
Subject: constraints for OperatingPeriod

Posted by [Andreas Tanner](#) on Tue, 25 Sep 2012 13:00:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

The operatingPeriod type currently is modelled quite flexibly to accommodate different use cases. However, the standard does not define which combinations of attributes can meaningfully be used together. A stricter definition would spare a a lot of discussions between different users of the standard. Here is a suggestion:

The OperatingPeriod element can be used for three different use cases.

- the calendar based operating period:
 - bitmask and startDate are mandatory,
 - endDate, operatingDay, specialService are not allowed. [endDate is not allowed since it would be redundant with startDate + bitmask length]

- standard week operating period:
 - operatingDay is mandatory (at least one),
 - specialService optional
 - startDate, endDate are optional and if used, both must be given
 - bitmask is not allowed

- abstract operating period
 - name or code are mandatory
 - bitmask, operatingDay, specialService are not allowed.

Always allowed, and optional if not declared otherwise, are name, additionalName, code, description, timetablePeriodRef, xml:lang, dayOffset

For railML 3.0, I would suggest to model the three cases as distinct types CalendarBasedOperatingPeriod, etc, all derived from base class OperatingPeriod.

Best regards

--Andreas Tanner.
