### seq (Sequence Time Node)

This element describes the Sequence time node and it can only be activated when the one before it finishes. Conceptually it can be though of as follows:

 

[Example: For example, suppose we have a simple animation with a blind entrance.

<p:[timing](timing.docx)>

 <p:[tnLst](tnLst.docx)>

 <p:[par](par.docx)>

 <p:[cTn](cTn.docx) id="1" dur="indefinite" restart="never" nodeType="tmRoot">

 <p:[childTnLst](childTnLst.docx)>

 <p:seq concurrent="1" nextAc="seek">

 ...

 </p:seq>

 </p:[childTnLst](childTnLst.docx)>

 </p:[cTn](cTn.docx)>

 </p:[par](par.docx)>

 </p:[tnLst](tnLst.docx)>

</p:[timing](timing.docx)>

End Example]

|  |
| --- |
| Parent Elements |
| [childTnLst](childTnLst.docx) (§); [subTnLst](subTnLst.docx) (§); [tnLst](tnLst.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [cTn](cTn.docx) (Common Time Node Properties) | § |
| [nextCondLst](nextCondLst.docx) (Next Conditions List) | § |
| [prevCondLst](prevCondLst.docx) (Previous Conditions List) | § |

|  |  |
| --- | --- |
| Attributes | Description |
| concurrent (Concurrent) | This attribute specifies if concurrency is enabled or disabled. By default this attribute has a value of "disabled". When the value is set to "enabled", the previous element is left enabled when advancing to the next element in a sequence instead of being ended. This is only relevant for advancing via the next condition element being triggered. The only other way to advance to the next element would be to have the current element end, which implies it is no longer concurrent.The possible values for this attribute are defined by the XML Schema boolean datatype. |
| nextAc (Next Action) | This attribute specifies what to do when going forward in sequence. By default this attribute has a value of "none". When this is set to seek it will seek the element to a natural end time (not necessarily the actual end time). The natural end position is defined as the latest non-infinite end time of the children. If a child loops forever, the end of its first loop is used as its "end time" for the purposes of this calculation.Some container elements may have infinite durations due to an infinite-duration child element. The engine needs to recurse down through all infinite duration containers to calculate their natural duration in case a child might have non-infinite duration within it that needs to be taken into account.The possible values for this attribute are defined by the [ST\_TLNextActionType](ST_TLNextActionType.docx) simple type (§). |
| prevAc (Previous Action) | This attribute specifies what to do when going backwards in a sequence. By default it is set to "none" and nothing special is done. When the value is "skipTimed", the sequence will continue to go backwards until it reaches a sequence element that was defined to begin only on the n[ext](ext.docx) condition element.The possible values for this attribute are defined by the [ST\_TLPreviousActionType](ST_TLPreviousActionType.docx) simple type (§). |

The following XML Schema fragment defines the contents of this element:

<complexType name="CT\_TLTimeNodeSequence">

 <sequence>

 <element name="[cTn](cTn.docx)" type="CT\_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>

 <element name="[prevCondLst](prevCondLst.docx)" type="CT\_TLTimeConditionList" minOccurs="0" maxOccurs="1"/>

 <element name="[nextCondLst](nextCondLst.docx)" type="CT\_TLTimeConditionList" minOccurs="0" maxOccurs="1"/>

 </sequence>

 <attribute name="concurrent" type="xsd:boolean" use="optional"/>

 <attribute name="prevAc" type="[ST\_TLPreviousActionType](ST_TLPreviousActionType.docx)" use="optional"/>

 <attribute name="nextAc" type="[ST\_TLNextActionType](ST_TLNextActionType.docx)" use="optional"/>

</complexType>