2020-03-31 TAPI Meeting notes

Date

31 Mar 2020

Attendees

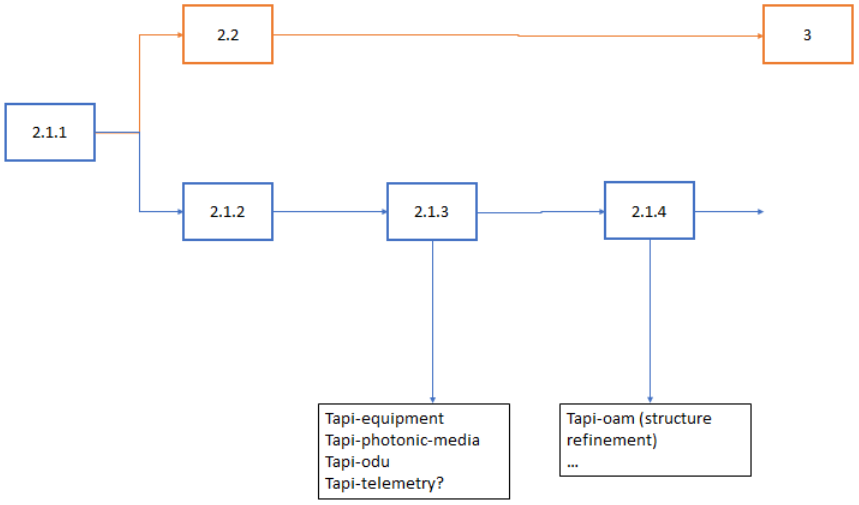
- Karthik Sethuraman
- Hing-Kam Lam
- Jonathan Sadler
- Andrea Mazzini
- Malcolm Betts
- Nigel Davis
- Italo Busi
- Pedro Amaral
- Arturo Mayoral

Goals

- **2.1.3 version freeze**
  - uml2yang tool clarifications
- ODU Attributes (payload type, tributary slot list, client type)
- Continue preparation of May Virtual Meeting agenda

Discussion items

5 mins	Administrative	Andrea Mazzini	• **Next F2F TAPI meeting, Virtual Meeting:**
			  • 04 May 2020 - 08 May 2020
			• Call slot assignment: last week we were preempted by another ONF meeting overlapping last scheduled hour
			  • ONF TAPI Call is scheduled by ONF admin from 3pm to 5pm CET
			  • easonably we can ask to extend to four hours - with option for two additional hours "on demand".
			  • Issue is temporary solved by ODTN project moving its call to Monday
			• **Poor quality of Zoom conference, unable to continue in the third hour**
			• 07 Apr 2020 TAPI Call: **2 hours**, note that both Europe and US are in daylight saving time, call will start at 3pm CET
			  • **2.1.3 version - status update**
			  • Continue preparation of May Virtual Meeting agenda

10 mins	Clarification on TAPI Versions	Malcolm Betts Hing-Kam Lam Karthik Sethuraman	Considering Arturo Mayoral diagram below, it was clarified that the *develop* branch (orange color, now called *Next Major Release*) is based on delivered version 2.2 (which according to our knowledge is not implemented), and that could lead to: • version 2.3 in case the added features do not break backward compatibility • version 3.0 in case the added features break backward compatibility. An example of "non backward compatible" feature is the alignment to IETF Topology (RFC 8345) 

30 mins	uml2yang tool	Nigel Davis Karthik Sethuraman	Nigel Davis presents a slide highlighting the differences found in Yang modules derived from same UML, • one module just generated without any additional editing • one module already present in 2.1.2 delivery Karthik Sethuraman explains that the differences are due to his handcrafting, specifically to replace "leaf-list" with "list" and "leafref with "container", according to a previous agreement which is still not coded in the tool. ☐ Nigel Davisand Andrea Mazzini to provide the detailed information regarding the necessary post-generation handcrafting of Yang modules.
10 mins	2.1.3 version freeze	Andrea Mazzini Nigel Davis	Andrea Mazzini presents otcc2020.AM.002_TAPI_2.1.3_Enhancements.pptx Reviewed all modifications and commits. Nigel Davis : YANG Streaming is fixed.

70 mins	ODU Attributes (payload type, tributary slot list, client type)	Andrea Mazzini
		
		Italo Busi
		
		Nigel Davis
		
		Malcolm Betts
		
		Hing-Kam Lam
			Andrea Mazzini, the G.875 v4.07 defines the following packages used by ODU TTP and CTP classes:
			
			• ODUkTp_Pac includes the
			  • Pointer to SNC
			• ODUCnTp_Pac includes the
			  • n of ODUCn
			
			• ODUkCtp_Pac includes the
			  • tributaryPortNumber
			  • tributarySlotList (with comment mentioning also ODUCn case)
			  • apsEnable and apsLevel
			• ODUCnCtp_Pac: empty
			
			• ODUkTtp_Pac: empty
			• ODUCnTtp_Pac includes the
			  • effectiveTimeSlotList (the list of effective time slots which are available for carrying ODUk clients)
			
			Italo Busi and Hing-Kam Lam provide some clarifications on G.709 and G.875 definitions, which lead to the following agreements:
			
			• Expected/Sent Payload Type values can be derived from OduType, OduMappingType and OduClientType. So only *acceptedPayloadType* remains in ODU TTP CEP.
			• Isolate ODU CSEP TTP and CTP related parameters in specific – mutually exclusive - pacs.
			• In ODU CSEP TTP pac replace *configuredPayloadType* with *configuredClientType.*
			
			Hing-Kam Lamasks about network provisioning scenario, i.e. considering both the ODU Connectivity Service end points - CSEPs.
			
			• Clarified that in case of asymmetric ODU Connectivity Service, where e.g. the ODU2 CEP is not visible at E-NNI side, because encapsulated in ODU4 HO Trail, there could be two CSEP instances to specify connectivity constraints, one referring to existing resource ODU4 CEP, the other representing the ODU2 position in server ODU4.
			• Another scenario is the constraining of position, e.g. an ODU2 Trail Connectivity Service with ODU2 TTP CSEP plus ODU2 CTP CSEP with tributaryPortNumber and tributarySlotList.