As we have not received a reply to the question, I would suggest to build upon the solution suggestion described in the forum posting from 2004 and extend it with the following:

@passable: Denotes if you can pass between the track the switch/crossing is placed on (the principal track; usually straight) and the track connecting to the switch/crossing (the diverging track).
As the movement possibilities are given for a fully functional switch ("true"), a simple crossing ("false" and "false") and a double switch crossing ("true" and "true) the passable does not need to be defined there.
For fully functional switches/crossings only use @passable for each of the IS:crossing connection of a crossing@type="simpleSwitchCrossing" (single slip switch) with either the value combination "true" and "false" or "false" and "true".

Furthermore I suggest to be able to set a switch/crossing inn a reduced state in railML.
The use case for this is, when you pad(lock) a switch/switch crossing, you can still run over it in the set direction.
It would be useful to know if the switch/crossing is (pad)locked and in which position it is locked.
For this scenario we suggest to use the combination of state@disabled=true and passable=true/false under the switch/crossing element.
As there is no passable defined in railML for passing over the switch/crossing on the track the switch is placed on (the principal/straight track). But as the switch (or switches in the crossing) can only have one position, this is implicit given through the @passable="true"/"false" of the switch connection (the diverging track) .