have some properties in common. Although it can be used as the top-level element in a document, its main use is to group together methods that appear in a `collection`.

The `collection` global element (or its type `collectionType`) can be used in a document that contains a collection of methods, possibly with quite different properties. The `collection` element contains 0 or more `methodSet` elements.

1.3 Optionality

The `method`, `methodSet` and `collection` have complex type definitions which contain several child elements. The majority of these child elements can be omitted from a particular instance of the parent element. There are several situations when a child element could be omitted:

- A child element must be omitted in cases where it would have no valid value, for example if a method has no hunt bells then its `method` element must not contain a `huntbellPath`.
- If a `method` is contained in a `methodSet` element then properties of that method can be specified at the level of the `methodSet` rather than the `method` itself. For example a property such as stage, that applies

to all the methods in the `methodSet` need only be specified at the `methodSet` level and does not need to be included inside each individual `method`. Conversely if a given property is specified on each method, it would be normal to omit it from the `methodSet` element.

- A child element can be omitted if the creator of the parent does not wish to include the information that it would have carried. For example falseness information might well be omitted if a method element is being used solely for the purpose of passing method information to a blue-line drawing program.

This specification does not define any "default" values for omitted elements. So if an element is omitted both from `method` and the containing `methodSet` (if there is one) then a processor of the `method` element should not infer any implicit values for these omitted elements. In addition we note that the schema also permits the child elements that are included to appear in any order; no significance is implied by the order in which they appear.

1.4 Extensibility

This specification includes a number of extensibility points which allow additional information to be included. Users of this specification are free to define additional XML elements, in a separate XML namespace, to insert in these extensibility points:

- The `method`, `methodSet` and `collection` elements can each contain a `notes` element which can be used to carry additional descriptive information.
- The `method` and `methodSet` elements can each contain a `meta` element which can be used to carry additional metadata elements, for example alternative classification schemes
- Additional `reference` elements can be used in place of or in addition to the reference elements defined in this specification. For example a Guild or Association could include a reference to its Annual Report. This extensibility point makes use of XML Schema Substitution Groups.
- Additional `performance` elements can be used in place of or in addition to the performance elements defined in this specification. This extensibility point makes use of XML Schema Substitution Groups.

In addition, the `method`, `methodSet` and `collection` elements each permit the addition of user-defined attributes to the elements themselves.

1.5 Versioning strategy

This version of the specification uses the following XML namespace:

http://www.cccbr.org.uk/methods/schemas/2007/05/methods

It is the intention of the Methods Committee that this namespace URI will not necessarily change at any subsequent revision of this specification, but rather it will change only if a subsequent revision results in non-backwardly compatible changes from a previously-published version. This is to minimise unnecessary disruption to users of the schema.

By "non-backwardly compatible" we mean a change which would cause a document valid against the earlier version of the schema to become invalid, were the namespace URI to have remained the same. For example, the following kinds of change would be "backwardly-compatible" and so would **not** result in assignment of a new namespace URI:

- addition of new global element, attribute, complexType and simpleType definitions
- addition of new elements or attributes in locations covered by a previously specified wildcard
- modifications to the pattern facet of a type definition for which the value-space of the previous definition remains valid or for which the value-space of the preponderance of instance would remain valid
- modifications to the cardinality of elements for which the value-space of possible instance documents conformant to the previous revision of the schema would still be valid with regards to the revised cardinality rule

If future versions are required, they will use the following scheme

http://www.cccbr.org.uk/methods/schemas/yyyy/mm/methods

Where *yyyy/mm* gives the year and month chosen for that version of the namespace URI.

The specification includes a `decisionsYear` attribute which can be used to show which version of the Central Council Decisions was being considered when the XML document was constructed.

1.6 Character Encoding

This specification imposes no restriction on the character encoding used. Any encoding that is supported by XML may be used, however it is recommended that users create documents in UTF8 encoding.

While many Method XML documents contain just characters from the 7-bit US-ASCII character set, users are free to use more unusual characters in some of the elements defined by this specification. In particular place names, a method's `name` or a method's `title` may contain such characters, for example the superscript 2 in the method "E=mc^2 Surprise Major"

1.7 Notational Conventions

When a sentence of English text refers to an XML element, the element name is given using the Courier font. For example, the previous section includes the phrase, "a method's `title` may contain such characters". In this phrase `title` refers to an XML element.

When the use of an element is being formally defined, it is referred to using an XPATH locationPath showing its descent from its nearest Global Element ancestor, for example

`/mx:methodSet/properties/huntbellPath`

Terms that are defined in section 2 (Terminology and Concepts) are shown as blue hyperlinks, for example row, in places where they are used elsewhere in this document.

This specification uses a notational convention, referred to as "Pseudo-schemas". This syntax looks like an XML instance document, but values in italics indicate data types instead of literal values. The following syntax is used

'?' denotes optionality (i.e. zero or one occurrences),

'*' denotes zero or more occurrences,

'+' denotes one or more occurrences,

'[' and ']' indicate that the contained items form a group for the purpose of expressing cardinality or choice,

'|' represents a choice of the items on either side of the '|'.

{any} is used to indicate an XML Schema element wildcard (xs:any)

… is used to indicate an XML Schema attribute wildcard (xs:anyAttribute)

```
<!-- sample pseudo-schema -->
<element
      required_attribute_of_type_QName="xs:QName"
      optional_attribute_of_type_string="xs:string"? >
    <required_element />
    <optional_element /> ?
    <one_or_more_of_these_elements /> +
     [ <choice_1 /> | <choice_2 /> ] *
</element>
```

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC 2119].

1.8 Namespaces

The following namespaces are used in this document:

Prefix	Namespace
mx	http://www.cccbr.org.uk/methods/schemas/2007/05/methods
xml	http://www.w3.org/XML/1998/namespace
xs	http://www.w3.org/2001/XMLSchema

2 Terminology and Concepts

This specification uses the following terms, consistent with the definitions given in the Decisions of the Central Council of Church Bell Ringers, 2005 [CC Decisions]

Row:

A *row* is a permutation of bells, each bell being included once and only once.

Stage:

The number of bells used in a row or round block.

Rounds:

The row (at a given stage) in which each bell is in its home position.

Change:

A *change* is the progress from one row to the next, effected by the interchange of bells in adjacent positions in the row.

Round Block:

A *round block* is an ordered sequence of rows (each at the same stage) produced by a sequence of two or more changes where the final change in the sequence produces the initial row of the block. A round block B is said to be a *rotation* of a round block A if the sequence of changes that produces B is a rotation of the sequence of changes that produces A.

Method:

Any round block that is *true* (i.e. contains no row more than once) and is divisible into two or more equal parts (called *leads),* and has more working bells than hunt bells, defines a *method.* Such a round block is called the *plain course* of the method. Starting the plain course from a different change does not give a different method, so two round blocks that are rotations of one another both define the same method.

Lead-head and **lead-end:**

The first row in each lead is known as the *lead-head*. The last row in each lead is known as the *lead-end row*. The change following the lead-end row is known as the *lead-end change*.

Hunt bells and **working bells:**

Bells that are in the same position at each lead-head in a course are known as *hunt bells.* Bells that are not in the same position at each lead-head in a course are known as *working bells*.

Type:

The first level of classification of a method. Four types of method are defined:

- Methods with hunt bells are known as *hunters* if all the working bells do the same work in the plain course and the number of leads is the same as the number of working bells.
- Methods with no hunt bells are known as *principles* if all the working bells do the same work in the plain course and the number of leads is the same as the number of bells.
- Methods with no hunt bells are known as *differentials* if all the working bells do not do the same work in the plain course or the number of leads is not the same as the number of bells.
- Methods with hunt bells are known as *differential hunters* if all the working bells do not do the same work in the plain course or the number of leads is not the same as the number of working bells.

Palindromic symmetry:

A method is said to have *palindromic symmetry* if the same method is produced when it is rung backwards, that is when the order of the changes is inverted.

Double symmetry:

A method is said to have *double symmetry* if the same method is produced when it is *reversed*, that is when the places within each change are inverted.

Rotational symmetry:

A method is said to have *rotational symmetry* if the same method is produced when it is simultaneously reversed and rung backwards.

Title:

Every named method has a *title.* This is a string that uniquely identifies the method and is constructed from its Name, Type, Class and stage. The way in which the title is constructed is defined in [CC Decisions].

This specification also uses the following terms:

Place Notation:

A compact notation used to denote a sequence of changes (permutations). This specification uses a conventional form of place notation (limited to 33 bells) and also defines an extended form which accommodates any stage.

Method Set:

A set of methods grouped together as they have some properties in common.

Method Collection:

A file containing a collection of method definitions and associated information

Metadata:

Information about a method that does not form part of the definition of the method itself. Some metadata, for example the method's lead-head, can be calculated from the definition of method. However this is not the case for other metadata, for example details of peal performances

3 Example

We illustrate some of the concepts and the syntax of this specification by means of an example. A fuller description of the elements shown here (and additional elements not shown in this example) appears in later sections.

```
<?xml version="1.0" encoding="UTF-8"?>
<method xmlns="http://www.cccbr.org.uk/methods/schemas/2007/05/methods">
  <stage>8</stage>
  <notation>-58-16-12-38-14-58-16-78,12</notation>
  <title>Pudsey Surprise Major</title>
  <name>Pudsey</name>
  <classification trebleDodging="true">Surprise</classification>
  <symmetry>palindromic</symmetry>
  <leadHeadCode>b</leadHeadCode>
  <falseness>
    <fchGroups>BcdY</fchGroups>
  </falseness>
  <extensionConstruction>1BC/1DE</extensionConstruction>
  <references>
       <rwRef>1924/179 181 1963/372</rwRef>
  </references>
  <performances>
    <firstTowerbellPeal>
      <date>1924-03-15</date>
      <location>
       <town>Bolsover</town>
       <county>Derbyshire</county>
       <country>GB</country>
      </location>
    </firstTowerbellPeal>
    <firstHandbellPeal>
       <date>1963-05-05</date>
    </firstHandbellPeal>
   </performances>
</method>
```

This example shows how the specification can be used to encode information about a single method – in this case Pudsey Surprise Major.

The example starts with a set of elements that say what the method is and what it is called. It then follows this with some technical information about the method. It concludes with some references and information about notable performances (in this case the first Tower bell and Handbell peals). At this point we will remark that this specification allows this information to be presented in any order – we could have put the Performances information at the beginning if we had wanted to. The specification also allows any of these pieces of information to be omitted

The `stage` element indicates that this is a method on 8 bells, and the `notation` element gives its place notation. The `title` element gives the full title for the method; the first part of the title, the *name* of the method, also appears as a separate `name` element so that it can be easily searched for in a document that contains many methods. The `classification` element, as its name suggests, tells us what kind of method this is, in this case a traditional Surprise method.

The `symmetry` element states that Pudsey has conventional palindromic symmetry, and `leadHeadCode` tells us that it is a group b method. The `fchGroups` element gives its falseness using the Hodgson/Baldwin falseness groups (in this case groups B, c, d and Y), while the `extensionConstruction` element shows the extension path that has been established from Pudsey Major, through Royal and Maximus and up to higher stages.

The `references` element lets us list references to this method that appear in periodicals or other publications. In this example we have included some references to the method from *The Ringing World.* As we mentioned above, the `performances` element allows interesting or noteworthy performances to be listed. In this example we have given the dates and locations of the first peal of Pudsey in tower and in hand. Dates are given in standard XML-schema format, Year-Month-Day.

4 The collection element

A Method Collection can be encoded as an XML document by using the `collection` element as the top-level element for the document.

The collection element looks like this in pseudo-schema notation:

```
<collection date="xs:date"? uuid="xs:anyURI"? decisionsYear="xs:gYear"? …>
   <collectionName>xs:token</collectionName>?
   <notes>{any}</notes>?
   <methodSet>methodSetType</methodSet>+
</collection>
```

`/mx:collection`

The top-level element in a Method Collection document.

`/mx:collection/mx:collectionName`

The name of the Method Collection, for example “Collection of Plain Methods”. This element may be omitted.

`/mx:collection/mx:notes`

Additional descriptive notes related to the Method Collection. This specification does not constrain the format or purpose of these notes. This element is a mixed open content element, it may contain a string or further nested XML elements. This element may be omitted.

`/mx:collection/mx:methodSet`

This element contains a group of method definitions and associated properties. The `methodSet` is described in detail in the next section. A collection may contain several `methodSet` elements. This is to allow methods with similar properties (for example all the methods in the collection at a particular stage) to be grouped with each other in the collection.

`/mx:collection/@date`

The date of publication of this revision of the collection.

`/mx:collection/@uuid`

An id that uniquely identifies this collection and its revision level.

`/mx:methodSet/@mx:decisionsYear`

If present, this attribute indicates that Classification and other data in this collection reflect the Central Council decisions current at the end of the meeting held in the year specified.

`/mx:collection/@{any}`

This is an extensibility mechanism to allow additional attributes to be specified.

5 The methodSet element

The methodSet element can be used to group together a number of methods which have some properties in common. Although it can be used as the top-level element in a document, its main use is to group together method elements that appear in a collection.

The methodSet element looks like this in pseudo-schema notation:

```
<methodSet decisionsYear="xs:gYear"? …>
   <notes>{any}</notes>?
   <properties>
      <classification      little="xs:boolean"?
                           differential="xs:boolean"?
                           plain="xs:boolean"?
                           trebleDodging="xs:boolean"?>
         [Place | Bob | Slow Course | Treble Bob | Delight |
         Surprise | Alliance | Treble Place | Hybrid]?
      </classification>?
      <stage> xs:positiveInteger </stage>?
      <lengthOfLead> xs:positiveInteger </lengthOfLead>?
      <numberOfHunts> xs:nonNegativeInteger </numberOfHunts>?
      <huntbellPath> list of xs:positiveInteger </huntbellPath>?
      <leadHead> rowType </leadHead>?
      <leadHeadCode> leadHeadCodeType </leadHeadCode>?
      <falseness>
        <falseCourseHeads "fixed=fixedType">
          <inCourse> list of rowType </inCourse>
          <outOfCourse> list of rowType </outOfCourse>
        </falseCourseHeads>*
       <fchGroups affected="affectedType"?> fchGroupString </fchGroups>?
      </falseness>?
      <symmetry>list of [palindromic|double|rotational]</symmetry>?
      <extensionConstruction>extensionType</extensionConstruction>?
      <notes>{any}</notes>?
      <meta>{any}</meta>?
   </properties>
   <method>methodType</method>*
</methodSet>
```

/mx:methodSet

The methodSet element contains a properties element followed by zero or more method elements.

/mx:collection/mx:notes

Additional descriptive notes related to the methods contained in the methodSet. This specification does not constrain the format or purpose of these notes. This element is a mixed open content element, it may contain a string or further nested XML elements. This element may be omitted.

/mx:methodSet/mx:properties

A set of properties relating to the methods contained in the methodSet. These properties are all optional and may be specified in any order. Producers of Method XML documents may choose to omit a property, either because they do not wish to include that piece of information, or because they provide it as a property on the method elements contained in the methodSet. If a property is specified on the methodSet then it applies to all the methods contained in the methodSet, however the individual method definitions can supply their own property values. If this happens, the value of an element (and any attributes that it carries) in the method definition takes precedence over the value of the corresponding element (and attributes) in the methodSet/properties.

/mx:methodSet/mx:properties/mx:classification

The classification of the method by Type and Class, If the Type of the method is differential or principle then this element must be empty, if not then it must take one of the values enumerated in the pseudo-schema. In conjunction with the attributes of this element, this value gives the Type and Class of the method.

`/mx:methodSet/mx:properties/mx:classification/@little`

A Boolean value which must be present and set to “true” if the Type of the method is hunter or differential hunter and the Class of the method is one of the little classes, and false otherwise. It qualifies the value contained in the classification element itself, for example if this attribute is set to true and the value of the element is Surprise, then the actual Class of the method is Little Surprise. Omitting this attribute is equivalent to including it with its value set to “false”.

`/mx:methodSet/mx:properties/mx:classification/@differential`

A Boolean value which must be present and set to “true” if the Type of the method is differential or differential hunter, and “false” otherwise. Omitting this attribute is equivalent to including it with its value set to “false”.

`/mx:methodSet/mx:properties/mx:classification/@plain`

A Boolean value which must be present and set to “true” if the Type of the method is hunter or differential hunter and the content of the classification element is Bob, Plain or Slow Course, and false otherwise. This attribute conveys no additional classification information but it provides a convenient way for an application to identify a plain method in a collection. Omitting this attribute is equivalent to including it with its value set to “false”.

`/mx:methodSet/mx:properties/mx:classification/@trebleDodging`

A Boolean value which must be present and set to “true” if the Type of the method is hunter or differential hunter and the content of the classification element is Treble Bob, Delight or Surprise, and false otherwise. This attribute conveys no additional classification information but it provides a convenient way for an application to identify a treble dodging method in a collection. Omitting this attribute is equivalent to including it with its value set to “false”.

`/mx:methodSet/mx:properties/mx:stage`

The stage of the method.

`/mx:methodSet/mx:properties/mx:lengthOfLead`

The number of rows in a single lead of the method.

`/mx:methodSet/mx:properties/mx:numberOfHunts`

The number of hunt bells in the method.

`/mx:methodSet/mx:properties/mx:huntbellPath`

The path of the principal hunt bell, expressed as a space-separated list of the positions visited by the bell. For example, the `huntbellPath` for a Plain Minimus method could be shown as 1 2 3 4 4 3 2 1

`/mx:methodSet/mx:properties/mx:leadHead`

The first row of the second lead of the method (the first row of the first lead being rounds). The way in which the row is represented as a string is defined in section 8.

`/mx:methodSet/mx:properties/mx:leadHeadCode`

A coded representation of the first row of the second lead of the method. This coded representation is defined only for single-hunt non-differential methods with Plain Bob lead-heads and twin-hunt non-differential methods with Grandsire lead-heads.

The coding system is shown in the following table. Lead-heads for single-hunt methods are in the top left and bottom right hand sections, codes a - f and p - q are for seconds place lead-ends and codes g - m and r - s for lead-ends with no internal places. Lead-heads for twin-hunt methods are in the top right and bottom left hand sections

.

	Minor	Major	Royal	Maximus			Doubles	Triples	Caters	Cinques	
a	135264	13527486	1352749608	13527496E8T0	g	A	12534	1253746	125374968	12537496E80	g
b	156342	15738264	1573920486	157392E4T608	h	b	-	1275634	127593846	127593E4068	h
c	-	17856342	*	1795E3T20486	j	c	-	-	129785634	*	j
c1	-	-	1907856342	19E7T5038264	j1	c1	-	-	-	12E90785634	j1
c2	-	-	-	1ET907856342	j2						
d2	-	-	-	1T0E89674523	k2						
d1	-	-	1089674523	108T6E492735	k1	d1	-	-	-	120E8967453	k1
d	-	18674523	*	18604T2E3957	k	d	-	-	128967453	*	k
e	164523	16482735	1648203957	1648203T5E79	l	e	-	1267453	126849375	1268403E597	l
f	142635	14263857	1426385079	142638507T9E	m	f	12453	1246375	124638597	124638507E9	m
p	125364	12537486	1253749608	12537496E8T0	r	p	13524	1352746	135274968	13527496E80	r
p1	-	-	1297058364	1297E5T30486	r1	p1	-	-	179583624	1795E302846	r1
q1	-	-	1280694735	12806T4E3957	s1	q1	-	-	186947253	18604E29375	s1
q	124635	12463857	1246385079	124638507T9E	s	q	14253	1426375	142638597	142638507E9	s

Stages higher than Maximus follow the same coding system with the addition of further number-suffixed c,d,j,k,p,q,r,s codes. For example a twin-hunt Sextuples method with code j2 has lead-head 12AET90785634, and at stage Fourteen a seconds place method where one lead-head is equivalent to 7 leads of Plain Bob has code d3.

`/mx:methodSet/mx:properties/mx:falseness`

This element is used to convey some or all of the falseness characteristics of the method. Falseness can be expressed using false course heads and/or (for some kinds of method) false course head groups.

`/mx:methodSet/mx:properties/mx:falseness/mx:falseCourseHeads`

This element contains false course head (FCH) rows for the method. A row qualifies as a false course head if

- Any bells specified in the `falseCourseHeads/@fixed` attribute are in their home position
- The course of the method that starts from this row has at least one row in common with the plain course of the method. For the purpose of this condition the method is assumed to be rung in the rotation given by the `method/notation` element: both courses (plain and false) contain the same changes in the same order, and the first change of both courses is the first change indicated by the `method/notation` element.

The false course heads are split into two lists, one for in-course and out-of-course course heads. One or other of these two lists may be omitted – this does not mean that there are no false course heads, merely that the information is not included. To indicate that there are no false course heads with the given set of fixed bells then the element must contain an empty list.

The full row is given including any fixed bells.

It can be seen that rounds meets the conditions for being a false course head for every method, and so is omitted from the list of falseCourseHeads.

`/mx:methodSet/mx:properties/mx:falseness/mx:falseCourseHeads/mx:inCourse`

This element contains the, possibly empty, list of in-course rows (even permutations), other than rounds, that meet the conditions for being a false course head of the method. If present, it must contain all such rows.

`/mx:methodSet/mx:properties/mx:falseness/mx:falseCourseHeads/mx:outOfCourse`

This element contains the, possibly empty, list of out-of-course rows (odd permutations) that meet the conditions for being a false course head of the method. If present, it must contain all such rows.

`/mx:methodSet/mx:properties/mx:falseness/mx:falseCourseHeads/@fixed`

The set of bells that are fixed for the purposes of false course head determination. It is expressed using the same syntax as a row, but includes only the bells to be fixed, for example at stage 8 a value of '178' or ‘{1}{7}{8}’ would mean “fix the treble, 7th and 8th”.

`/mx:methodSet/mx:properties/mx:falseness/mx:fchGroups`

For certain kinds of method, it is more convenient to identify the false course heads by listing the “FCH groups” to which they belong, rather than spelling out each one individually. See Section 9 and Appendix B for details on the syntax of the `fchGroups` string and the definition of the FCH groups. Note that the fchGroups list is not governed by `falseCourseHeads/@fixed` and it is possible to include both `fchGroups` and `falseCourseHeads.`

`/mx:methodSet/mx:properties/mx:falseness/mx:fchGroups/@affected`

This attribute indicates how the FCH group letters are to be interpreted for a method with non-Plain Bob lead-heads. See Appendix B for details.

`/mx:methodSet/mx:properties/mx:symmetry`

The symmetry properties of the method. The value is a list which can contain zero or more of the values palindromic, double or rotational. Note that the nature of the definition of these symmetry types means that if any two are present then all three must be present.

`/mx:methodSet/mx:properties/mx:extensionConstruction`

If this method is related to another named method with the same Name, Type and Class in accordance with [CC Decisions] Decision (G) and the two methods differ by an even number of stages, then this element can be used to indicate the nature of the extension. It should not be used otherwise.

The value of the element comprises two sections separated by a / character where the first section gives the construction above the treble, and the second section gives the construction below. Each section starts with a number indicating the mode of the extension followed by a string of characters A..Z that gives the construction using the formula notation of Decision (G).C.2. So for example 2CD/4EF means construction CD mode-2 above the treble, extension EF mode-4 below. Note that for some methods there is more than one way to notate the same extension construction. If this is the case, then any of the equivalent representations can be used.

`/mx:methodSet/mx:properties/mx:notes`

Additional descriptive notes related to the methods in the `methodSet`. This specification does not constrain the format or purpose of these notes. This element is a mixed open content element, it may contain a string or further nested XML elements. This element may be omitted.

`/mx:methodSet/mx:properties/mx:meta`

This element provides an extensibility point which can be used to insert additional metadata. This specification does not constrain the format or purpose of this metadata. This element is a mixed open content element, it may contain a string or further nested XML elements. This element may be omitted.

`/mx:methodSet/@mx:decisionsYear`

If present, this attribute indicates that Classification and other data in this collection reflect the Central Council decisions current at the end of the meeting held in the year specified.

`/mx:methodSet/@{any}`

This is an extensibility mechanism to allow additional attributes to be specified.

6 The method element

The method element is used to convey information about a method. It can be used as a top-level element, or it can be carried inside a methodSet element. It can contain child elements that are concerned with

- Naming and Classification of the method
- Definition of the method itself (for example its place notation)
- Technical metadata that can be derived from 2, such as the leadHead element
- Other metadata (e.g. dates of first performances, references in journals)

The method element can also be used to encode information about a round block that does not meet the requirements for a method. In such cases it MUST NOT include classification or title elements.

The method element looks like this in pseudo-schema notation:

```
<method id="xs:ID"? decisionsYear="xs:gYear"? …>
   <name> xs:token </name>?
   <classification  little="xs:boolean"?
                           differential="xs:boolean"?
                           plain="xs:boolean"?
                           trebleDodging="xs:boolean"?>
         [Place | Bob | Slow Course | Treble Bob | Delight |
         Surprise | Alliance | Treble Place | Hybrid] ?
   </classification>?
   <title> xs:token </title>?
   <stage> xs:positiveInteger </stage>?
   <notation> notationType </notation>?
   <lengthOfLead> xs:positiveInteger </lengthOfLead>?
   <numberOfHunts> xs:nonNegativeInteger </numberOfHunts>?
   <huntbellPath> list of xs:positiveInteger </huntbellPath>?
   <leadHead> rowType </leadHead>?
   <leadHeadCode> leadHeadCodeType </leadHeadCode>?
   <falseness>
       <falseCourseHeads fixed="fixedType">
          <inCourse> list of rowType </inCourse>
          <outOfCourse> list of rowType </outOfCourse>
       </falseCourseHeads>*
       <fchGroups affected="affectedType"?> fchGroupString </fchGroups>?
   </falseness>?
   <symmetry>list of [palindromic|double|rotational]</symmetry>?
   <extensionConstruction>extensionType</extensionConstruction>?
   <notes>{any}</notes>?
   <meta>{any}</meta>?
   <references><ref>+</references>?
   <performances><performance>+</performances>?
</method>
```

Many of these elements also appear as methodSet/properties elements. The description of these elements is the same as that given in section 5. Recall that the presence of such an element in the method definition overrides any similarly named property of its containing methodSet.

The child elements of method may be specified in any order, and are all optional. If an element is omitted, then a program interpreting the file should use the value from methodSet/properties in its place. If methodSet/properties does not contain a value either, then the program should not assume any particular value.

/mx:method/mx:name

The Name of the method. An empty name with the attribute xsi:nil="true" is used to indicate that this method has not yet been officially named. A blank name with xsi:nil="false" is valid and is used in the case of the method “Little Bob”.

/mx:method/mx:title

The full title of the method.

/mx:method/mx:notation

The place notation for a lead of the method as described in section 8.

/mx:method/mx:references

This element is used to contain references to the method from a published journal (or other external source). This specification defines a number of standard reference elements that can appear as children of this element (these are listed below). Users can define their own references in their own namespaces and add them to the `mx:ref` substitution group

/mx:method/mx:references/mx:rwRef

A list of one more references to this method in *The Ringing World*. The first reference must consist either of a volume number or a year number followed by the / character and a page-number. Volume numbers are preceded by an upper case V character. Subsequent references may omit the year or volume number. If a reference does not include a year or volume number then its year or volume is the same as that of the preceding reference in the list.

Examples: 2004/123 145 refers to pages 123 and 145 for year 2004

V7/99 1952/45 refers to page 99 from Volume 7 and page 45 from year 1952

/mx:method/mx:references/mx:rwRef/@journal

References to the method that appear in a journal can carry an attribute that gives the official name of the journal. For the rwRef element the journal attribute (if included) must have the value "The Ringing World".

/mx:method/mx:references/mx:bnRef

A list of one more references to this method in *The Bell News and Ringers' Record*. The first reference must consist of a volume or year followed by the / character and a page-number. Subsequent references may omit the year or volume, e.g. 1902/123 145. If a reference does not include a year or volume then its year or volume is the same as that of the preceding reference in the list.

/mx:method/mx:references/mx:bnRef/@journal

References to the method that appear in a journal can carry an attribute that gives the official name of the journal. For the bnRef element the journal attribute (if included) must have the value "The Bell News".

/mx:method/mx:references/mx:cbRef

A list of one more references to this method in *Church Bells*. The first reference must consist of a volume or year followed by the / character and a page-number. Subsequent references may omit the year or volume, e.g. 1902/123 145. If a reference does not include a year or volume then its year or volume is the same as that of the preceding reference in the list.

/mx:method/mx:references/mx:cbRef/@journal

References to the method that appear in a journal can carry an attribute that gives the official name of the journal. For the cbRef element the journal attribute (if included) must have the value "Church Bells".

/mx:method/mx:references/mx:journalRef

The journalRef element provides a model that should be used for references from other journals. Users wishing to add a reference to journal other than one of those listed above should define their own element and place it in the "substitution group" defined by this element.

/mx:method/mx:references/mx:journalRef/@journal

This attribute gives the official name of the journal.

/mx:method/mx:references/mx:tdmmRef

The numerical index of this method as it appears in the Treble Dodging Minor Methods collection

/mx:method/mx:references/mx:pmmRef

The numerical index of this method as it appears in the Plain Minor Methods collection

/mx:method/mx:references/*mx:ref*

The abstract *ref* element provides a model that should be used for references from external sources other than journals. Users wishing to add a reference other than one of those listed above should define their own element, which extend this one, and place it in the "substitution group" defined by this element.

/mx:method/mx:performances

This element is used to list significant performances of the method. This specification defines a number of standard elements that can appear as children of this element (these are listed below), and users are free to add their own additional elements in their own namespaces, by making them members of the mx:performance substitution group.

/mx:method/mx:performances/*mx:performance*

The abstract *performance* element provides a model that should be to include additional kinds of performance reference. Users should define their own element, which extend this one, and place it in the “substitution group” defined by this element.

/mx:method/@id

This attribute can be used to assign an identifier to the `method`. This specification does not define the format or meaning of this identifier, other than the rules imposed by XML Schema. These are that the value of `@id` must be an NCName as defined in [XML-Namespaces], and the value of each id must be unique within the containing XML document.

/mx:method/@mx:decisionsYear

If present, this attribute indicates that Classification and other data in this collection reflect the Central Council decisions current at the end of the meeting held in the year specified.

/mx:method/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

7 Performance element

The `performance` element and is an abstract element which is used to contains information about a significant performance of the method. This specification defines a type, called `performanceType,` for this element. As it is abstract, a performance element cannot be used directly in an XML document, instead documents use concrete performance elements which extend `performanceType`. This specification defines a number of concrete performance elements, and users can define their own additional performance elements.

The Performance element looks like this in pseudo-schema notation:

```
<performance id="xs:ID"?  …>
   <date> xs:date </date>?
   <location>
      <room> xs:normalizedString </room>?
      <building> xs:normalizedString </building>?
      <address> xs:normalizedString </address>?
      <town> xs:normalizedString </town>?
      <county> xs:normalizedString </county>?
      <region> xs:normalizedString </region>?
      <country> xs:normalizedString </country>?
   </location>?
   <society> xs:normalizedString </society>?
   <references><ref>+</references>?
</performance>
```

/mx:method/mx:performances/*mx:performance*/mx:date

The date of the performance, in XML schema date format

/mx:method/mx:performances/*mx:performance*/mx:location

The location at which the performance took place.

/mx:method/mx:performances/*mx:performance*/mx:location/mx:room

The name or number of the room at which the performance took place.

/mx:method/mx:performances/*mx:performance*/mx:location/mx:building

The building in which the performance took place. In the case of a church this would be the dedication of the church if there is one.

/mx:method/mx:performances/*mx:performance*/mx:location/mx:address

The address at which the performance took place.

/mx:method/mx:performances/*mx:performance*/mx:location/mx:town

The town in which the performance took place.

/mx:method/mx:performances/*mx:performance*/mx:location/mx:county

The county in which the performance took place.

/mx:method/mx:performances/*mx:performance*/mx:location/mx:region

The state, province or other administrative region in which the performance took place.

/mx:method/mx:performances/*mx:performance*/location/country

The country in which the performance took place. ISO 3166-1 2-letter codes may be used to show the country, for example US for the United States.

/mx:method/mx:performances/*mx:performance*/mx:society

The society to which the performance was accredited.

/mx:method/mx:performances/*mx:performance*/mx:references

References to the performance. The type of this element is the type used for /mx:method/mx:references

/mx:method/mx:performances/*mx:performance*/@id

This attribute can be used to assign an identifier to the performance reference. This specification does not define the format or meaning of this identifier, other than the rules imposed by XML Schema. These are that the value of `id` must be an NCName as defined in [XML-Namespaces], and the value of each `id` must be unique within the containing XML document.

This specification defines the following concrete mx:performance elements. They all have the type `mx:performanceType` which is the same as the type of the abstract mx:performance element that has just been described.

`/mx:method/mx:performances/mx:firstTowerbellPeal`

Contains details of the first single-method tower bell peal of this method.

`/mx:method/mx:performances/mx:firstHandbellPeal`

Contains details of the first single-method handbell peal of this method.

`/mx:method/mx:performances/mx:firstInclusionInTowerbellPeal`

Contains details of the first tower bell peal (single-method or multi-method) that includes this method.

`/mx:method/mx:performances/mx:firstInclusionInHandbellPeal`

Contains details of the first handbell peal (single-method or multi-method) that includes this method.

`/mx:method/mx:performances/mx:firstTowerbellExtent`

Contains details of the first extent of this method rung on tower bells.

`/mx:method/mx:performances/mx:firstHandbellExtent`

Contains details of the first extent of this method rung on handbells.

8 Rows and Place Notation

This section describes how rows and sequences of place notation are represented.

Rows are used in several places, for example to show lead-heads. A row is represented as a string containing a sequence of bell-units with no separator character between these units. This specification allows a choice of two ways of representing these bell-units themselves, which we refer to as “Standard” and “Extended” notation. It’s possible to tell which notation is being used at any one time, since as we will see a row that uses the Extended bell unit notation starts with a { character, whereas a row in Standard notation does not.

8.1 Standard bell-unit notation

In the “standard” bell-unit notation bells one to nine are represented using the digits 1 to 9 and the symbols in the following table are used for bell numbers above nine.

Ten	**0**	**Twenty-two**	**L**
Eleven	**E**	**Twenty-three**	**M**
Twelve	**T**	**Twenty-four**	**N**
Thirteen	**A**	**Twenty-five**	**P**
Fourteen	**B**	**Twenty-six**	**Q**
Fifteen	**C**	**Twenty-seven**	**R**
Sixteen	**D**	**Twenty-eight**	**S**
Seventeen	**F**	**Twenty-nine**	**U**
Eighteen	**G**	**Thirty**	**V**
Nineteen	**H**	**Thirty-one**	**W**
Twenty	**J**	**Thirty-two**	**Y**
Twenty-one	**K**	**Thirty-three**	**Z**

Note that ten is represented by the digit 0 not the letter O. The letter I is not used because of its potential confusion with the number one. The letter O is not used because of potential confusion with 0.The letter X is not used because this could be confused with a place notation symbol. Lower case letters are also permitted and represent the same bell numbers as their upper case counterparts.

For example rounds at stage Fourteen is represented as a row as

```
1234567890ETAB
```

8.2 Extended bell-unit notation

Standard notation is limited to a maximum stage of 33. Extended notation places no limit on stage and provides a more regular way to represent bell numbers.

In Extended Notation bells are represented by using their bell number enclosed in curly braces {}. For instance the treble is represented as {1}, twelve as {12}, one-hundred-and-one as {101}.

A row in Extended Notation is represented as a sequence of these bell representations. For example rounds at stage Fourteen is represented as

```
{1}{2}{3}{4}{5}{6}{7}{8}{9}{10}{11}{12}{13}{14}
```

8.3 Place Notation

Place Notation is used to represent either a single change or a sequence of changes.

A single change is represented by listing the places made (place positions that are unaffected) in the change. The positions are shown using bell-unit notation and are listed in ascending order with no separator characters. The bell-unit notation can be either Standard or Extended, but the choice of notation must be the same for all places in the change. All places that are made must be shown, including places at the start and end of the change. The character – (hyphen) is used to represent the change in which no places are made. The character X is not permitted.

A sequence of changes is shown by concatenating the place notations for each successive change. The character . (dot) is used to separate the notations in the sequence. It may be omitted before or after the character – (hyphen) but must be inserted between each pair of successive place notations if neither of these notations is – (hyphen),

If a method does not have palindromic symmetry, then the `method/notation` element must contain the place notation for an entire lead up to and including the lead-end change.

If a method has palindromic symmetry, then the `method/notation` element may contain the place notation for an entire lead up to and including the lead-end change, but to save space, the element may instead contain two sequences of place notation separated by a comma. Each sequence is to be interpreted as a palindrome, that is to say when the last change in the sequence is reached the changes are then repeated in the inverse order starting with the penultimate change, if any.

Whatever the symmetry of the method, any rotation of the notation may be given in the value of `method/notation`, but the rotation chosen must be consistent with the values used for other properties (such as `leadHead` or `falseness`) that are affected by the choice of rotation.

In practice palindromic methods are usually shown in a rotation that either starts or ends with the lead-end change. For example Cambridge Surprise Minor is typically shown like this

```
-36-14-12-36-14-56,12
```

Here the first palindrome is made up of the lead up to, but not including, the lead-end change and the second palindrome consists of just one change, the lead-end change.

Methods like Grandsire Doubles are typically shown like this

```
3,1.5.1.5.1
```

as this is the rotation in which they are usually rung. Here it’s the first palindrome that consists of just one change.

9 False Course Head Groups

A list of all the false course heads of a method can be somewhat verbose. In his 1953 paper [FCHs] Maurice Hodgson examined the false course heads of palindromic Major methods with 178 fixed. He observed that a method's symmetry imposes some structure on its set of false course heads. In particular the false course heads occur in groups, such that if a method has one false course head from a group it must have all the other false course heads as well. By assigning a name to each possible group of false course heads, we can indicate a set of false course heads by listing the names of these groups, rather than by itemising each individual false course head. This gives a much more concise way of showing the falseness of the method; it also makes it more straightforward to compare the falseness of a pair of methods, or to match a method against a composition.

The grouping of false course heads depends on the set of lead-head and (for methods with palindromic symmetry) lead-end rows that the method possesses. While in principle the approach of grouping is applicable to any kind of method, in this specification we limit the use of FCH groups to methods that

- are at an even stage greater than 6, and
- are non-differential, with exactly one hunt bell, and
- have palindromic symmetry.

For Major methods (stage 8) the set of all false course heads (including those where 7 is not fixed) is sufficiently small that we can cover them all in 28 groups. For a Major method meeting the criteria listed above, the `falseness/fchGroups` property conveys the same information as the `falseness/falseCourseHead` property would do if it had the attribute `fixed="18"`.

For higher stages it becomes impractical to group all the possible false course heads, and so only courses with five unfixed bells are considered - for methods with Plain Bob lead-heads they are the courses which have bells 7 and above fixed. Thus for a Royal method with Plain Bob lead-heads, the `falseness/fchGroups` property conveys the same information as the `falseness/falseCourseHead` property with the attribute `fixed="17890"`.

See Appendix B for a more detailed discussion of False Course Head groups and a description of the members of each group.

9.1 The fchGroups element syntax

The set of FCH groups that apply to a particular method is shown as a string made up of their group letters concatenated together. For Major methods this string simply consists of the relevant letters placed in alphabetical order with the upper case letters appearing first and lower case following, for example BDce.

For Royal and above we have to distinguish between the in-course and out-of-course groups, so the `fchGroups` string lists the in-course groups first followed by the out-of-course ones, the two sets being separated by a / character. Thus E/C means in-course E and out-of-course C. This also applies to the lower-case groups, even though there is no ambiguity. So if a Royal method had in-course E and out-of-course c, it must be notated as E/c. The / character is always required for Royal and above, even if the method happens to have no in-course, or no out-of-course FCH groups.

We conclude this section with some examples

Major methods:

BDe	In-course and out-of-course BD, out-of-course e
BD/e	invalid – the / character is not used in Major methods

Royal and above:

BD/e	In-course BD, out-of-course e
BDe	invalid – omitting the / character is only allowed for Major methods
/e	out-of-course e only (no in-course groups at all)
B/	in-course B only
/	no FCH groups at all

10 References

[RFC2119] *Key words for use in RFCs to Indicate Requirement Levels*, S. Bradner, Author. Internet Engineering Task Force, June 1999. Available at http://www.ietf.org/rfc/rfc2119.txt.

[XML] *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, Editors. World Wide Web Consortium, 10 February 1998, revised 16 August 2006. This version of the XML 1.0 Recommendation is http://www.w3.org/TR/2006/REC-xml-20060816. The latest version of XML 1.0 is available at http://www.w3.org/TR/REC-xml.

[XML-Namespaces] *Namespaces in XML 1.0*, T. Bray, D. Hollander, A. Layman, and R. Tobin, Editors. World Wide Web Consortium, 14 January 1999, revised 16 August 2006. This version of the Namespaces in XML Recommendation is http://www.w3.org/TR/2006/REC-xml-names-20060816/. The latest version of Namespaces in XML is available at http://www.w3.org/TR/REC-xml-names.

[XPATH] XML Path Language (XPath) Version 1.0 , W3C Recommendation. eds. James Clark and Steven DeRose. 16 November 1999. The latest version of XML Path Language 1.0 is available at http://www.w3.org/TR/xpath.

[XML Schema Structures] *XML Schema Part 1: Structures Second Edition*, H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, Editors. World Wide Web Consortium, 2 May 2001, revised 28 October 2004. This version of the XML Schema Part 1 Recommendation is http://www.w3.org/TR/2004/REC-xmlschema-1-20041028. The latest version of XML Schema Part 1 is available at http://www.w3.org/TR/xmlschema-1.

[XML Schema Datatypes] *XML Schema Part 2: Datatypes Second Edition*, P. Byron and A. Malhotra, Editors. World Wide Web Consortium, 2 May 2001, revised 28 October 2004. This version of the XML Schema Part 2 Recommendation is http://www.w3.org/TR/2004/REC-xmlschema-2-20041028. The latest version of XML Schema Part 2 is available at http://www.w3.org/TR/xmlschema-2.

[Unicode] *The Unicode Standard, Version 3.2.* The Unicode Consortium. Defined by: The Unicode Standard, Version 3.0 (Reading, MA, Addison-Wesley, 2000. ISBN 0-201-61633-5) See http://www.unicode.org/unicode/standard/standard.html.

[FCHs] *Symbolic treatment of False Course Heads.* M Hodgson 1962.

[CC Decisions] *The Council's Decisions*. Central Council of Church Bell Ringers revised 30 May 2005. This is available at http://www.methods.org.uk/ccdecs.htm

Appendix A. Acknowledgments

The following individuals were members of the Methods Committee during the development of this specification:

Andrew Alldrick, Roger Bailey, Philip Earis, Peter Niblett, Philip Saddleton, Tony Smith, Stephanie Warboys, Robin Woolley.

The committee wishes to acknowledge comments and contributions received from Martin Bright, Richard Smith and Ben Willetts.

Appendix B. False Course Head Groups

FCH groups for methods with Plain Bob lead-heads

In his original paper, Maurice Hodgson was concerned only with palindromic Major methods that had Plain Bob lead-ends, and only with the 60 in-course false course heads that fixed 178. He observed that they split into 19 groups which he labelled A..U. He did not use the letter J or Q.

When we investigate Royal and higher stages we discover that groups L, P and U each split into two, so in this specification we use the following classification for the in-course FCH groups

A	B	C	D	E	F	G	H	I	K	L1
23456	24365	25634	32546 46253	32465 43265	32654 45236	56423 63542	53462 63425	54632 65324	53624 65432	26543
L2	M	N	O	P1	P2	R	S	T	U1	U2
36245 42563	23564 23645 25463 26435	34562 46325 54263 62345	36524 46532 52643 65243	54326 64352	56342 64523	35642 45623 56234 62534	34625 45362 52364 64235	24536 24653 25346 26354 36452 43526 53246 62453	34256 35426 42356 43652 52436 63254	35264 42635

Group A consists just of rounds which, as we observed in section 5, is a false course head for every method and so we don't bother to include it in the list of FCH groups. For Major methods L1 and L2 always appear together, as do P1, P2 and U1, U2, so when notating the FCH groups for a Major method we use the letter L to represent L1+L2, similarly for the letters P and U, thus matching Hodgson's original notation.

Roger Baldwin extended this analysis to examine the 60 out-of-course false course heads affecting only bells 2,3,4,5 and classified their groups as follows

B	C	D	E	F	H	K1	K2	N1	N2
23654 25436 32456 43256	34265 42365	24356 53426 63452	24635 25364	24563 26345	36542 46523 56243 62543	34526 46352 52346 64253	54362 64325	35462 43625 53264 62435	35624 45632 52634 65234
O	T	a1	a2	B	c	d	e	f	
53642 56432 63524 65423	32564 32645 45263 46235	23465	23546 26453	25643 26534	35246 36254 42536 42653	34652 45326 54236 62354	36425 43562 52463 63245	54623 56324 64532 65342	

As with the in-course FCH groups, three pairs of these, K1 and K2, N1 and N2, and a1 and a2 always appear together for Major methods, and so at that stage they are designated using the symbols K, N and a respectively.

You will notice that some of these groups (those with upper case letters) have the same names as the "in-course" groups that we looked at earlier. This is intentional, because if a palindromic Major method contains a false course head from one of these upper-case out-of-course groups, then it must also contain all the false course heads from the correspondingly named in-course group. This means that for Major methods we can use an upper case letter to represent both the in-course and out-of-course groups. So as an example we can use the letter B for Major methods to represent the set of false course heads {23456, 23654, 25436, 32456, 43256}. This does not apply to other stages however, and so for any other stage we need to be clear whether we mean “in-course B” or “out-of-course B”. The syntax used for the `fchGroups` element value separates the in-course groups from the out-of-course groups for stages Royal and above.

So far we have discussed 25 groups for Major methods, and between them these groups contain all the False Course Heads with 178 fixed. If we now consider all the false courses, including those where the 7 is not fixed, we find that the majority of them fall into the 25 groups we have just discussed, alongside the FCHs which have 178 fixed. One important point to note is that although groups a..f contain no in-course members with 178 fixed they do contain some in-course members with just 18 fixed.

We need only three further groups to cover all the remaining FCHs for a Major method with Plain Bob lead-heads. These are sometimes referred to as “tenors-parted” groups; in this specification these are given letter X, Y, Z, and they are shown here with all their members

<table>
<tr><td>X</td><td>Y</td></tr>
<tr><td rowspan="3">257643 374652 627534 723645
265743 437625 632754 724653
276354 457632 635742 736425
276435 475623 657423 746532
346752 526734 672453 762354
367245 546723 673245 763542
367524 564732 675324 764235</td><td>267534 364752 625743 724635
275643 367542 635724 726534
276453 376425 637425 734652
675423 764532</td></tr>
<tr><td>Z</td></tr>
<tr><td>457623 546732 672354 763245</td></tr>
</table>

The differences between Major and higher stages

The differences between FCH groups at Major and all higher stages can be summarised as

- The Major FCH groups include false course heads in which the 7 is affected. In contrast the groups at Royal and above are defined only to contain false heads that fix 1 and 7, 8 and above (for methods with Plain Bob lead-heads). A consequence of this is that there are only 25 named FCH groups for Royal and higher stages. Groups X, Y and Z are only defined for Major.
- Every possible Major FCH is a member of a named FCH group. This is not true for Royal or above, there are simply too many FCHs for this to be practical.
- The FCH groups that end with a numbered suffix, for example P1, P2, always appear together at the Major stage, and so are represented by a single letter with no suffix (e.g. P instead of P1 and P2). At other stages these groups are always shown with their suffixes, so if a given Royal method happens to have FCHs from both P1 and P2 then its falseness is denoted as P1P2.

The in-course and out-of-course groups that have the same letter always appear together at the Major stage and so the letter is only listed once when a Major method’s falseness is notated. In contrast for Royal and above, the notation has to distinguish between in-course and out-of-course groups as described in section 9.1.

FCH groups for Major methods with non-Plain Bob lead-heads

Single-hunt Major methods that are palindromic but which don’t have Plain Bob lead-heads nevertheless have an FCH group structure which is analogous (in fact isomorphic) to the FCH group structure for methods with Plain Bob lead-heads.

The way the FCHs group together depends on the lead-head of the method. Note that we are talking here about the full set of 28 groups with only 1 and 8 fixed, i.e. including groups X, Y and Z, and for some lead-ends we don’t find three exclusively tenors-parted groups.

Although the lead-head determines which FCHs occur together, we do have a choice when it comes to assigning an FCH group letter to each group. The convention we choose in this specification is to base the assignment on the assumption that a method is likely to be rung with bobs that affect three bells. The assignment is then chosen so that the the course entered as a result of calling a bob is a member of group U (just as it is for Plain Bob lead-heads).

The `fchGroups/@affected` attribute lists a set of three bells affected by a bob in the plain course. This is sufficient to indicate which FCH group should be designated as group U, and once that is done the assignment of the other groups can be done to match. The three bells given in the `fchGroups/@affected` can show the result of a bob at any lead in the plain course.

As an example, consider the method K522 Surprise Major. This is a seconds place method with a lead-head of 15478263. Its falseness expressed using full false course heads is

```
<falseness>
  <falseCourseHeads fixed="18">
    <inCourse>13254678 13625748 14275368 14327658 14357268 14625378 16324758
              12563478 12746538 13526478 14263578 14753628 17524638 17543628
    </inCourse>
    <outOfCourse>12463578 12536478 12546738 12743568 15432768 15743628 17542638
                 12475368 12635748 14375268 14627358 15347268 16325748 16342758
    </outOfCourse>
  </falseCourseHeads>
</falseness>?
```

This can be expressed using FCH groups as follows

```
<falseness>
  <fchGroups affected="234">FL</fchGroups>
</falseness>?
```

When a Method XML document is being read, the set of false course heads can be derived from the FCH groups listed using the following process:

1. Let **e** be the lead-head of the method when expressed in a rotation such that each lead has palindromic symmetry (in our example 15478263) and **b** a row permuting just the three bells included in the `fchGroups/@affected` attribute (in our example 14235678 or 13425678).
2. Find a row **k** that satisfies both of the following conditions:
   a) **k.e.k**$^{-1}$ is a Plain Bob lead-head
   b) **k.b.k**$^{-1}$ is a row that is member of the regular FCH group U

   In our example the row 12435678 will work for **k**.
3. Now examine the groups listed in the `fchGroups` element. Take the set of FCHs that would be members of the group if this were a method with a regular Plain Bob lead-head, and for each such FCH f, compute **k**$^{-1}$**.f.k**. These will be the true FCHs for the method. Note that you have to consider FCH group A (**f** = 13254768) in addition to the set of groups listed in `fchGroups`.

If you are only interested in FCHs that fix both 7 and 8, and if you can find a value of **k** that fixes both 7 and 8, then in step 3 it is only necessary to consider values of **f** that have 7 and 8 in their home position.

In our example the five in-course FCHs fixing 1,7,8 can be computed as follows:

12435678 . 13265478 . 12435678 = 14625378 (group F)

12435678 . 14523678 . 12435678 = 13254678 (group F)

12435678 . 12654378 . 12435678 = 12563478 (group L)

12435678 . 13624578 . 12435678 = 14263578 (group L)

12435678 . 14256378 . 12435678 = 13526478 (group L)

When constructing the `fchGroups` element in a Method XML document, the converse process can be followed. Steps 1 and 2 are the same as above, but instead of Step 3 you take the FCHs for the method, and for each such **f** compute **k.f.k**$^{-1}$ and then examine the table of regular FCH groups to determine which group it belongs to.

Note that the choice of bob suggested by the `fchGroups/@affected` does not constrain ringers to use that kind of bob, however if they do, then the FCH group letters will more naturally line up with the falseness of compositions.

FCH groups for methods with non-Plain Bob lead-heads at higher stages

A similar approach can be followed for the stages Royal and above, except that condition 2b becomes

2b) **k.b.k**$^{-1}$ is a row that is member of the regular FCH group U1

Note that in the case of Royal, it is not always possible to find a **k** that satisfies conditions 2a and 2b. To see an example, consider the lead-head 1850742639 with the choice of {2,3,4} as the three affected bells These three bells come together three times in the plain course (at the fourth lead-head 1423896075 and at the seventh lead-head 1342079568) . The group U1 contains permutations of {2,3,4}, {2,3,5} and {2,4,6}, and none of these come together more than once in the set of plain-bob lead-heads, so it is not possible to find a **k** that will work. The `fchGroups` element must only be used if such a **k** does in fact exist.

The other thing to be aware of at higher stages is that the false course head groups do not necessarily refer to courses in which 7, 8 and higher bells are all fixed. The set of courses is in fact the set { **k**$^{-1}$**.*f*.k** } where ***f*** runs over all the permutations of 2..6 while **k** stays constant with the value we chose to meet conditions 2a and 2b. There are two special kinds of **k** that do mean that 7, 8 and higher bells are fixed:

- A value of **k** that itself keeps 7, 8 and higher bells fixed, for example 1243567890
- A value of **k** that permutes 7, 8 and higher bells among themselves, for example 1243569078

It is easy to see that if **k** takes one of these forms then the set { **k**$^{-1}$**.*f*.k** } will itself only contain permutations that fix 7,8 and higher bells. So if we can find a value of **k** that meets conditions 2a and 2b, and has one of these forms, then it does make sense to think of the false course head groups as fixing 7, 8 and higher bells.

Appendix C. XML Schema

The XML types and elements used in this specification are defined in the following XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
	Central Council of Church Bell Ringers, Methods Committee

	XML Schema for method definitions and method collections
	Version 1.0 specification
	http://www.cccbr.org.uk/methods/schemas/2007/05/methods

	Copyright (c) 2005, 2007

	This schema contains three global element definitions that are intended for use as
document roots:

	1. The <method> global element. This contains data concerning a single method
	2. The <methodSet> global element. This is intended to contain a group of methods
which have some properties in common
	3. The <collection> global element which can be used in a document that contains a
whole collection of methods, possibly with
	quite different properties. The <collection> element contains 0 or more
<methodSet> elements

	In addition it contains the complex types used by these three elements:

	4. The complex type definition called methodType, which is the type used by
<method>.
	5. The complex type definition called methodSetType, which is the type used by
<methodSet>.
	6. The complex type definition called collectionType, which is the type used by
<collection>.

	It also defines two substitution groups:

	7. A substitution group for references, with an abstract head and some concrete
instantiations.
	8. A substitution group for performance references, with an abstract head and some
concrete instantiations.

	The methodSetType and methodType definitions are fairly open-ended - most of their
contents are optional. This is to allow them to be used in a variety of different
contexts.

	Users can adapt these definitions as follows:

	- Omit optional contents
	- Add references or performance references from their own namespaces to the method
element, using the substitution groups.
	- Add additional elements from their own namespaces as children of the notes and
meta elements.
	- Add attributes from their own namespaces to the collection, methodSet or method
elements.
-->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
	xmlns:mx="http://www.cccbr.org.uk/methods/schemas/2007/05/methods"
	targetNamespace="http://www.cccbr.org.uk/methods/schemas/2007/05/methods"
elementFormDefault="qualified">
	<annotation>
		<documentation>
		Central Council of Church Bell Ringers, Methods Committee.

		XML Schema for method definitions and method collections. Version 1.0
specification.
```

```

        This schema defines the format used by Central Council XML methods 
collections, and also provides a global
        element definition called method which can be used to pass XML-formatted 
method definition and metadata
        between ringing programs.

        </documentation>
    </annotation>

    <import namespace="http://www.w3.org/XML/1998/namespace" 
schemaLocation="http://www.w3.org/2001/xml.xsd" />

    <!--  The Global Element Definitions    -->

    <element name="method" type="mx:methodType" />
    <element name="methodSet" type="mx:methodSetType" />
    <element name="collection" type="mx:collectionType" />

    <!--  The types used by these Global Element Definitions    -->

    <complexType name="collectionType">
        <sequence>
            <element name="collectionName" type="token" minOccurs="0"/>
            <element ref="mx:notes" minOccurs="0">
                <annotation>
                    <documentation>
                        An human-language description of the collection, 
along with any relevant comments to assist in
                        its interpretation.
                    </documentation>
                </annotation>

            </element>
            <element ref="mx:methodSet" maxOccurs="unbounded" />
        </sequence>
        <attribute name="uuid" type="anyURI">
            <annotation>
                <documentation>An id that uniquely identifies this collection 
and its revision level.</documentation>
            </annotation>
        </attribute>
        <attribute name="date" type="date"/>
        <attribute ref="mx:decisionsYear" />
        <anyAttribute namespace="##other" processContents="lax" />

    </complexType>

    <!--  The type used by the methodSet Global Element Definition   -->
    <complexType name="methodSetType">
        <annotation>
            <documentation>
                This type contains a list of methods, and can contain a set of 
properties. These properties apply to all
                the methods in the list, unless the method itself specifies a 
conflicting property value. If this
                happens then the value in the method definition takes 
precedence.
            </documentation>
        </annotation>

        <sequence>
            <element ref="mx:notes" minOccurs="0"/>
            <element name="properties">
                <complexType>
                    <all>
                        <element ref="mx:classification" minOccurs="0" />
```

```
                                        <element ref="mx:stage" minOccurs="0" />
                                        <element ref="mx:lengthOfLead" minOccurs="0" />
                                        <element ref="mx:numberOfHunts" minOccurs="0" />
                                        <element ref="mx:huntbellPath" minOccurs="0" />
                                        <element ref="mx:leadHead" minOccurs="0" />
                                        <element ref="mx:leadHeadCode" minOccurs="0" />
                                        <element ref="mx:falseness" minOccurs="0" />
                                        <element ref="mx:symmetry" minOccurs="0" />
                                        <element ref="mx:extensionConstruction"
minOccurs="0" />
                                        <element ref="mx:notes" minOccurs="0" />
                                        <element ref="mx:meta" minOccurs="0" />
                                 </all>
                           </complexType>
                    </element>
                    <element ref="mx:method" minOccurs="0" maxOccurs="unbounded"/>
             </sequence>
             <attribute ref="mx:decisionsYear" />
             <anyAttribute namespace="##other" processContents="lax" />
       </complexType>


       <!--    Global definitions shared by more than one type    -->

       <element name="classification">
             <complexType>
                    <simpleContent>
                           <extension base="mx:classType">
                                 <attribute name="little" type="boolean"
default="false"></attribute>
                                 <attribute name="differential" type="boolean"
default="false"></attribute>
                                 <attribute name="plain" type="boolean"
default="false"></attribute>
                                 <attribute name="trebleDodging" type="boolean"
default="false"></attribute>
                           </extension>
                    </simpleContent>
             </complexType>
       </element>

       <simpleType name="classType">
             <restriction base="string">
                 <enumeration value="" />
                    <enumeration value="Place" />
                    <enumeration value="Bob" />
                    <enumeration value="Slow Course" />
                    <enumeration value="Treble Bob" />
                    <enumeration value="Delight" />
                    <enumeration value="Surprise" />
                    <enumeration value="Alliance" />
                    <enumeration value="Treble Place" />
                    <enumeration value="Hybrid" />
             </restriction>
       </simpleType>

       <element name="lengthOfLead" type="positiveInteger" />
       <element name="stage" type="positiveInteger" />
       <element name="numberOfHunts" type="nonNegativeInteger" />
       <element name="huntbellPath" type="mx:pathType">
             <annotation>
                    <documentation>
                           The path of the principal hunt bell, expressed using the
positions visited by the bell.
                    </documentation>
             </annotation>
       </element>

       <element name="leadHead" type="mx:rowType" />

```

```
       <element name="leadHeadCode" type="mx:leadHeadCodeType" />

       <element name="falseness">
              <complexType>
                     <sequence>
                            <element name="falseCourseHeads" minOccurs="0"
maxOccurs="unbounded">
                                   <complexType>
                                          <all>
                                                 <annotation>
                                                        <documentation>
                                                               In-course and out-of-course
false course heads are listed separately. Note that
                                                               these elements are optional,
however the absence of one or other of these element
                                                               does not imply that there are
no such false course heads, it merely indicates that
                                                               they are not recorded in the
falseness element.
                                                        </documentation>
                                                 </annotation>
                                                 <element name="inCourse" type="mx:fchType"
minOccurs="0" />
                                                 <element name="outOfCourse"
type="mx:fchType" minOccurs="0" />
                                          </all>
                                          <attribute name="fixed" type="mx:fixedType"
use="required"></attribute>
                                   </complexType>
                            </element>
                            <element name="fchGroups" type="mx:fchGroupType" minOccurs="0"
/>

                     </sequence>
              </complexType>
       </element>

       <element name="notes">
              <complexType mixed="true">
                     <sequence>
                            <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
                     </sequence>
                     <attribute ref="xml:lang"/>
              </complexType>
       </element>

       <element name="extensionConstruction">
              <simpleType>
                     <restriction base="string">
                            <pattern value="\d+([A-Z][A-Z])+/\d+([A-Z][A-Z])+"></pattern>
                     </restriction>
              </simpleType>
       </element>

       <element name="symmetry">
              <simpleType>
                     <list itemType="mx:symmetryType" />
              </simpleType>
       </element>

       <element name="meta">
              <complexType>
                     <sequence>
```

```
                <any namespace="##other" processContents="lax" 
maxOccurs="unbounded" />
            </sequence>
        </complexType>
    </element>

    <element name="references">
        <complexType>
            <sequence>
                <element ref="mx:ref" minOccurs="1" maxOccurs="unbounded" />
            </sequence>
        </complexType>
    </element>

    <attribute name="decisionsYear" type="gYear">
        <annotation>
            <documentation>
                Classification and other data reflects the Central Council 
decisions current at the end of the meeting
                held in the year given by this attribute
            </documentation>
        </annotation>
    </attribute>

    <!--       Method definition type                            -->
    <complexType name="methodType">

        <all>

            <!--     Naming and Classification                -->

            <element name="name" type="token" nillable="true" minOccurs="0">
                <annotation>
                    <documentation>
                        The name of the method. An empty name with the 
attribute xsi:nil="true" indicates that this
                        method has not yet been officially named. Note 
that a blank name with xsi:nil="false" is valid
                        in the case of Little Bob.
                    </documentation>
                </annotation>
            </element>

            <element ref="mx:classification" minOccurs="0" />

            <element name="title" nillable="true" minOccurs="0">
                <simpleType>
                    <restriction base="token">
                        <minLength value="1" />
                    </restriction>
                </simpleType>
            </element>

            <!--    Definitional elements                    -->

            <element ref="mx:stage" minOccurs="0" />

            <element name="notation" minOccurs="0" type="mx:notationType">
                <annotation>
                    <documentation>Place notation for a lead of the method. 
Use - rather than x or X.</documentation>
                </annotation>
            </element>

            <!--    Technical metadata                       -->

            <element ref="mx:lengthOfLead" minOccurs="0" />

            <element ref="mx:leadHead" minOccurs="0" />

            <element ref="mx:numberOfHunts" minOccurs="0" />

            <element ref="mx:huntbellPath" minOccurs="0" />

            <element ref="mx:leadHeadCode" minOccurs="0" />

            <element ref="mx:falseness" minOccurs="0" />

             <element ref="mx:symmetry" minOccurs="0" />

            <element ref="mx:extensionConstruction" minOccurs="0" />


            <!--      Other metadata                        -->

            <element ref="mx:notes" minOccurs="0" />

            <element ref="mx:meta" minOccurs="0" />

            <element ref="mx:references" minOccurs="0" />

            <element name="performances" minOccurs="0">
                <complexType>
                    <sequence>
                        <element ref="mx:performance" minOccurs="1"
maxOccurs="unbounded" />
                    </sequence>
                </complexType>
            </element>

        </all>
        <attribute name="id" type="ID" />
     <attribute ref="mx:decisionsYear" />
        <anyAttribute namespace="##other" processContents="lax" />
    </complexType>

    <!--          References defined by this schema             -->

    <element name="ref" type="mx:refType" abstract="true">
        <annotation>
            <documentation>
                This is the abstract definition of a reference to this method
in some external source. This schema
                defines a number of concrete references. Users can define
their own references in their own namespaces
                and adding them to the mx:ref substitution group
            </documentation>
        </annotation>
    </element>

    <element name="journalRef" substitutionGroup="mx:ref">
        <annotation>
            <documentation>References to this method in a
journal.</documentation>
        </annotation>
        <complexType>
            <simpleContent>
                <restriction base="mx:journalRefType">
                    <attribute name="journal" type="normalizedString"
use="required" />
                </restriction>
            </simpleContent>
        </complexType>
    </element>
```

```
       <element name="rwRef" substitutionGroup="mx:ref">
              <annotation>
                     <documentation>
                            A list of one more references to this method in &quot;The
Ringing World&quot;.
                     </documentation>
              </annotation>
              <complexType>
                     <simpleContent>
                            <restriction base="mx:journalRefType">
                                   <pattern value="V?\d+/\d+(\s(V?\d+/)?\d+)*"></pattern>
                                   <attribute name="journal" type="normalizedString"
fixed="The Ringing World" />
                            </restriction>
                     </simpleContent>
              </complexType>
       </element>

       <element name="bnRef" substitutionGroup="mx:ref">
              <annotation>
                     <documentation>
                            A list of one or more references to this method in &quot;The
Bell News&quot;.
                     </documentation>
              </annotation>
              <complexType>
                     <simpleContent>
                            <restriction base="mx:journalRefType">
                                   <pattern value="V?\d+/\d+(\s(V?\d+/)?\d+)*"></pattern>
                                   <attribute name="journal" type="normalizedString"
fixed="The Bell News" />
                            </restriction>
                     </simpleContent>
              </complexType>
       </element>

       <element name="cbRef" substitutionGroup="mx:ref">
              <annotation>
                     <documentation>
                            A list of one or more references to this method in
&quot;Church Bells&quot;.
                     </documentation>
              </annotation>
              <complexType>
                     <simpleContent>
                            <restriction base="mx:journalRefType">
                                   <pattern value="V?\d+/\d+(\s(V?\d+/)?\d+)*"></pattern>
                                   <attribute name="journal" type="normalizedString"
fixed="Church Bells" />
                            </restriction>
                     </simpleContent>
              </complexType>
       </element>

       <element name="tdmmRef" substitutionGroup="mx:ref">
              <annotation>
                     <documentation>Numerical index in the Treble Dodging Minor Methods
collection</documentation>
              </annotation>
              <complexType>
                     <simpleContent>
                            <restriction base="mx:refType">
                                   <pattern value="\d+"></pattern>
                            </restriction>
                     </simpleContent>
              </complexType>
       </element>
```

```

	<element name="pmmRef" substitutionGroup="mx:ref">
		<annotation>
			<documentation>Numerical index in the Plain Minor Methods 
collection</documentation>
		</annotation>
		<complexType>
			<simpleContent>
				<restriction base="mx:refType">
					<pattern value="\d+"></pattern>
				</restriction>
			</simpleContent>
		</complexType>
	</element>

	<!--                 Performance References defined by this schema             -->

	<element name="performance" type="mx:performanceType" abstract="true">
		<annotation>
			<documentation>
				This is the abstract definition of a performance reference. 
This schema defines a number of concrete
				references. Users can define their own references in their own 
namespaces and adding them to the mx:
				performance substitution group
			</documentation>
		</annotation>
	</element>

	<element name="firstTowerbellPeal" type="mx:performanceType" 
substitutionGroup="mx:performance">
		<annotation>
			<documentation>Details of the first single-method tower bell peal of 
this method</documentation>
		</annotation>
	</element>

	<element name="firstHandbellPeal" type="mx:performanceType" 
substitutionGroup="mx:performance">
		<annotation>
			<documentation>Details of the first single-method handbell peal of 
this method</documentation>
		</annotation>
	</element>

	<element name="firstInclusionInTowerbellPeal" type="mx:performanceType" 
substitutionGroup="mx:performance">
		<annotation>
			<documentation>Details of the first tower bell peal that includes 
this method</documentation>
		</annotation>
	</element>

	<element name="firstInclusionInHandbellPeal" type="mx:performanceType" 
substitutionGroup="mx:performance">
		<annotation>
			<documentation>Details of the first handbell peal that includes this 
method</documentation>
		</annotation>
	</element>


	<element name="firstTowerbellExtent" type="mx:performanceType" 
substitutionGroup="mx:performance">
		<annotation>
			<documentation>Details of the first extent of this method rung on 
tower bells</documentation>
		</annotation>
```

```
        </element>

        <element name="firstHandbellExtent" type="mx:performanceType"
substitutionGroup="mx:performance">
                <annotation>
                        <documentation>Details of the first extent of this method rung on
handbells</documentation>
                </annotation>
        </element>

        <!--                 Types used by these definitions              -->

        <complexType name="refType">
                <simpleContent>
                        <extension base="string">
                                <attribute name="id" type="ID" />
                                <anyAttribute namespace="##any" processContents="lax" />
                        </extension>
                </simpleContent>
        </complexType>

        <complexType name="journalRefType">
                <simpleContent>
                        <restriction base="mx:refType">
                                <attribute name="journal" type="normalizedString" />
                        </restriction>
                </simpleContent>
        </complexType>

        <complexType name="performanceType">
                <all>
                        <element name="date" type="date" minOccurs="0" />
                        <element name="location" type="mx:locationType" minOccurs="0" />
                        <element name="society" type="normalizedString" minOccurs="0" />
                        <element ref="mx:references" minOccurs="0" />
                </all>
                <attribute name="id" type="ID" />
                <anyAttribute namespace="##other" processContents="lax" />
        </complexType>

        <complexType name="locationType">
                <all>
                    <element name="room" type="normalizedString" minOccurs="0" />
                        <element name="building" type="normalizedString" minOccurs="0" />
                        <element name="address" type="normalizedString" minOccurs="0" />
                        <element name="town" type="normalizedString" minOccurs="0" />
                        <element name="county" type="normalizedString" minOccurs="0" />
                <element name="region" type="normalizedString" minOccurs="0" />
                        <element name="country" type="normalizedString" minOccurs="0" />
                </all>
                <attribute name="id" type="ID" />
                <anyAttribute namespace="##other" processContents="lax" />
        </complexType>

        <simpleType name="pathType">
                <list itemType="positiveInteger" />
        </simpleType>

        <simpleType name="fixedType">
                <restriction base="string">
                        <pattern value="\w+" />
                </restriction>
        </simpleType>

        <simpleType name="leadHeadCodeType">
                <restriction base="string">
                        <pattern value="[abefghlm]|[cdjkpqrs]\d*" />
                </restriction>
```

```
       </simpleType>

       <complexType name="fchGroupType">
              <simpleContent>
                     <extension base="mx:fchGroupString">
                           <attribute name="affected" type="mx:affectedType"
default="234"></attribute>
                     </extension>
              </simpleContent>
       </complexType>

       <simpleType name="fchType">
              <list itemType="mx:rowType" />
       </simpleType>

       <simpleType name="fchGroupString">
              <restriction base="string">
                     <pattern

       value="A?B?C?D?E?F?G?H?I?K?(L|((L1)?(L2)?))M?N?O?(P|((P1)?(P2)?))R?S?T?(U|((U1)?(U
2)?))(/(B?C?D?E?F?H?(K|((K1)?(K2)?))(N|((N1)?(N2)?))O?T?))?(a|((a1)?(a2)?))?b?c?d?e?f?X?Y
?Z?" />
              </restriction>
       </simpleType>

       <simpleType name="affectedType">
              <restriction base="string">
                 <pattern value="([A-HJ-NP-WYZa-hj-np-wyz0-9]|\{\d+\})+" />
               <length value="3" />
              </restriction>
       </simpleType>

       <simpleType name="symmetryType">
              <restriction base="string">
                     <enumeration value="palindromic" />
                     <enumeration value="rotational" />
                     <enumeration value="double" />
              </restriction>
       </simpleType>

       <!--                 Place Notation                    -->

       <complexType name="notationType">
              <simpleContent>
                     <extension base="mx:placeNotationType">
                           <attribute name="sym" type="boolean" default="true" />
                     </extension>
              </simpleContent>
       </complexType>


       <simpleType name="placeNotationType">
              <annotation>
                     <documentation>
                           A sequence of changes. Each change is represented using
conventional place notation, in which positions
                           1 to 9 are represented using their digit, 10 by 0, 11 by E, 12
by T, 13-16 by A-D, 17-19 by F-H, 20-24
                           by J-N, 25-28 by P-S, 29-31 by U-W, 32-33 by Y-Z (lower case
letters are also permitted and are
                           interpreted as equivalent to their upper case counterparts).
In addition positions may be represented as
                           {nnn} where nnn is a positive integer, so T is equivalent to
{12}. The . character is used to separate
                           the changes in the sequence. It may be omitted before or after
-
                     </documentation>
```

```
            </annotation>
            <restriction base="string">
                  <pattern
                        value="(\-|(\{\d+\}|[A-HJ-NP-WYZa-hj-np-wyz0-9])+)(\.?(\-
|(\{\d+\}|[A-HJ-NP-WYZa-hj-np-wyz0-9])+))*(,(\-|(\{\d+\}|[A-HJ-NP-WYZa-hj-np-wyz0-
9])+)(\.?(\-|(\{\d+\}|[A-HJ-NP-WYZa-hj-np-wyz0-9])+))*)?" />
            </restriction>
      </simpleType>

      <simpleType name="rowType">
            <restriction base="string">
                  <pattern value="([A-HJ-NP-WYZa-hj-np-wyz0-9]|\{\d+\})+" />
            </restriction>
      </simpleType>

</schema>
```

Appendix D. Revision History

Rev	Date	By Whom	What
0.1	2006-01-24	Peter Niblett	Initial version.
0.3		Peter Niblett	Major revision
0.4	2007-03-25	Peter Niblett	Added Versioning strategy and new section on FCHs
0.9	2007-05-19	Peter Niblett	Support for non-Plain Bob FCH groups at stages > 8 Added region element
0.91	2007-05-31	Peter Niblett	Added methodSet/notes. Assorted corrections
1.0	2008-03-04	Peter Niblett	Allow expression of false course heads that affect the hunt bells, and relax restrictions on fch groups for Royal and above. Minor formatting and other corrections.