الاتحـــــــاد الدولـــــــي للاتصـــــــالات

ITU-T

قطـاع تقييس الاتصـالات
في الاتحاد الدولي للاتصالات

Z.145
(2006/03)

السلسلة Z: اللغات والجوانب العامة للبرمجيات في أنظمة الاتصالات

تقنيات الوصف الشكلي (FDT) – الاختبار وترميز ضبط الاختبار (TTCN)

الإصدار 3 من الاختبار وترميز ضبط الاختبار (TTCN-3): السطح البيني للتحكم (TCI)

التوصيـــة ITU-T Z.145

الاتحاد الدولي للاتصالات ITU

توصيات السلسلة Z الصادرة عن قطاع تقييس الاتصالات

اللغات والجوانب العامة للبرمجيات في أنظمة الاتصالات

تقنـــيات الوصـــف الشـــكلي (FDT)	
لغة المواصفة والوصف (SDL)	Z.109–Z.100
تطبيق تقنيات الوصف الشكلي	Z.119–Z.110
مخطط تعاقب الرسائل (MSC)	Z.129–Z.120
لغة تعريف الغرض الموسّعة (eODL)	Z.139–Z.130
الاختبار وترميز ضبط الاختبار (TTCN)	**Z.149–Z.140**
ترميز متطلبات المستعملين (URN)	Z.159–Z.150
لغـــات البرمجـــة	
CHILL: لغة المستوى الرفيع لدى قطاع تقييس الاتصالات	Z.209–Z.200
لغـــة الإنســـان–الآلـــة	
مبادئ عامة	Z.309–Z.300
قواعد النَظْم الأساسية وإجراءات التحاور	Z.319–Z.310
لغة الإنسان–الآلة (MML) الموسّعة من أجل مطاريف العرض المرئي	Z.329–Z.320
مواصفة السطح البيني الإنسان–الآلة	Z.349–Z.330
السطوح البينية الإنسان–الآلة الموجهة للمعطيات	Z.359–Z.350
السطوح البينية الإنسان–الآلة من أجل إدارة شبكات الاتصالات	Z.379–Z.360
الجـــودة	
جودة برمجيات الاتصالات	Z.409–Z.400
مظاهر الجودة للتوصيات المرتبطة بالبروتوكولات	Z.459–Z.450
الطـــرائق	
طرائق للتثبت من الصلاحية وللاختبار	Z.519–Z.500
البرمجيـــات الوســـيطة	
بيئة المعالجة الموزعة	Z.609–Z.600

لمزيد من التفاصيل، يرجى الرجوع إلى قائمة التوصيات الصادرة عن قطاع تقييس الاتصالات.

التوصيـــة ITU-T Z.145

الاختبار وترميز التحكم في اختبار الصياغة 3 (TTCN-3): السطح البيني للتحكم (TCI)

الملخص

تصف هذه التوصية السطوح البينية للتحكم TTCN-3 (الاختبار وترميز التحكم في الاختبار3) لتنفيذ نظام اختبار. ويوفر السطح البيني للتحكم TTCN-3 التكييف الموصي به لإدارة ومناولة عنصر اختبار وتشفير/فك تشفير نظام اختبار لمنصة اختبار معينة. وتعرف هذه التوصية السطوح البينية كمجموعة من عمليات مستقلة للغة مستهدفة.

تعرف السطوح البينية على أنها متلائمة مع التوصية ITU-T Z.140. ويستخدم تعريف السطح البيني في هذه التوصية لغة تعريف سطح بيني CORBA (IDL) ليحدد TCI تماماً. ويحدد القسمان 8 و9 تقابلات لغة لمواصفة مجردة للغات مستهدفة Java وANSI-C. ويرد في الملحق A موجزاً لمواصفة سطح بيني قائمة على IDL.

المصدر

وافقت لجنة الدراسات 17 (2005-2008) التابعة لقطاع تقييس الاتصالات بتاريخ 16 مارس 2006 على التوصية TU-T Z 145. بموجب الإجراء الوارد في التوصية ITU-T A.8.

تمهيـــد

الاتحاد الدولي للاتصالات وكالة متخصصة للأمم المتحدة في ميدان الاتصالات وتكنولوجيات المعلومات والاتصالات (ICT). وقطاع تقييس الاتصالات (ITU-T) هو هيئة دائمة في الاتحاد الدولي للاتصالات. وهو مسؤول عن دراسة المسائل التقنية والمسائل المتعلقة بالتشغيل والتعريفة، وإصدار التوصيات بشأنها بغرض تقييس الاتصالات على الصعيد العالمي.

وتحدد الجمعية العالمية لتقييس الاتصالات (WTSA) التي تجتمع مرة كل أربع سنوات المواضيع التي يجب أن تدرسها لجان الدراسات التابعة لقطاع تقييس الاتصالات وأن تُصدر توصيات بشأنها.

وتتم الموافقة على هذه التوصيات وفقاً للإجراء الموضح في القرار رقم 1 الصادر عن الجمعية العالمية لتقييس الاتصالات.

وفي بعض مجالات تكنولوجيا المعلومات التي تقع ضمن اختصاص قطاع تقييس الاتصالات، تعد المعايير اللازمة على أساس التعاون مع المنظمة الدولية للتوحيد القياسي (ISO) واللجنة الكهرتقنية الدولية (IEC).

ملاحظـــة

تستخدم كلمة "الإدارة" في هذه التوصية لتدل بصورة موجزة سواء على إدارة اتصالات أو على وكالة تشغيل معترف بها.

والتقيد بهذه التوصية اختياري. غير أنها قد تضم بعض الأحكام الإلزامية (بهدف تأمين قابلية التشغيل البيني والتطبيق مثلاً). ويعتبر التقيّد بهذه التوصية حاصلاً عندما يتم التقيّد بجميع هذه الأحكام الإلزامية. ويستخدم فعل "يجب" وصيغ ملزمة أخرى مثل فعل "ينبغي" وصيغها النافية للتعبير عن متطلبات معينة، ولا يعني استعمال هذه الصيغ أن التقيّد بهذه التوصية إلزامي.

حقوق الملكية الفكرية

يسترعي الاتحاد الانتباه إلى أن تطبيق هذه التوصية أو تنفيذها قد يستلزم استعمال حق من حقوق الملكية الفكرية. ولا يتخذ الاتحاد أي موقف من القرائن المتعلقة بحقوق الملكية الفكرية أو صلاحيتها أو نطاق تطبيقها سواء طالب بها عضو من أعضاء الاتحاد أو طرف آخر لا تشمله عملية إعداد التوصيات.

وعند الموافقة على هذه التوصية، لم يكن الاتحاد قد تلقى إخطاراً بملكية فكرية تحميها براءات الاختراع يمكن المطالبة بها لتنفيذ هذه التوصية. ومع ذلك، ونظراً إلى أن هذه المعلومات قد لا تكون هي الأحدث، يوصى المسؤولون عن تنفيذ هذه التوصية بالاطلاع على قاعدة المعطيات الخاصة ببراءات الاختراع في مكتب تقييس الاتصالات (TSB) في الموقع http://www.itu.int/ITU-T/ipr/.

© ITU 2010

جميع الحقوق محفوظة. لا يجوز استنساخ أي جزء من هذه المنشورة بأي وسيلة كانت إلا بإذن خطي مسبق من الاتحاد الدولي للاتصالات.

المحتـــويات

الصفحة

1 مجال التطبيق 1
2 المراجع 1
3 التعاريف والمختصرات 2
1.3 التعاريف 2
2.3 المختصرات 3
4 مقدمة 3
5 الامتثال 4
6 الهيكل العام لنظام اختبار TTCN-3 4
1.6 كيانات في نظام اختبار TTCN-3 4
2.6 متطلبات تنفيذ نظام اختبار TTCN-3 7
7 السطح البيني للتحكم في TTCN-3 والعمليات 7
1.7 نظرة شاملة على TCI 7
2.7 معطيات TCI 9
3.7 عمليات TCI 18
8 تقابل لغة Java 83
1.8 مقدمة 83
2.8 الأسماء والمنظورات 83
3.8 الثوابت 97
4.8 تقابل سطوح بينية 98
5.8 معلمات خيارية 105
6.8 تدميث TCI 105
7.8 مناولة خطأ 105
9 تقابل لغة ANSI-C 105
1.9 مقدمة 105
2.9 السطوح البينية مع قيمة 106
3.9 سطح بيني مع تسجيل 109
4.9 سطوح بينية مع عملية 109
5.9 معطيات 123
6.9 Miscellaneous 125
10 تقابل W3C XML 125
1.10 مقدمة 125
2.10 منظورات 125
3.10 تقابل نمط 125
4.10 تقابل عمليات على السطح البيني للتسجيل 145

الصفحة

11 سيناريوهات الاستخدام 166
1.11 التدميث وجمع المعلومات والتسجيل 167
2.11 تنفيذ اختبارات مجردة وتحكم 169
3.11 مناولة مكون 172
4.11 إنهاء اختبارات مجردة وتحكم 179
5.11 الاتصالات 184
الملحق A – مواصفة IDL لـــ TCI 189
الملحق B – تقابل XML لـــ TCI TL موفر 204
1.B تخطيط TCI-TL XML لأنماط بسيطة 204
2.B تخطيط TCI-TL XML للأنماط 204
3.B تخطيط TCI-TL XML لقيم 206
4.B تخطيط TCI-TL XML لمقاسات 211
5.B تخطيط TCI-TL XML لأحداث 218
6.B تخطيط TCI-TL XML لسجل 236
بيبليوغرافيا 238

الاختبار وترميز التحكم في اختبار الصياغة 3 (TTCN-3): السطح البيني للتحكم (TCI)

1 مجال التطبيق

تصف هذه التوصية السطوح البينية للتحكم TTCN-3 (الاختبار وترميز التحكم في الاختبار3) لتنفيذ نظام اختبار. ويوفر السطح البيني للتحكم TTCN-3 التكييف الموصي به لإدارة ومناولة عنصر اختبار وتشفير/فك تشفير نظام اختبار لمنصة اختبار معينة. وتعرف هذه التوصية السطوح البينية كمجموعة من عمليات مستقلة للغة مستهدفة.

تعرف السطوح البينية على أنها متلائمة مع التوصية ITU-T Z.140. ويستخدم تعريف السطح البيني في هذه التوصية لغة تعريف سطح بيني CORBA (IDL) ليحدد TCI تماماً. ويحدد القسمان 8 و9 تقابلات لغة لمواصفة مجردة للغات مستهدفة Java وANSI-C. ويرد في الملحق A موجزاً لمواصفة سطح بيني قائمة على IDL.

2 المراجع

تتضمن التوصيات التالية لقطاع تقييس الاتصالات وغيرها من المراجع أحكاماً تشكل من خلال الإشارة إليها في هذا النص جزءاً لا يتجزأ من هذه التوصية. وقد كانت جميع الطبعات المذكورة سارية الصلاحية في وقت النشر. ولما كانت جميع التوصيات والمراجع الأخرى تخضع إلى المراجعة، يرجى من جميع المستعملين لهذه التوصية السعي إلى تطبيق أحدث طبعة من التوصيات والمراجع الواردة أدناه. وتُنشر بانتظام قائمة توصيات قطاع تقييس الاتصالات السارية الصلاحية. والإشارة إلى وثيقة في هذه التوصية لا يضفي على الوثيقة في حد ذاتها صفة التوصية.

[1] ITU-T Recommendation Z.144 (2006), *Testing and Test Control Notation version 3 (TTCN-3): Runtime interface (TRI)*.

[2] ITU-T Recommendation Z.140 (2006), *Testing and Test Control Notation version 3 (TTCN-3): Core language*.

[3] ITU-T Recommendation Z.143 (2006), *Testing and Test Control Notation version 3 (TTCN-3) Operational semantics*.

[4] ITU-T Recommendation X.290 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts*.

ISO/IEC 9646-1:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts*.

[5] W3C Recommendation (2004), *XML Schema Part 0: Primer*.
NOTE – See at http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/.

[6] W3C Recommendation (2004), *XML Schema Part 1: Structures*.
NOTE – See at http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/.

[7] W3C Recommendation (2004), *XML Schema Part 2: Datatypes*.
NOTE – See at http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/.

3 التعاريف والمختصرات

1.3 التعاريف

لأغراض هذه التوصية، تنطبق المصطلحات والتعاريف التالية الواردة في التوصية ITU-T X.290 [4]:

1.1.3 متوالية اختبار مجردة: متوالية اختبار تتألف من حالات اختبارات مجردة.

2.1.3 كودك: كيان مشفر–مفكك مستخدم لتشفير وتفكيك معطيات لترسل وتستقبل على التوالي.

3.1.3 تشفير–تفكيك تشفير: كيان يدير مناولة القيمة والنمط بما في ذلك تشفير وتفكيك في نظام اختبار TTCN-3.

4.1.3 مناولة مكون: كيان يدير مناولة مكونات اختبار في نظام اختبار TTCN-3.

5.1.3 منفذ اتصالات: آلية مجردة لتيسير الاتصالات بين مكونات اختبار.

ملاحظة – يوضع نموذج منفذ اتصالات باعتباره صف انتظار FIFO في اتجاه الاستقبال. ويمكن للمنافذ أن تكون قائمة على رسالة أو على إجراء أو خليط من الاثنين.

6.1.3 مكون التحكم: مكون ينفذ سلوك جزء التحكم لوحدة TTCN-3.

7.1.3 متوالية اختبارات قابلة للتنفيذ: ارجع إلى ITU-T X.290 [4].

8.1.3 تنفيذ معلومات eXtra لاختبار: ارجع إلى ITU-T X.290 [4].

9.1.3 مُكيف منصة: كيان يكيف TTCN-3 قابل للتنفيذ لمنصة تنفيذ معينة.

ملاحظة – يخلق مكيف المنصة فكرة وحيدة لوقت نظام اختبار TTCN-3، وينفذ، صراحة وضمنيا، كل من المؤقتات ووظائف خارجية.

10.1.3 سطح بيني لنظام اختبار حقيقي: ارجع إلى ITU-T X.290 [4].

11.1.3 نظام تحت الاختبار: ارجع إلى ITU-T X.290 [4].

12.1.3 مكيف نظام SUT: كيان يكيف عمليات اتصالات TTCN-3 مع SUT قائم على أساس سطح بيني مجرد لنظام اختبار. وينفذ السطح البيني لنظام اختبار حقيقي.

13.1.3 اختبار وترميز التحكم في اختبار: ارجع إلى ITU-T X.290 [4].

14.1.3 حالة اختبار: ITU-T X.290 [4].

15.1.3 اختبار مجرد: سواء يرسل أو يستقبل معطيات اختبار (رسالة أو نداء إجراء) على منفذ اتصالات هو جزء من السطح البيني لنظام اختبار وكذلك أحداث إمهال لمؤقتات.

16.1.3 إدارة اختبار: كيان يوفر سطح بيني لمستعمل وكذلك إدارة نظام اختبار TTCN-3.

17.1.3 تسجيل اختبار: كيان يوفر معلومات تسجيل حول تنفيذ اختبار (بما في ذلك أيضاً معلومات وفرها بيان سجل TTCN-3).

18.1.3 إدارة اختبار والتحكم فيه: مجموعة من الكيانات توفر إدارة اختبار والتحكم فيه؛ وتتألف من إدارة اختبار ومناولة مكون وتسجيل اختبار وتشفير/تفكيك تشفير.

ملاحظة – إن TMC هو تنفيذ TCI.

19.1.13 نظام اختبار: ارجع إلى ITU-T Rec. X.290 [4].

20.1.3 سطح بيني لنظام اختبار: مكون اختبار يوفر تقابل المنافذ المتاحة في نظام اختبار TTCN-3 (مجرد) إلى التي يوفرها نظام اختبار حقيقي.

21.1.3 TTCN-3 قابل للتنفيذ: جزء من نظام اختبار يتناول تفسير أو تنفيذ TTCN-3 ETS.

22.1.3 سطوح بينية للتحكم في TTCN-3: ثلاثة سطوح بينية تعرف تفاعل TTCN-3 قابل للتنفيذ مع إدارة الاختبار وتشفير وفك تشفير ومناولة مكون الاختبار في نظام اختبار.

23.1.3 سطح بيني لتنفيذ TTCN-3: سطح بيني يعرف التفاعل لـــ TTCN-3 القابل للتنفيذ مع SUT ومكيفات منصة في نظام اختبار.

2.3 المختصرات

لأغراض هذه التوصية، تنطبق المختصرات التالية:

ATS متوالية اختبار مجردة (*Abstract Test Suite*)

CD تشفير/تفكيك تشفير (*Coding/Decoding*)

CH مناولة مكون (*Component Handling*)

ETS متوالية اختبار قابل للتنفيذ (*Executable Test Suite*)

IDL لغة تعريف سطح بيني (*Interface Definition Language*)

IXIT تنفيذ معلومات eXtra لاختبار (*Implementation eXtra Information for Testing*)

MSC خريطة تتابع الرسائل (*Message Sequence Chart*)

MTS مكون اختبار رئيسي (*Main Test Component*)

OMG زمرة إدارة شيء (*Object Management Group*)

PA مكيف منصة (*Platform Adaptor*)

PTC مكون اختبار متوازي (*Parallel Test Component*)

SA مكيف نظام SUT (*SUT Adaptor*)

SUT نظام تحت الاختبار (*System Under Test*)

TC تحكم في اختبار (*Test Control*)

TCI سطوح بينية للتحكم في TTCN-3 (*TTCN-3 Control Interfaces*)

TE TTCN-3 قابل للتنفيذ (*TTCN-3 Executable*)

TL تسجيل اختبار (*Test Logging*)

TM إدارة اختبار (*Test Management*)

TMC إدارة اختبار والتحكم فيه (*Test Management and Control*)

TRI سطح بيني لتنفيذ TTCN-3 (*TTCN-3 Runtime Interface*)

TSI سطح بيني لنظام اختبار (*Test System Interface*)

TTCN-3 اختبار وترميز تحكم في اختبار الصيغة 3 (*Testing and Test Control Notation Version 3*)

4 مقدمة

تتألف هذه التوصية من جزأين متميزين، يصف الجزء الأول هيكل تنفيذ نظام اختبار TTCN-3 ويقدم الجزء الثاني مواصفة السطوح البينية للتحكم في TTCN-3.

يعرض الجزء الأول تجزئة نظام اختبار TTCN-3 في أربعة كيانات رئيسية:

- إدارة اختبار والتحكم فيه (TMC)؛
- TTCN-3 قابل للتنفيذ (TE)؛
- مُكيف نظام SUT (SA)؛
- مُكيف منصة (PA).

تتألف TMC من ثلاثة كيانات: TM وCD وCH. وبالإضافة إلى ذلك، يعرف التفاعل بين هذه الكيانات، أي، السطوح البينية المتوافقة.

يصف الجزء الثاني من هذه التوصية السطوح البينية للتحكم في TTCN-3. وتعرف السطوح البينية على أساس العمليات المنفذة كجزء من كيان واحد وتطلبها كيانات نظام اختبار أخرى. ولكل عملية، تعرف مواصفة السطح البيني هياكل معطيات متصاحبة والأثر المقصود على نظام الاختبار وأي عوائق على استخدام العملية. ولاحظ أن مواصفات الأسطح البينية هذه تعرف فقط التفاعلات بين TE وTM وTE وCD وTE وCH. وبالنسبة للتفاعلات بين TE وSA وTE وPA، رجاء الرجوع إلى مواصفة السطح البيني لتنفيذ TTCN-3 ([1] ITU-T Z.144).

5 الامتثال

إن الحد الأدنى المطلوب لامتثال TCI لنظام اختبار TTCN-3 هو الالتزام بمواصفة السطح البيني الواردة في هذه التوصية. وينبغي أن يلتزم علم دلالات TTCN-3 في نظام الاختبار بعلم الدلالات التشغيلي المعرف في [3] ITU-T Z.143. وبالإضافة إلى ذلك، ينبغي دعم تقابل لغة واحدة. فمثلاً، إذا دعم بائع Java، ينبغي على نداءات عملية TCI وتنفيذها، التي هي جزء من TTCN-3 قابل للتنفيذ، الامتثال لـــ IDL مع تقابل Java المحدد في هذه التوصية. وبالنسبة للسطح البيني للتسجيل، يمكن استخدام تقابل XML بدلاً من Java أو تقابل C.

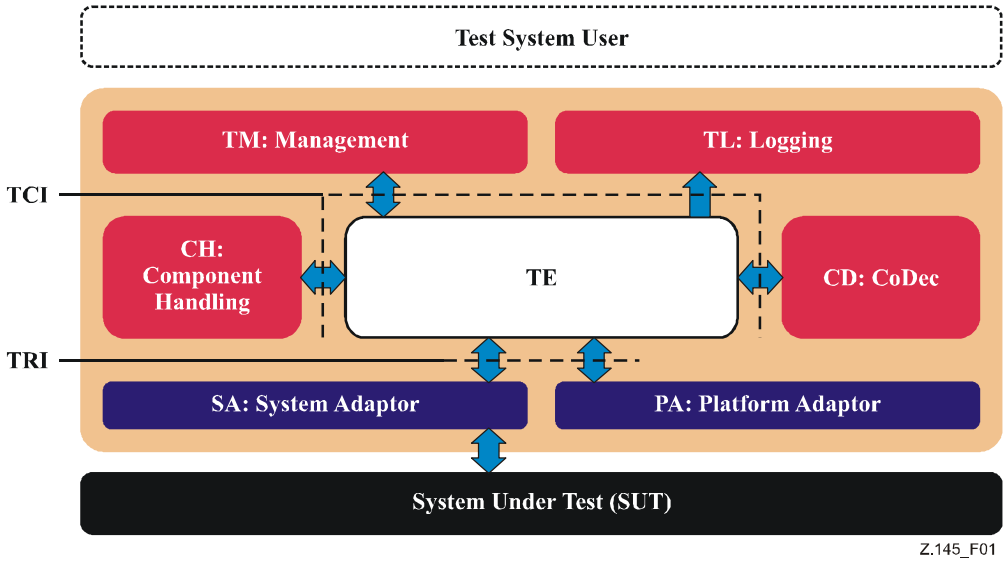
6 الهيكل العام لنظام اختبار TTCN-3

يمكن النظر إلى نظام اختبار TTCN-3 مفهومياً على أنه مجموعة من الكيانات المتفاعلة. وينفذ كل كيان وظيفة محددة لنظام الاختبار. وهذه الكيانات هي:

- إدارة تنفيذ اختبار؛
- تفسير أو تنفيذ شفرة TTCN-3 مجمعة؛
- تحقيق اتصالات صحيحة مع نظام SUT؛
- إدارة أنماط وقيم ومكونات اختبار؛
- تنفيذ وظائف خارجية؛
- مناولة عمليات مؤقت.

1.6 كيانات في نظام اختبار TTCN-3

يوضح الشكل 1 هيكل تنفيذ نظام اختبار TTCN-3.



الشكل Z.145/1 – الهيكل العام لنظام اختبار TTCN-3

كما يوضح الشكل 1، يفسر TTCN-3 قابل للتنفيذ، يشار إليه أيضاً بمتوالية اختبارات قابلة للتنفيذ، ويفسر وينفذ وحدات TTCN-3. ويمكن تعريف المكونات الهيكلية TE المختلفة: التحكم والسلوك والمكونات والأنماط والقيم وصفوف الانتظار. وتمثل المكونات الهيكلية في TE الوظيفية المعرفة في وحدة TTCN-3 أو بواسطة معيار TTCN-3 ([2] ITU-T Z.140) نفسه. فمثلاً، يمثل المكون الهيكلي "Control" جزء التحكم في وحدة TTCN-3، بينما يمثل المكون الهيكلي "Queues" المتطلب على TTCN-3 قابل للتنفيذ الذي يحتفظ كل منفذ لمكون اختبار بصف انتظار منفذه. وبينما الأول محدد في وحدة TTCN-3، يتطلب الأخير مواصفة TTCN-3.

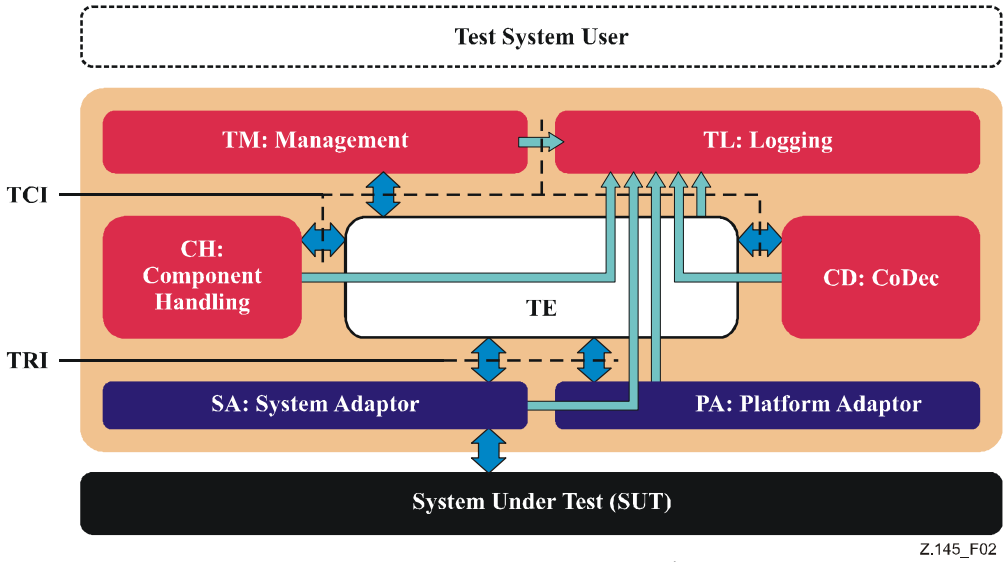
إن تنقيح TE، كما يبين الشكل 1، يوفر المساعدة في تعريف السطوح البينية للتحكم في TTCN-3. ويتوافق TE نمطياً في تنفيذ نظام اختبار إما مع شفرة قابلة للتنفيذ أنتجها مصرف TTCN-3 أو بواسطة مترجم TTCN-3.

يمكن تنفيذ TE بطريقة مركزية أو موزعة، أي، على جهاز اختبار وحيد أو عبر أجهزة اختبار متعددة، على التوالي. وبالرغم من أن الكيانات الهيكلية لـــ TE تنفذ وحدة TTCN-3 كاملة، يمكن توزيع كيانات هيكلية وحيدة عبر أجهزة اختبار عديدة.

ينفذ TE وحدة TTCN-3 على مستوى مجرد. وتجعل الكيانات الأخرى لنظام اختبار TTCN-3 المفاهيم المجردة هذه محسوسة. فمثلاً، لا يمكن تنفيذ مفهوم مجرد لإرسال رسالة أو استقبال إمهال في TE. وينفذ الجزء المتبقي من نظام الاختبار تشفير الرسالة وإرسالها عبر وسائل مادية محسوسة أو قياس الوقت وتحديد متى ينتهي المؤقت، على التوالي.

يعرف SA وPA وتفاعلهما مع TE في ITU-T Z.140 [1]. وتعرف مواصفة TCI التفاعل بين TE وTMC.

يوفر السطح البيني للتسجيل مقدرات تسجيل لجميع مكونات معمارية نظام الاختبار، أي، إن TE وTM وCH وCD وSA وPA قادرة على تسجيل معلومات تنفيذ الاختبار عبر TL. ويمثل الشكل 2 نظرة تفصيلية أكثر على TL.



الشكل Z.145/2 – نظرة تفصيلية على TL

1.1.6 إدارة اختبار والتحكم فيه (TMC)

يشمل كيان TMC الوظيفية المتعلقة بإدارة:

- تنفيذ اختبار؛
- مكونات؛
- تشفير وتفكيك تشفير.

1.1.1.6 إدارة اختبار (TM)

إن كيان TM مسؤول عن الإدارة الشاملة لنظام الاختبار. وبعد تدميث نظام الاختبار، يبدأ تنفيذ الاختبار في كيان TM. والكيان مسؤول عن التنفيذ الصحيح لوحدات TTCN-3، أي انتشار معلمات وحدة مثل معلومات IXIT إلى TE إذا لزم الأمر. ونمطياً، ينفذ هذا الكيان أيضاً السطح البيني لمستعمل نظام اختبار.

2.1.1.6 تشفير وفك تشفير (CD)

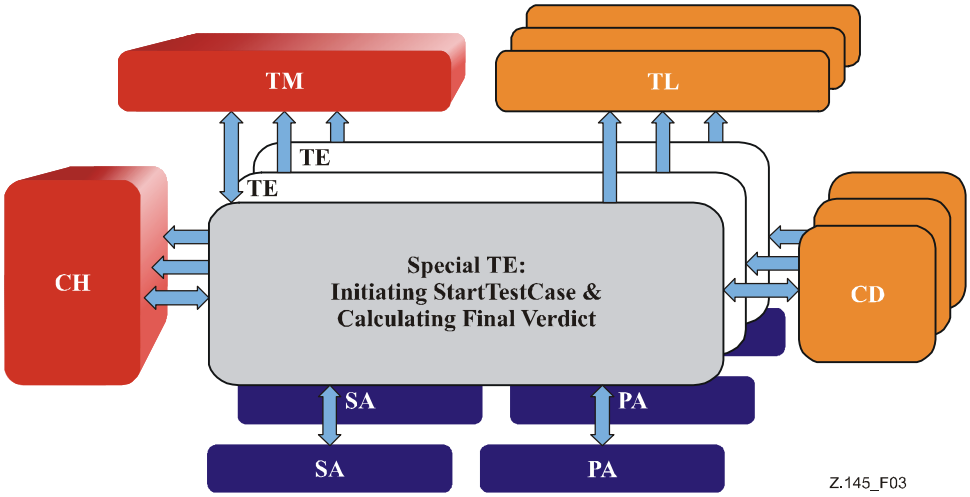
إن كيان CD مسؤول عن تشفير فك تشفير قيم TTCN-3 في سلاسل ثنائية مناسبة لترسل إلى نظام تحت الاختبار. ويحدد TE أي من الكودكات تستخدم. ويرسل معطيات TTCN-3 إلى المشفر الملائم للحصول على معطيات مشفرة. ويفك تشفير المعطيات المستقبلة في كيان CD باستخدام مشفر ملائم، الذي يترجم المعطيات المستقبلة إلى قيم TTCN-3.

3.1.1.6 مناولة مكون (CH)

يمكن توزيع TE فيما بين أجهزة اختبار متعددة. وتنفذ CH الاتصالات بين كيانات نظام اختبار موزعة. ويوفر كيان CH وسائل لتزامن كيانات نظام اختبار يمكن أن توزع في عقد عديدة.

الملاحظة 1 – تستخدم العقد وأجهزة اختبار باعتبارها مترادفات.

يرد في الشكل 3 الهيكل العام لنظام اختبار موزع فيما بين عقد عديدة.



الشكل Z.145/3 – هيكل عام لنظام اختبار TTCN-3 موزع

تشمل كل عقدة في نظام اختبار كيانات TE وSA وPA وCD وTL. ويقوم الكيانان CH وTM بالوساطة بين إدارة الاختبار ومناولة مكون اختبار بين TEs على كل عقدة. ويكون TE الذي يبدأ حالة اختبار هو TE خاص. ويقوم بحساب الحكم النهائي على اختبار مجرد. وبجانب هذا، تتم مناولة جميع TEs بنفس الطريقة.

الملاحظة 2 – ينفذ نظام اختبار اختبار مجرد واحد على الأكثر في أي وقت من الأوقات. أي، لا يمكن لوحدة TTCN-3 تنفيذ اختبارات مجردة متعددة في نفس الوقت.

إن إنشاء MTC وPTCs ومكون التحكم في TEs يتحكم فيها CH. رجاء ملاحظة الدور الخاص لمكون النظام، الذي يوجد فقط مفهومياً وليس مكون اختبار ينفذ في TE. ويمكن توزيع منافذ نظام، أي، منافذ مكون نظام، فيما بين عقد عديدة. وفضلاً عن ذلك، يمكن أن يكون لمكونات اختبار ذات عقد مختلفة نفاذ إلى نفس المنفذ المادي لـــ SUT، أي، يمكن أن تتقابل مع نفس منفذ السطح البيني لنظام الاختبار.

مثـــال: يمكن تحقيق نفاذ إلى منافذ SUT حقيقية بواسطة TEs عبر وكلاء محليين.

إن الاتصالات بين مكونات TTCN-3 إما أن تقوم على أساس رسالة أو إجراء. ولهذا، يكيف CH الاتصالات القائمة على رسالة وإجراء لمكونات TTCN-3 على منصة تنفيذ معينة لنظام اختبار. وتكون على وعي بالتوصيلات بين منافذ اتصالات مكون اختبار TTCN-3. وتقوم بانتشار طلب إرسال عمليات من مكون TTCN-3 واحد إلى مكون TTCN-3 آخر. وقد يوجد المكون المستقبل في حالة مختلفة لنفس TE الموجود على عقدة مختلفة. ثم يبلغ TE بأي أحداث اختبار مستقبلة بواسطة الانتظار في صف من صفوف انتظار المنفذ لـــ TE.

تكون عمليات الاتصالات القائمة على إجراء بين مكونات TTCN-3 مرئية عند CH. وتميز CH بين الأنواع المختلفة للاتصالات القائمة على إجراء، أي، نداء وإجابة وتنفيذ، وتعمل على انتشارها بطريقة ملائمة على TE حيث يوجد المكون المستهدف. وتتم مناولة علم دلالات الاتصالات القائمة على إجراء TTCN-3 ، أي، تأثير هذه العملية على تنفيذ مكون اختبار TTCN-3، في TE.

هناك حاجة إلى اتصالات إضافية لتنفيذ مكونات اختبار في عقد عديدة. وتشمل اتصالات إدارة مكون دلالة إنشاء مكونات اختبار، وبدء تنفيذ مكون الاختبار وتوزيع الحكم وكذلك دلالة إنهاء المكون. ولا تنفذ CH سلوك مكون TTCN-3. ولكن بالأحرى، تنفذ الاتصالات بين مكونات عديدة نفذها TE.

4.1.1.6 تسجيل اختبار (TL)

يؤدى كيان TL تسجيل حدث اختبار وتقديمه إلى مستعمل نظام الاختبار. ويوفر تسجيل المعلومات حول تنفيذ الاختبار مثل أي مكونات اختبار تم إنشاؤها وبدأت وانتهت، وأي معطيات أرسلت إلى SUT، واستقبلت من SUT وتمت مواءمتها مع مقاييس TTCN-3، وأي مؤقتات بدأت أو توقفت أو أمهلت وما إلى ذلك.

2.1.6 TTCN-3 قابل للتنفيذ (TE)

ينفذ أو يفسر كيان TE وحدة TTCN-3. ومفهومياً، يمكن تجزئة TE إلى ستة كيانات متفاعلة: كيان Control وBehavior وComponent وType وValue وQueue. وتعرف التجزئة الهيكلية هذه في [1] ITU-T Z.144. وتستخدم مصطلحات TE المعرفة في [1] ITU-T Z.144 في التوصية هذه.

3.1.6 مُكيف SUT (SA)

إن SA هو تنفيذ مُكيف نظام تحت الاختبار (SA) كما عرف في [1]. وتستخدم مصطلحات SA المعرفة في [1] ITU-T Z.144 في التوصية هذه.

4.1.6 مُكيف منصة (PA)

إن PA هو تنفيذ مُكيف منصة كما عرف في [1] ITU-T Z.144. وتستخدم مصطلحات PA المعرفة في [1] ITU-T Z.144 في التوصية هذه.

2.6 متطلبات تنفيذ نظام اختبار TTCN-3

يعامل كل نداء عملية TCI كعملية ذرية في الكيان الطالب. ويعيد الكيان المطلوب، الذي ينفذ عملية TCI، التحكم إلى الكيان الطالب بمجرد انجاز تأثيره المقصود أو إذا لم يكن من الممكن إكمال العملية بنجاح. ولا يُسد الكيان المطلوب عند تنفيذ اتصالات قائمة على إجراء.

وكما ورد قبلاً، ليس هناك افتراض بان نظام اختبار TTCN-3 أو كيانات فردية قد نفذت في كيان وحيد قابل للتنفيذ أو عملية أو ما إذا كانت موزعة فيما بين عمليات أو حتى أجهزة اختبار.

يوفي تنفيذ TCI بالمتطلبات الواردة أعلاه.

7 السطح البيني للتحكم في TTCN-3 والعمليات

يعرف هذا القسم مجموعة أنماط معطيات مجردة مستخدمة لتمثل اتصالات معطيات بين TE وTMC. وبالإضافة إلى ذلك، يعرف عمليات TCI على أساس توقيعاتها، عندما تستخدم وآثارها على نظام اختبار TTCN-3.

ويشمل هذا التعريف أيضاً وصفاً تفصيلياً أكثر لمعلمات الدخل المطلوبة لكل نداء عملية TCI وقيمتها المعادة.

1.7 نظرة شاملة على TCI

يعرف TCI التفاعل بين كيانات TTCN-3 قابل للتنفيذ وCH وTM وCD وTL في نظام اختبار TTCN-3. ويوفر وسائل لـــ TE إلى:

- إدارة تنفيذ اختبار؛
- توزيع تنفيذ مكونات اختبار فيما بين أجهزة اختبار مختلفة؛
- تشفير معطيات اختبار وتفكيك تشفيرها؛
- تسجيل معلومات عن تنفيذ اختبار.

ويتألف TCI من أربعة سطوح بينية فرعية:

- **سطح بيني لإدارة اختبار TCI (TCI-TM)**: يشمل السطح البيني هذا جميع العمليات المطلوبة لإدارة تنفيذ اختبار وتوفير معلمات وحدة وثوابت خارجية وتوفير تسجيل حدث اختبار.
- **سطح بيني لمناولة مكون TCI (TCI- CH)**: يتألف السطح البيني هذا من العمليات المطلوبة لتنفيذ إدارة والاتصالات بين مكونات اختبار TTCN-3 في نظام اختبار مركزي أو موزع. ويشمل عمليات لإنشاء وبدء ووقف مكونات اختبار وإنشاء توصيل بين مكونات TTCN-3 وإدارة مكونات اختبار وأحكامها، ومناولة اتصالات قائمة على رسالة وإجراء بين مكونات TTCN-3.
- **سطح بيني لتشفير/تفكيك تشفير TCI (TCI- CD)**: يشمل السطح البيني هذا جميع العمليات المطلوبة لاسترداد ونفاذ كودكات، أي، مشفرات أو مفككات شفرة، لتشفير معطيات ترسل، معرفة باستخدام نعت تشفير TTCN-3 وتفكيك تشفير معطيات مستقبلة.
- **سطح بيني لتسجيل اختبار TLI (TCI-TL)**: يشمل السطح البيني هذا جميع العمليات لاسترداد معطيات عن تنفيذ اختبار والتحكم في مستوى تفاصيل هذه المعلومات.

إن جميع السطوح البينية ثائية الاتجاه بحيث توجد الأجزاء الطالبة والمطلوبة في TE وفي نظام اختبار TMC. ويجرى جمع السطوح البينية الموفرة (تلك العمليات التي يوفر فيها سطح بيني لـــ TE) والعمليات المطلوبة (تلك العمليات التي يحتاج فيها سطح بيني أن يستخدم من TE) في السطح البيني الفرعي الموفر والمطلوب لكل سطح بيني، أي، TCI-TM Provided/TCI-TM Required وTCI-CH Provided/TCI-CH Required وTCI-CD Provided/TCI-CD Required وTCI-TL Provided/TCI-TL Required

1.1.7 الترابط بين TTCN-3 وتنفيذ عمليات TCI

لبعض عمليات تنفيذ TTCN-3، هناك ترابط مباشر بتنفيذ عملية TCI، التي تظهر في الجدول 1. وتترابط بعض عمليات TTCN-3 مع زوجين لطلب عملية TCI وعملية TCI لتنفيذ انتشار عمليات TTCN-3 خلال نظام الاختبار. وبالنسبة لتنفيذ عمليات TCI الأخرى، هناك ترابط غير مباشر – فهي مطلوبة لتنفيذ علم دلالات TTCN-3 لمفاهيم تحتية.

يصلح الترابط المبين لعمليات اتصالات TTCN-3 (أي، send وcall وreply وraise) فقط إذا نفذت هذه العمليات على منفذ مكون اختبار موصول بمنفذ مكون اختبار آخر. ويعرف الترابط لعمليات اتصالات تنفذ على منافذ مكونات اختبار متقابلة مع منافذ سطوح بينية لنظام اختبار في ITU-T Z.144 [1].

الجدول Z.145/1 – الترابط بين TTCN-3 وتنفيذ عمليات TCI

TTCN-3 operation name	TCI operation name	TCI interface name
send	tciSendConnected (see Note 1)	TCI-CH Provided
	tciSendConnectedBC (see Note 2)	
	tciSendConnectedMC (see Note 3)	
	tciEnqueueMsgConnected	TCI-CH Required
call	tciCallConnected (see Note 1)	TCI-CH Provided
	tciCallConnectedBC (see Note 2)	
	tciCallConnectedMC (see Note 3)	
	tciEnqueueCallConnected	TCI-CH Required
reply	tciReplyConnected (see Note 1)	TCI-CH Provided
	tciReplyConnectedBC (see Note 2)	
	tciReplyConnectedMC (see Note 3)	
	tciEnqueueReplyConnected	TCI-CH Required
raise	tciRaiseConnected (see Note 1)	TCI-CH Provided
	tciRaiseConnectedBC (see Note 2)	
	tciRaiseConnectedMC (see Note 3)	
	tciEnqueueRaiseConnected	TCI-CH Required
create	tciCreateTestComponentReq	TCI-CH Provided
	tciCreateTestComponent	TCI-CH Required
start (a component)	tciStartTestComponentReq	TCI-CH Provided
	tciStartTestComponent	TCI-CH Required
stop (a component)	tciStopTestComponentReq	TCI-CH Provided
	tciStopTestComponent	TCI-CH Required
kill	tciKillTestComponentReq	TCI-CH Provided
	tciKillTestComponent	TCI-CH Required
connect	tciConnectReq	TCI-CH Provided
	tciConnect	TCI-CH Required
disconnect	tciDisconnectReq	TCI-CH Provided
	tciDisconnect	TCI-CH Required
map	tciMapReq	TCI-CH Provided
	tciMap	TCI-CH Required
unmap	tciUnmapReq	TCI-CH Provided
	tciUnmap	TCI-CH Required
running	tciTestComponentRunningReq	TCI-CH Provided
	tciTestComponentRunning	TCI-CH Required
alive	tciTestComponentAliveReq	TCI-CH Provided
	tciTestComponentAlive	TCI-CH Required

الجدول Z.145/1 – الترابط بين TTCN-3 وتنفيذ عمليات TCI

TTCN-3 operation name	TCI operation name	TCI interface name
done	tciTestComponentDoneReq	TCI-CH Provided
	tciTestComponentDone	TCI-CH Required
killed	tciTestComponentKilledReq	TCI-CH Provided
	tciTestComponentKilled	TCI-CH Required
mtc	tciGetMTCReq	TCI-CH Provided
	tciGetMTC	TCI-CH Required
execute	tciTestCaseExecuteReq	TCI-CH Provided
	tciTestCaseExecute	TCI-CH Required
log	tliLog	TCI-TL Provided
الملاحظة 1 – للاتصال الوحيد. **الملاحظة 2** – للاتصال الإذاعي. **الملاحظة 3** – للاتصال المتعدد.		

2.7 معطيات TCI

تعرف مواصفة TCI مجموعة أنماط معطيات مجردة. وتصف هذه الأنماط، على مستوى عالي جداً، أي نوع من المعطيات تمرر من الكيان الطالب إلى الكيان المطلوب. وتستخدم أنماط المعطيات المجردة لتحديد:

- كيف تمرر معطيات TTCN-3 من TE إلى مشفر، وتشفير تمثيل قيمة TTCN-3 في سلسلة ثنائية؛ وفي الحالة العكسية؛
- كيف مرت المعطيات من المشفر إلى TE لتشفر من سلسلة ثنائية في تمثيل قيمتها TTCN-3.

بالنسبة لأنماط المعطيات المجردة هذه، تعرف مجموعة العمليات لمعالجة المعطيات بواسطة مشفر/مفكك تشفير.

يعرف تقديم محسوس لأنماط المعطيات المجردة هذه وكذلك تعريف أنماط معطيات أساسية مثل string و Boolean في تقابلات لغة معينة في القسمين 8 و9.

لاحظ أن القيم لأي نمط معطيات مُعرف تكون وحيدة في تنفيذ نظام اختبار حيث تعرف الوحيدية على أنها متميزة عالمياً في وقت من الأوقات. ويضمن هذا أن الأشياء مختلفة، مثلاً، يعرف، مؤقتان، بواسطة معرفات مختلفة ومعرفات لا تستخدم.

1.2.7 الأنماط العامة لمعطيات مجردة

تعرف أنماط المعطيات المجردة التالية وتستخدم لتعريف عمليات TCI:

1.1.2.7 الإدارة

`TciModuleIdType` إن قيمة `TciModuleIdType` هي اسم وحدة TTCN-3 كما حددت في TTCN-3 ATS. ويستخدم النمط المجرد هذا لمناولة وحدة.

`TciModuleParameterIdType` إن قيمة `TciModuleParameterIdType` هي الاسم المؤهل لمعلمة وحدة TTCN-3 كما حددت في TTCN-3 ATS. ويستخدم النمط المجرد هذا لمناولة معلمة وحدة.

`TciTestCaseIdType` إن قيمة `TciTestCaseIdType` هي الاسم المؤهل لاختبار مجرد TTCN-3 كما حددت في TTCN-3 ATS. ويستخدم النمط المجرد هذا لمناولة اختبار مجرد.

`TciModuleIdListType` إن قيمة نمط `TciModuleIdListType` هي قائمة `TciModuleIdType`. ويستخدم النمط المجرد هذا عند استرداد قائمة وحدات استوردتها وحدة TTCN-3.

`TciModuleParameterType` إن قيمة نمط `TciModuleParameterType` هي هيكل `TciModuleParameterIdType` ويستخدم النمط المجرد هذا ليمثل اسم المعلمة والقيمة بالتغيب لمعلمة وحدة.

`TciModuleParameterListType` إن قيمة نمط `TciModuleParameterListType` هي قائمة `TciModuleParameterType`. ويستخدم النمط المجرد هذا عند استرداد معلمات وحدة لوحدة TTCN-3.

TciParameterType	إن قيمة نمط TciParameterType تشمل Value TTCN-3 وقيمة TciParameterPassingModeType لتمثل معلمة تمرر أسلوب محدد للمعلمة في TTCN-3 ATS.
TciParameterPassingModeType	إن قيمة نمط TciParameterPassingModeType هي إما IN أو INOUT أو OUT. ويستخدم النمط المجرد هذا عند بدء اختبار مجرد أو عند الدلالة على إنتهاء اختبار مجرد.
TciParameterListType	إن قيمة نمط TciParameterListType هي قائمة (TciParameterType). ويستخدم النمط المجرد هذا عند بدء اختبار مجرد أو عند الدلالة على انتهاء اختبار مجرد.
TciParameterTypeType	إن قيمة نمط TciParameterTypeType هي هيكل TciParameterPassingModeType. ويستخدم النمط المجرد هذا ليمثل نمط ومعلمة تمرير أسلوب لمعلمة اختبار مجرد.
TciParameterTypeListType	إن قيمة نمط TciParameterTypeListType هي قائمة TciParameterTypeType. ويستخدم النمط المجرد هذا ليمثل قائمة معلمات اختبار مجرد.
TciTestComponentKindType	إن قيمة نمط TciTestComponentKindType هي حرفي لمجموعة من أنواع مكونات اختبار TTCN-3، أي MTC، PTC، PTC_ALIVE، وCONTROL. ويستخدم النمط المجرد هذا لمناولة مختصة.
TciTypeClassType	إن قيمة نمط TciTypeClassType هي حرفي لمجموعة أصناف نمط في TTCN-3 مثل بولاني وطليق وسجل وما إلى ذلك. ويستخدم النمط المجرد هذا لمناولة قيمة.

2.1.2.7 الاتصالات

TciBehaviourIdType	تعرف قيمة نمط TciBehaviourIdType وظائف سلوك TTCN-3.

تؤخذ أنماط معطيات مجردة إضافية ذات سابقة Tri من ITU-T Z.144 [1]: TriPortIdType، TriPortIdListType، TriComponentIdType، TriComponentIdListType، TriAddressType، TriAddressListType، TriMessageType.

2.2.7 أنماط وقيم معطيات مجردة TTCN-3

يعرف هذا القسم مجموعة أنماط معطيات تبني نمط TTCN-3 وعرض قيمة. وتعرف وظيفية كل نمط معطيات بواسطة مجموعة العمليات المتصاحبة. وتعيد العمليات في أو المستخدمة لنمط المعطيات هذا إما قيمة نمط المعطيات هذا أو نمط أساسي مثل boolean.

تم تعريف جميع العمليات باستخدام لغة وصف سطح بيني (IDL). ويرد في القسمين 8 و9 تقابلات لغة محسوسة لعمليات على أنماط معطيات مجردة. وفي بعض اللغات، يمثل تطبيق عملية على نمط معطيات مجردة بواسطة تمرير (سواء حسب القيمة أو حسب المرجع، يعتمد على التقابل) القيمة المحسوسة كمعلمة للعملية. وقد تختار لغات أخري طرق مرجعية أخرى للقيمة المحسوسة، مثل، اعتبار القيمة كشيء ينفذ بطريقة تتوافق فيه مع العملية. وللدلالة على عدم القدرة على أداء مهمة معينة أو الدلالة على غياب معلمة تشغيلية فيما يلي، تستخدم القيمة المميزة null. ويمكن اعتبارها على أنها قيمة محجوزة تدل على قيمة خاصة. وتعرف تقابلات لغة عرض محسوس للقيمة المميزة null هذه.

يتألف نمط TTCN-3 المجرد وعرض قيمة من جزئين:

- نمط معطيات مجرد Type يمثل جميع أنماط TTCN-3 في وحدة TTCN-3؛
- أنماط معطيات مجردة مختلفة تمثل قيم TTCN-3، أي، قيم TTCN-3 لنمط TTCN-3 معين. ويمكن أن يكون هذا إما قيما لـــ TTCN-3 لأنماط معرفة مسبقا أو لأنماط معرفة لمستعمل TTCN-3.

وبالنسبة لنفاذ وتقييم وتشفير معطيات TTCN-3، يستخدم نظام الاختبار نمط معطيات مجرد Type وأنماط معطيات قيمة مجردة مختلفة. ولهذا، تعرف أنماط المعطيات المجردة هذه مستوى التجرد بين TTCN-3 قابل للتنفيذ ونظام الاختبار المتبقي باستخدام السطوح البينية TCI.

1.2.2.7 أنماط معطيات TTCN-3 مجردة

طبقاً لأنماط TTCN-3 لهذه التوصية، سواء معرفة مسبقا أو معرفة لمستعمل، تمثل عند السطوح البينية TCI باستخدام نمط معطيات مجرد Type.

بالنسبة لنمط معطيات مجرد Type، تعرف مجموعة عمليات:

- أنماط معطيات TTCN-3 معرفة مسبقا لمرجع ومعرفة لمستعمل؛
- وضع قيم TTCN-3 والاحتفاظ بها.

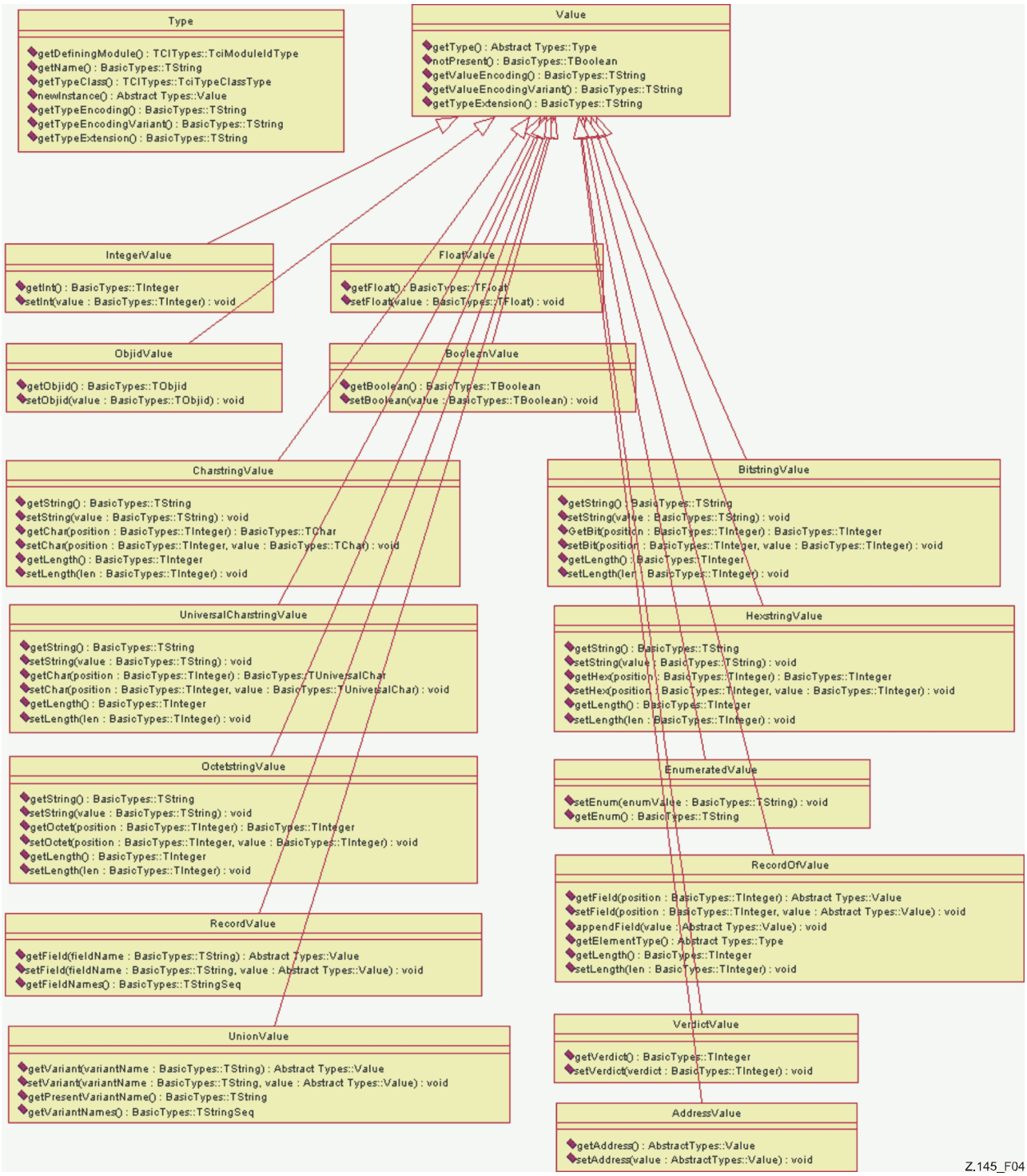
تعرف العمليات التالية لنمط معطيات مجرد Type:

`TciModuleIdType getDefiningModule()`	يعيدُ معرف الوحدة للوحدة التي يعرف فيها النمط. يعيد القيمة المميزة `null` إذا كان النمط هو نمط قاعدة TTCN-3، مثل، بولاني وصحيح وما إلى ذلك
`TString getName()`	يعيد اسم النمط كما عرف في وحدة TTCN-3.
`TciTypeClassType getTypeClass()`	يعيد صنف النمط للنمط المعني. ويمكن لقيمة (`TciTypeClassType`) أن يكون لها واحد من الثوابت التالية: `ADDRESS`، `ANYTYPE`، `BITSTRING`، `BOOLEAN`، `CHARSTRING`، `COMPONENT`، `ENUMERATED`، `FLOAT`، `HEXSTRING`، `INTEGER`، `OBJID`، `OCTETSTRING`، `RECORD`، `RECORD_OF`، `SET`، `SET_OF`، `UNION`، `UNIVERSAL_CHARSTRING`، `VERDICT`.
`Value newInstance()`	يعيد قيمة موضوعة جديدة لنمط معين. و لا تعرف القيمة الأولية هذه للقيمة الموضوعة.
`TString getTypeEncoding()`	يعيد نعت تشفير النمط كما عرف في وحدة TTCN-3.
`TString getTypeEncodingVariant()`	تعيد هذه العملية نعت متغير لتشفير القيمة كما عرف في TTCN-3، إن وجد. وإذا لم يعرف نعت متغير لتشفير تعاد القيمة المميزة `null`.
`TStringseq getTypeExtension()`	يعيد نعت تمديد النمط كما عرف في وحدة TTCN-3.

2.2.2.7 قيم TTCN-3 مجردة

تمثل قيم TTCN-3، طبقاً لهذه التوصية، عند السطوح البينية TCI عبر أنماط معطيات مجردة عديدة.

يعرض الشكل 4 التراتبية بين أنماط معطيات مجردة لقيم TTCN-3 (اختصار: قيم مجردة).



الشكل Z.145/4 – تراتبية قيم مجردة

كما يبين الشكل 4، تتقاسم جميع القيم المجردة TTCN-3 نفس نمط معطيات مجرد لقاعدة `Value`. إن جميع العمليات المعرفة في نمط معطيات لقاعدة مشتركة تعرف ضمنيا أيضاً لأنماط قيمة مجردة مشتقة منها.

1.2.2.2.7 نمط معطيات مجرد `Value`

تعرف العمليات التالية على نمط معطيات مجرد لقاعدة `Value`. ويعرف العرض المحسوس لهذه العمليات في أقسام تقابل لغة معنية:

`Type getType()` يعيد نمط قيمة محددة.

`TBoolean notPresent()` يعيد `true` `omit`، إذا كانت القيمة المحددة هي `omit`، `false` وإلا.

TString getValueEncoding()	يعيد نعت تشفير القيمة كما عرف في TTCN-3، إن وجد. وإذا لم يعرف نعت تشفير، تعاد القيمة المميزة null.
TString getValueEncodingVariant()	يعيد نعت متغير لتشفير القيمة كما عرف في TTCN-3، إن وجد. وإذا لم يعرف نعت تشفير، تعاد القيمة المميزة null.

2.2.2.2.7 نمط معطيات مجرد IntegerValue

يقوم نمط معطيات مجرد IntegerValue على أساس نمط معطيات مجرد Value. ويمثل قيم integer TTCN-3.

تعرف العمليات التالية لنمط معطيات مجرد IntegerValue:

TString getValueEncoding()	يعيد قيمة صحيح للصحيح TTCN-3 هذا.
TString getValueEncodingVariant()	يضبط IntegerValue هذا على value.

3.2.2.2.7 نمط معطيات مجرد FloatValue

يقوم نمط معطيات مجرد FloatValue على أساس نمط معطيات مجرد Value. ويمثل قيم float TTCN-3.

تعرف العمليات التالية لنمط معطيات مجرد FloatValue:

TFloat getFloat()	يعيد قيمة الطليق لطليق TTCN-3 هذا.
void setFloat(in TFloat value)	يضبط FloatValue هذا على قيمة value.

4.2.2.2.7 نمط معطيات مجرد FloatValue.

يقوم نمط معطيات مجرد BooleanValue على أساس نمط معطيات مجرد Value. ويمثل قيم Boolean TTCN-3.

تعرف العمليات التالية لنمط معطيات مجرد BooleanValue:

TBoolean getBoolean()	يعيد قيمة بولاني لـــ boolean TTCN-3.
void setBoolean(in TBoolean value)	يضبط قيمة بولاني على value.

5.2.2.2.7 نمط معطيات مجرد ObjidValue.

يقوم نمط معطيات مجرد ObjidValue.على أساس نمط معطيات مجرد Value. ويمثل قيم objid.TTCN-3.

تعرف العمليات التالية لنمط معطيات مجرد ObjidValue:

TObjid getObjid()	يعيد قيمة تعرف شيء لـ TObjid getObjid TTCN-3
void setObjid(in TObjid value)	يضبط value هذا على objid.

6.2.2.2.7 نمط معطيات مجرد CharstringValue.

يقوم نمط معطيات مجرد CharstringValue على أساس نمط معطيات مجرد Value. ويمثل قيم charstring TTCN-3. ويكون TChar هو سمة في قيمة سلسلة سمات.

تعرف العمليات التالية لنمط معطيات مجرد CharstringValue:

TString getString()	يعيد قيمة سلسلة charstring.TTCN-3. ويكون العرض النصي لـ charstring. TTCN-3 فارغ هو ''، بينما طوله صفراً.
void setString(in TString value)	يضبط CharstringValue.هذا على value.
TChar getChar(in TInteger position)	يعيد قيمة سمة لسلسلة سمات عند position. ويدل موضع صفر على السمة الأولى لـ charstring TTCN-3. وتكون القيم الصالحة للموضع هي من 0 إلى length - 1.
void setChar(in TInteger position, in TChar value)	
TInteger getLength()	اضبط السمة عند الموضع على قيمة. وتكون القيم الصالحة للموضع هي من 0 إلى length - 1.
TInteger getLength	يعيد طول CharstringValue في سمات، صفر إذا كانت القيمة لـ CharstringValue هذه محذوفة.
void setLength(in TInteger len)	يعيد تدميث setlength الأول CharstringValue هذه إلى قيمتها الأولية ويضبط بعد ذلك طول CharstringValue هذه في سمات على len.

7.2.2.2.7 نمط معطيات مجرد UniversalCharstringValue.

يقوم نمط معطيات مجرد UniversalCharstringValue على أساس نمط معطيات مجرد Value. ويمثل قيم universal charstring TTCN-3. ويكون TUniversalChar هو سمة في قيمة سلسلة سمات عالمية.

تعرف العمليات التالية لنمط معطيات مجرد UniversalCharstringValue:

TString getString()	يعيد العرض النصي لـــ UniversalCharstringValue هذا كما عرف في TTCN-3.
void setString(in TString value)	يضبط قيمة UniversalCharstringValue هذا طبقاً للعرض النصي كما عرف بواسطة Value.
TUniversalChar getChar(in TInteger position)	
TInteger getLength	يعيد قيمة السمة العالمية لـــ universal charstring TTCN-3 عند موضع. ويدل موضع صفر على أن TUniversalChar الأول لـــuniversal charstring TTCN-3. وتكون القيم الصالحة لموضع هي من صفر إلى length - 1.
void setChar(in TInteger position, in TUniversalChar value)	
	يضبط سمة عالمية عند موضع على قيمة. وتكون القيم الصالحة لموضع هي من صفر إلى length - 1.
TInteger getLength()	يعيد طول قيمة سلسلة سمات عالمية هذا في سمات عالمية، صفر إذا كانت القيمة لقيمة سلسلة سمات عالمية هذه محذوفة.
void setLength(in TInteger len)	يعيد تدميث setlength الأول UniversalCharstringValue إلى قيمتها الأولية ويضبط بعد ذلك طول UniversalCharstringValue هذه في سمات عالمية على len.

7.2.2.2.8 نمط معطيات مجرد BitstringValue.

يقوم نمط معطيات مجرد BitstringValue على أساس نمط معطيات مجرد Value. ويمثل قيم TTCN-3 values bitstring:

تعرف العمليات التالية لنمط معطيات مجرد BitstringValue.

TString getString()	يعيد العرض النصي لـــ BitstringValue هذا، كما عرف في TTCN-3. مثلاً، عرض نصي لـــ 0101. والعرض النصي لـــ B"0101" لـــ TTCN-3 B""، بينما طوله هو صفر.
void setString(in TString value)	يضبط قيمة BitstringValue.هذا طبقاً للعرض النصي كما عرف بواسطة Value. مثلاً، تكون القيمة لـــ BitstringValue.هذا هي 0101. إذا كان العرض النصي في Value هو B"0101".
TChar getBit(in TInteger position)	يعيد قيمة (1 | 0) عند موضع سلسلة بتات TTCN-3 هذه باعتبارها سمة. ويدل موضع صفر على البتة الأولي لسلسلة بتات TTCN-3. وتكون القيم الصالحة لموضع هي من صفر إلى length - 1.
void setBit(in TInteger position, TInteger value)	
	يضبط البتة عند موضع على القيمة (1 | 0) ويدل موضع صفر على البتة الأولي لسلسلة بتات TTCN-3. وتكون القيم الصالحة لموضع هي من صفر إلى length - 1.
TInteger getLength()	يعيد طول BitstringValue هذه في بتات، صفر إذا كانت قيمة BitstringValue هي omit.
void setLength(in TInteger len)	يعيد تدميث setlength الأول UniversalCharstringValue إلى قيمتها الأولية ويضبط بعد ذلك طول BitstringValue هذه في سمات عالمية على len.

9.2.2.2.7 نمط معطيات مجرد OctetstringValue.

يقوم نمط معطيات مجرد OctetstringValue.على أساس نمط معطيات مجرد Value. ويمثل قيم octetstring TTCN-3.

تعرف العمليات التالية لنمط معطيات مجرد OctetstringValue:

TString getString()	يعيد العرض النصي لـــ OctetstringValue.هذا، كما عرف في TTCN-3. مثلاً، عرض نصي لـــ 0xCAFFEE هو O"CAFFEE". والعرض النصي لـــ O"CAFFEE" لـــ TTCN-3 فارغ هو octetstring، بينما طوله هو صفر.

void setString(in TString value)	يضبط قيمة OctetstringValue.هذا طبقاً للعرض النصي كما عرف بواسطة Value. مثلاً، تكون القيمة لـــ OctetstringValue. هذا هي 0xCAFFEE. إذا كان العرض النصي في Value هو O"CAFFEE".
TChar getOctet(in TInteger position)	يعيد قيمة (0..255) عند موضع سلسلة أثمونات TTCN-3 هذه. ويدل موضع صفر على الأثمون الأول لسلسلة أثمونات TTCN-3. وتكون القيم الصالحة لموضع هي من صفر إلى length - 1.
void setOctet(in TInteger position, in TInteger value)	يضبط الأثمون عند موضع على القيمة (0..255). ويدل موضع صفر على الأثمون الأول لسلسلة أثمونات TTCN-3. وتكون القيم الصالحة لموضع هي من صفر إلى length - 1.
TInteger getLength()	يعيد طول OctetstringValue هذه في أثمونات، صفر إذا كانت قيمة OctetstringValue هي omit.
void setLength(in TInteger len)	يعيد تدميث setlength الأول OctetstringValue هذه إلى قيمتها الأولية ويضبط بعد ذلك طول OctetstringValue هذه في أثمونات على len.

10.2.2.2.7 نمط معطيات مجرد HexstringValue.

يقوم نمط معطيات مجرد HexstringValue.على أساس نمط معطيات مجرد Value. ويمثل قيم hexstring TTCN-3.

تعرف العمليات التالية لنمط معطيات مجرد HexstringValue:

TString getString()	يعيد العرض النصي لـــ HexstringValue.هذا، كما عرف في TTCN-3. مثلاً، عرض نصي لـــ 0xAFFEE هو H"AFFEE". والعرض النصي لـــ hexstring لـــ TTCN-3 فارغ هو H""، بينما طوله هو صفر.
void setString(in TString value)	يضبط قيمة HexstringValue هذا طبقاً للعرض النصي كما عرف بواسطة Value. مثلاً، تكون القيمة لـــ HexstringValue هذا هي 0xAFFEE. إذا كان العرض النصي في Value هو H"AFFEE".
TChar getHex(in TInteger position)	يعيد قيمة (0..15) عند موضع سلسلة ست عشرية TTCN-3 هذه. ويدل موضع صفر على الأرقام الست عشرية الأولي لسلسلة ست عشرية TTCN-3. وتكون القيم الصالحة لموضع هي من صفر إلى length - 1.
void setHex(in TInteger position, in TInteger value)	يضبط الرقم الست عشري عند موضع على قيمة (0..15). ويدل موضع صفر على الأثمون الأول في hexstring. وتكون القيم الصالحة لموضع هي من صفر إلى length - 1.
TInteger getLength()	يعيد طول HexstringValue هذه في أثمونات، صفر إذا كانت قيمة HexstringValue هي omit.
void setLength(in TInteger len)	يعيد تدميث setlength الأول HexstringValue هذه إلى قيمتها الأولية ويضبط بعد ذلك طول HexstringValue هذه في أرقام ست عشرية على len.

11.2.2.2.7 نمط معطيات مجرد RecordValue

يقوم نمط معطيات مجرد RecordValue على أساس نمط معطيات مجرد Value. ويحدد كيفية الحصول وضبط نمط TTCN-3 record. وينطبق نفس نمط معطيات مجرد لقيم يكون صنف نمطها هو SET. وهذا التمييز بين record وset يتعلق فقط عند وقت المواءمة.

تعرف العمليات التالية لنمط معطيات مجرد RecordValue:

Value getField(in TString fieldName)	يعيد القيمة للمجال المسمي fieldName وتكون القيمة المعادة هي نمط قاعدة مجردة مشتركة Value، وباعتبارها مجال مسجل يمكن أن يكون لها أي نمط معرف في TTCN-3. وإذا لم يكن من الممكن الحصول على المجال من السجل تعاد القيمة المميزة null.
void setField(in TString fieldName, in Value value)	يضبط المجال المسمي fieldName للسجل على Value. ولا تكون هناك افتراضات عن كيفية تخزين مجال في سجل. وقد يختار التنفيذ الداخلي تخزين مرجع لهذه القيمة أو نسخ القيمة. ومن الأسلم افتراض أن القيمة تنسخ. ولهذا، ينبغي افتراض أن التعديلات اللاحقة لـــ Value لا تنعكس في السجل.
TStringSeq getFieldNames	يعيد تتابع سلسلة أسماء مجالات وتتابع فارغ، إذا لم يكن للسجل مجالات.

12.2.2.2.7 نمط معطيات مجرد RecordOfValue

يقوم نمط معطيات مجرد RecordOfValue على أساس نمط معطيات مجرد Value. ويحدد كيفية الحصول وضبط مكونات في أنماط record of TTCN-3. وينطبق نفس نمط معطيات مجرد لقيمة يكون صنف نمطها هو SET_OF. وهذا التمييز بين record of. وset of يتعلق فقط عند وقت المواءمة.

تعرف العمليات التالية لنمط معطيات مجرد RecordOfValue:

Value getField(in TInteger position)

يعيد قيمة record of عند position إذا كان position.هو بين zero و length - 1، وإلا تكون القيمة المميزة null. وتكون القيمة المعادة هي نمط مجرد لقاعدة مشتركة Value، باعتباره record of يمكن أن يكون له مجالات من أي نمط معرف في TTCN-3.

void setField(in TInteger position, in Value value)

يضبط المجال position على value. إذا كان position.أكبر من (length - 1) السجل يمدد ليكون له طول .(position + 1). ويضبط record of لمكونات بين الموضع الأصلي عند length. و1 - position.على omit. ولا تكون هناك افتراضات عن كيفية تخزين مجال في record of. وقد يختار التنفيذ الداخلي تخزين مرجع لهذه القيمة أو نسخ القيمة. ومن الأسلم افتراض أن القيمة تنسخ. ولهذا، ينبغي افتراض أن التعديلات اللاحقة لـــ Value لا تنعكس في record of.

void appendField(in Value value)

تذيل القيمة عند طرف record of، أي، عند موضع length. ولا تكون هناك افتراضات عن كيفية تخزين مجال في record of. وقد يختار التنفيذ الداخلي تخزين مرجع لهذه القيمة أو نسخ القيمة. ومن الأسلم افتراض أن القيمة تنسخ. ولهذا، ينبغي افتراض أن التعديلات اللاحقة لـــ Value لا تنعكس في record of.

Type getElementType()

يعيد Type. لمكونات record of هذا.

TInteger getLength()

يعيد الطول الفعلي لـــ record of للقيمة، ويكون صفر إذا كانت قيمة record of هي omit.

void setLength(in TInteger len)

يضبط طول record of على len. وإذا كان len أكبر من الطول الأصلي، يكون للمكونات الجديدة قيمة omit. وإذا كان len أقل من أو يساوي الطول الأصلي، يتم تجاهل هذه العملية.

13.2.2.2.7 نمط معطيات مجرد UnionValue

يقوم نمط معطيات مجرد UnionValue على أساس نمط معطيات مجرد Value. ويحدد كيفية الحصول على وضبط متغيرات في نمط union. TTCN-3 ويمثل TTCN-3 anytype بواسطة UnionValue حيث صنف النمط للنمط الذي تم الحصول عليه بواسطة ()getType هو ANYTYPE. ولمزيد من التفاصيل عن أصناف أنماط، ارجع إلى 1.2.2.7.

تعرف العمليات التالية لنمط معطيات مجرد UnionValue:

Value getVariant(in TString variantName)

يعيد قيمة اتحاد variantName TTCN-3، إذا كان variantName مساو لنتيجة getPresentVariantName، وإلا تكون القيمة المميزة null ويدل variantName على اسم متغير اتحاد كما عرف في TTCN-3.

void setVariant(in TString variantName, in Value value)

يضبط variantName للاتحاد على قيمة. وإذا كان variantName غير معرف لهذا الاتحاد، يتم تجاهل هذه العملية. وإذا اختير متغير آخر، يختار المتغير الجديد بدلاً منه.

TString getPresentVariantName()

يعيد سلسلة تمثل اسم المتغير المختار الحالي في اتحاد TTCN-3 معين. وتعاد القيمة المميزة null. إذا لم يتم اختيار متغير.

TStringSeq getVariantNames()

يعيد تتابع سلسلة أسماء متغيرات، والقيمة المميزة null.، إذا لم يكن للاتحاد مجالات. وإذا مثلت UnionValue anytype TTCN-3، أي، صنف النمط للنمط الذي حصل عليه ()getType يكون ANYTYPE، تعاد جميع أنماط TTCN-3 المعرفة مسبقاً والمعرفة لمستعمل.

14.2.2.2.7 نمط معطيات مجرد EnumeratedValue

يقوم نمط معطيات مجرد EnumeratedValue على أساس نمط معطيات مجرد Value. ويحدد كيفية الحصول على وضبط enumerated TTCN-3.

تعرف العمليات التالية لنمط معطيات مجرد EnumeratedValue:

TString getEnum()	يعيد معرف السلسلة لـــ EnumeratedValue هذا. ويساوي هذا المعرف المعرف في مواصفة TTCN-3.
void setEnum(in TString enumValue)	يضبط العد على enumValue. وإذا لم يكن enumValue مسموحاً له بقيمة لهذا العد، يتم تجاهل العملية.

15.2.2.2.7 نمط معطيات مجرد VerdictValue.

يقوم نمط معطيات مجرد VerdictValue على أساس نمط معطيات مجرد Value. ويحدد كيفية الحصول وضبطTTCN-3 verdict.

تعرف العمليات التالية لنمط معطيات مجرد VerdictValue:

TInteger getVerdict()	يعيد قيمة الصحيح لـــ VerdictValue هذا. ويكون الصحيح هو واحد من الثوابت التالية: ERROR، FAIL، INCONC، NONE، PASS.
void setVerdict(in TInteger verdict)	يضبط VerdictValue على verdict. ولاحظ إن VerdictValue. يمكن ضبطه على أي من الأحكام الواردة أعلاه في أي وقت. ولا يؤدى VerdictValue.أي حسابات حكم كما يعرف في TTCN-3. فمثلاً، يعتبر قانونياً أن يضبط VerdictValue أولاً على ERROR ثم على PASS.

16.2.2.2.7 نمط معطيات مجرد AddressValue

تعرف العمليات التالية لنمط معطيات مجرد AddressValue. والعروض المحسوسة لهذه العمليات معرفة في أقسام تقابل اللغة المعنية:

Value getAddress()	يعيد قيمة العنوان، التي لم تعد من صنف نمط ADDRESS ولكن من نمط فعلي مستخدم لعنوان.
void setAddress(in Value value)	تضبط قيمة العنوان هذا على Value.

3.2.7 أنماط تسجيل مجردة

1.3.2.7 نمط معطيات مجرد TciValueTemplate

تعرف العمليات التالية لنمط معطيات مجرد TciValueTemplate والعروض المحسوسة لهذه العمليات معرفة في أقسام تقابل اللغة المعنية:

boolean isOmit()	يعيد true إذا كان المقاس هو مقاس حذف.
boolean isAny()	يعيد true إذا كان المقاس هو أي مقاس.
boolean isAnyOrOmit()	يعيد true إذا كان المقاس هو أي مقاس أو مقاس حذف.
TString getTemplateDef()	يعيد تعريف ذلك المقاس.

2.3.2.7 نمط معطيات مجرد TciNonValueTemplate

تعرف العمليات التالية لنمط معطيات مجرد TciNonValueTemplate. والعروض المحسوسة لهذه العمليات معرفة في أقسام تقابل اللغة المعنية:

boolean isAny()	يعيد true إذا كان المقاس هو أي مقاس.
boolean isAll()	يعيد true إذا كان المقاس هو لجميع المقاسات.
TString getTemplateDef()	يعيد تعريف ذلك المقاس.

3.3.2.7 قائمة القيم وأنماط Mismatch

تعرف أنماط المعطيات المجردة التالية وتستخدم لتسجيل الاختلافات بين القيم والمقاسات:

TciValueList	إن قيمة TciValueList هي قائمة قيم.
TciValueDifference	إن قيمة TciValueDifference هي هيكل يحتوي على قيمة ومقاس ووصف لسبب هذا الاختلاف.
TciValueDifferenceList	إن قيمة TciValueDifferenceList هي تتابع لاختلافات القيمة.

3.7 عمليات TCI

يحدد هذا القسم العمليات التي يمكن لـــ TTCN-3 قابل للتنفيذ أن يوفر نظام اختبار (*عمليات مطلوبة*) والتي يوفر وظيفتها نظام الاختبار لـــ TTCN-3 قابل للتنفيذ (*عمليات موفرة*).

يعكس المصطلحان "مطلوبة" و "موفرة" حقيقة أن هذه التوصية تعرف متطلبات TTCN-3 قابل للتنفيذ من وجهة نظر المستعمل. و"يطلب" المستعمل من TTCN-3 قابل للتنفيذ بعض الوظائف لبناء نظام اختبار كامل قائم على TTCN-3. وللإيفاء بمهمته، يتعين على TTCN-3 قابل للتنفيذ أن يبلغ المستعمل عن بعض الأحداث حيث يتعين على المستعمل أن "يوفر" هذه الإمكانية لـــ TTCN-3 قابل للتنفيذ.

تعرف جميع تعاريف العمليات في هذا القسم باستخدام IDL. وتعرف تقابلات لغة محسوسة في القسمين 8 و9. ويوفر الملحق B للسطح البيني للتسجيل تقابل بديل مع XML.

لكل نداء عملية TCI، تكون جميع معلمات *in* و*inout* و*out* الواردة في تعريف عملية معينة إلزامية. إن قيمة معلمة *required*. يحددها كيان طالب. ويشير كيان طالب إلى اتجاه النداء. وبالنسبة لعمليات على سطح بيني *مطلوب*، يكون الكيان الطالب هو نظام اختبار بينما الكيان المطلوب هو TTCN-3 قابل للتنفيذ. وبالنسبة لعمليات على سطح بيني *موفر*، يكون الكيان الطالب هو TTCN-3 قابل للتنفيذ بينما يكون نظام الاختبار هو الكيان المطلوب.

وبالمثل، تحدد قيمة معلمة *out*. الكيان المطلوب. وفي حالة معلمة *inout*، تكون القيمة هي التي يحددها أولاً الكيان الطالب ولكن يمكن استبدالها بقيمة جديدة من قبل الكيان المطلوب. ولاحظ أنه بالرغم من أن TTCN-3 تستخدم أيضاً *in* و*inout* و*out* لتعاريف التوقيعات، لا تتعلق الدلالات المستخدمة في مواصفة TCI IDL بتلك الواردة في مواصفة TTCN-3.

ينبغي أن تستخدم نداءات عملية القيمة المحجوزة لتدل على غياب معلمات. وتعرف القيم المحجوزة لهذه الأنماط في كل تقابل لغة ويشار إليها لاحقاً باعتبارها قيمة *null*.

وبالإضافة إلى ذلك، تستخدم أيضاً قيمة *null* لتدل على عدم القدرة على أداء مهمة معينة.

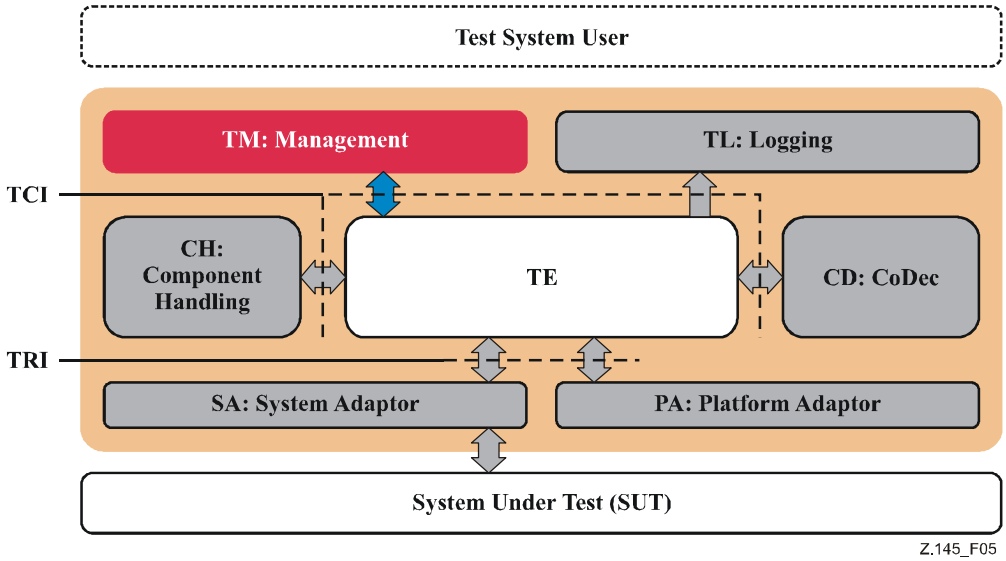
بما أن هذا القسم يحدد السطوح البينية فقط ولا يتناول تنفيذ محسوس عن كيفية أداء وظيفية محددة، يستخدم مصطلح "كيان" لتعريف الجزء من تنفيذ نظام اختبار الذي ينفذ هذا السطح البيني ويؤدى الوظيفية المطلوبة. فمثلاً، الكيان الطالب في عملية *tciSendConnected*. يكون TE، أي، الجزء من تنفيذ نظام الاختبار الذي يوفر وظيفية TE.

يجرى وصف جميع الوظائف في السطح البيني باستخدام المقاس التالي. أما الوصف الذي لا ينطبق على بعض العمليات فيجرى سحبه.

IDL Signature	**Signature**
Description of data passed as parameters to the operation from the calling entity to the called entity.	**In Parameters**
Description of data passed as parameters to the operation from the called entity to the calling entity.	**Out Parameters**
Description of data passed as parameters to the operation from the calling entity to the called entity and from the called entity back to the calling entity.	**InOut Parameters**
Description of data returned from the operation to the calling entity.	**Return Value**
Description of any constraints when the operation can be called.	**Constraint**
Behaviour required of the called entity before the operation may return.	**Effect**

1.3.7 السطح البيني TCM-TM

يصف السطح البيني لإدارة اختبار TCI (TCM-TM) العمليات التي يطلبها TTCN-3 قابل للتنفيذ وتنفيذ عمليات إدارة اختبار موفرة إلى TE (الشكل 5).



الشكل Z.145/5 – السطح البيني TCM-TM

يوفر تنفيذ إدارة اختبار الإدارة الشاملة لاختبار لمستعمل نظام اختبار. ويطلب من TE وجود عمليات لبدء وتوقف تنفيذ الاختبار لوحدة TTCN-3 أو لبعض اختبارات مجردة في وحدة TTCN-3. ويوفر بدوره عمليات إلى TE لحل معلمة وحدة عند التنفيذ والدلالة على انتهاء التنفيذ.

يوضح القسم 10 الاستخدام والترتيب اللاحق لنداءات عملية سواء بواسطة TE أو إدارة الاختبار.

1.1.3.7 TCM-TM المطلوبة

يحدد هذا القسم العمليات التي تطلبها TM من TE. وبالإضافة إلى العمليات المحددة في هذا القسم، تطلب إدارة اختبار عمليات كما تطلب عند السطح البيني TCI-CD.

1.1.1.3.7 tciRootModule

Signature	`void tciRootModule(in TciModuleIdType moduleName)`	
In Parameters	`moduleName`	The `moduleName` denotes the module identifiers as defined in TTCN-3.
Return Value	`void`	
Constraint	Shall be used only if neither the control part nor a test case is currently being executed.	
Effect	`tciRootModule` selects the indicated module for execution through a subsequent call using `tciStartTestCase` or `tciStartControl`. A `tciError` will be issued by the TE if no such module exists.	

2.1.1.3.7 getImportedModules

Signature	`TciModuleIdListType getImportedModules()`
Return Value	قائمة بجميع الوحدات المستوردة لوحدة الجذر. وترتب الوحدات حسب ظهورها في وحدة TTCN-3. وإذا لم توجد وحدات مستوردة، تعاد قائمة وحدات فارغة.
Constraint	تستخدم فقط إذا كانت وحدة الجذر قد ضبطت قبلاً.
Effect	يوفر TE للإدارة قائمة الوحدات المستوردة لوحدة الجذر. وإذا لم توجد وحدات مستوردة، تعاد قائمة وحدات فارغة. وإذا لم يستطع TE توفير قائمة، تعاد قيمة `null` المميزة.

tciGetModuleParameters 3.1.1.3.7

Signature	`TciModuleParameterListType tciGetModuleParameters(in TciModuleIdType moduleName)`	
In Parameters	`moduleName`	يدل `moduleName` على معرفات وحدة ينبغي على معلمات الوحدة استردادها.
Return Value	قائمة بجميع معلمات الوحدة المعرفة. وترتب المعلمات كما تظهر في وحدة TTCN-3. وإذا لم توجد معلمات، تعاد قائمة فارغة لمعلمات وحدة.	
Constraint	تستخدم فقط إذا كانت وحدة الجذر قد ضبطت قبلاً.	
Effect	يوفر TE إلى الإدارة قائمة بمعلمات الوحدة المعرفة. وإذا لم توجد معلمات وحدة، تعاد قائمة فارغة لمعلمات وحدة. وإذا لم يستطع TE أن يوفر قائمة، تعاد قيمة `null` المتميزة.	

tciGetTestCases 4.1.1.3.7

Signature	`TciTestCaseIdListType tciGetTestCases()`
Return Value	قائمة بجميع الاختبارات المجردة التي تكون سواء معرفة أو مستوردة في وحدة جذر.
Constraint	يستخدم فقط إذا كانت وحدة الجذر قد ضبطت قبلاً.
Effect	يوفر TE إلى الإدارة قائمة بالاختبارات المجردة. وإذا لم توجد اختبارات مجردة، تعاد قائمة فارغة لاختبارات مجردة. وإذا لم يستطع TE أن يوفر قائمة، تعاد قيمة `null` المتميزة.

tciGetTestCaseParameters 5.1.1.3.7

Signature	`TciParameterTypeListType tciGetTestCaseParameters(in TciTestCaseIdType testCaseId)`	
In Parameters	`testCaseId`	معرف اختبار مجرد كما عرف في وحدة TTCN-3.
Return Value	قائمة بجميع أنماط معلمة اختبار مجرد. وترتب أنماط معلمة كما تظهر في توقيع TTCN-3 لاختبار مجرد. وإذا لم توجد معلمات، تعاد قائمة فارغة لنمط معلمة.	
Constraint	يستخدم فقط إذا كانت وحدة الجذر قد ضبطت قبلا.	
Effect	يوفر TE إلى الإدارة قائمة بأنماط معلمة لاختبار مجرد. وإذا لم توجد معلمات اختبار مجرد، تعاد قائمة فارغة لنمط معلمة. وإذا لم يستطع TE أن يوفر قائمة، تعاد قيمة `null` المتميزة.	

tciGetTestCaseTSI 6.1.1.3.7

Signature	`TriPortIdListType tciGetTestCaseTSI(in TciTestCaseIdType testCaseId)`	
In Parameters	`testCaseId`	معرف اختبار مجرد كما عرف في وحدة TTCN-3.
Return Value	قائمة بجميع منافذ النظام لاختبار مجرد تم الإعلان عنه في تعريف مكون النظام لاختبار مجرد، أي، منافذ TSI. وإذا لم يكن مكون نظام معرفاً بوضوح لاختبار مجرد، فإن القائمة تحتوى على جميع منافذ الاتصالات لمكون اختبار MTC. وترتب المنافذ كما تظهر في إعلان نمط مكون TTCN-3 المعني. وإذا لم توجد منافذ نظام، تعاد قائمة، أي قائمة طولها صفراً.	
Constraint	يستخدم فقط إذا كانت وحدة الجذر قد ضبطت قبلا.	
Effect	يوفر TE إلى الإدارة قائمة بمنافذ نظام لاختبار مجرد. وإذا لم توجد منافذ نظام، تعاد قائمة فارغة لمنفذ. وإذا لم يستطع TE أن يوفر قائمة، تعاد قيمة `null` المتميزة.	

tciStartTestCase 7.1.1.3.7

Signature	void tciStartTestCase(in TciTestCaseIdType testCaseId, in TciParameterListType parameterList)	
In Parameters	testCaseId	معرف اختبار مجرد كما عرف في وحدة TTCN-3.
	parameterList	قائمة Values حيث كل قيمة تعرف معلمة من قائمة المعلمات كما عرفت في تعريف اختبر مجرد TTCN-3. وترتب المعلمات في parameterList كما تظهر في توقيع TTCN-3 لاختبار مجرد. وإذا لم تمرر معلمات سواء قيمة null أو parameterList فارغة، أي، تمرر قائمة طولها صفر.
Return Value	void	
Constraint	يطلب فقط إذا تم اختيار وحدة قبلاً. وتمرر testCaseIds فقط لاختبارات مجردة تم الإعلان عنها في وحدة TTCN-3 المختارة حالياً. لا يمكن أن تبدأ الاختبارات المجردة المستوردة في وحدة مرجعية. وللبدء في اختبارات مجردة مستوردة، ينبغي اختيار الوحدة المرجعية (المستوردة) للاستخدام الأول لعملية tciRootModule.	
Effect	يبدأ tciStartTestCase اختبار مجرد في الوحدة المختارة حاليا مع معلمات معينة. ويصدر TE tciError إذا لم يوجد الاختبار المجرد هذا. إن جميع *inout* لمعلمات اختبار مجرد في parameterList تحتوي Value. إن جميع معلمات اختبار مجرد *out* في parameterList تحتوي على قيمة مميزة null نظرا لأنها تتعلق فقط عندما ينتهي اختبار مجرد.	

tciStopTestCase 8.1.1.3.7

Signature	void tciStopTestCase()
Return Value	void
Constraint	يطلب فقط إذا تم اختيار الوحدة قبلا.
Effect	يوقف tciStopTestCase اختبار مجرد يجرى تنفيذه حالياً. وإذا لم يكن TE ينفذ اختبار مجرد، يتم تجاهل العملية. وإذا كان جزء التحكم يجرى تنفيذه، يوقف tciStopTestCase تنفيذ اختبار مجرد منفذ حالياً، أي، تنفيذ اختبار مجرد تمت الدلالة عليه مؤخراً باستخدام عملية *provided* لـــ tciTestCaseStarted. ويتواصل تنفيذ ممكن لجزء التحكم كما لو كان الاختبار المجرد قد توقف عادة ويعاد مع حكم ERROR.

tciStartControl 9.1.1.3.7

Signature	TriComponentId tciStartControl()
Return Value	ينفذ TriComponentId الذي يمثل مكون اختبار لجزء تحكم الوحدة. وإذا لم يتمكن TE من بدء جزء التحكم لوحدة مختارة، تعاد القيمة المميزة null.
Constraint	يطلب فقط إذا تم اختيار الوحدة قبلاً.
Effect	يبدأ جزء التحكم لوحدة مختارة. ويبدأ جزء التحكم اختبارات مجردة TTCN-3 كما ورد في TTCN-3. وبينما ينفذ جزء التحكم، يطلب TE عملية *provided* لـــ tciTestCaseStarted وtciTestCaseTerminated لكل اختبار مجرد قد بدأ وانتهي. وبعد الانتهاء من جزء التحكم، يطلب TE عملية *provided* لـــ tciControlPartTerminated.

tciStopControl 10.1.1.3.7

Signature	void tciStopControl()
Return Value	void
Constraint	يطلب فقط إذا تم اختيار الوحدة قبلاً.
Effect	يوقف tciStopControl تنفيذ جزء التحكم. وإذا كان يجرى تنفيذ جزء التحكم حالياً، يتم تجاهل العملية. وإذا كان اختبار مجرد قد بدأ مباشرة يوقف هذا تنفيذ الاختبار المجرد كما لو كان tciStopTestCase قد طلب.

2.1.3.7 موفرة TCI-TM

يصف هذا القسم العمليات التي يتعين على TM أن توفرها إلى TE.

1.2.1.3.7 tciTestCaseStarted

Signature	`void tciTestCaseStarted(in TciTestCaseIdType testCaseId, in TciParameterListType parameterList, in TFloat timer)`	
In Parameters	`testCaseId`	معرف اختبار مجرد كما عرف في وحدة TTCN-3.
	`parameterList`	قائمة القيم التي هي جزء من توقيع اختبار مجرد. وترتب المعلمات في `parameterList` كما تظهر في إعلان اختبار مجرد TTCN-3.
	`timer`	قيمة مطلقة تمثل مدة مؤقت اختبار مجرد.
Return Value	`void`	
Constraint	يطلب فقط بعد ابتداء إما جزء التحكم للوحدة أو الاختبار المجرد باستخدام عملية *required* لـــ `tciStartControl` أو `tciStartTestCase`.	
Effect	يدل `tciTestCaseStarted` لـــ TM أن اختبار مجرد قد بدأ مع `testCaseId`. ولا يمكن تمييزه سواء بدأ الاختبار المجرد بوضوح باستخدام عملية *required*) لـــ `tciStartTestCase` أو ضمنيا بينما ينفذ جزء التحكم.	

2.2.1.3.7 tciTestCaseTerminated

Signature	`void tciTestCaseTerminated(in VerdictValue verdict, in TciParameterListType parameterList)`	
In Parameters	`verdict`	الحكم النهائي لاختبار مجرد.
	`parameterList`	قائمة قيم هي جزء من توقيع اختبار مجرد. وترتب المعلمات في قائمة معلمات كما تظهر في إعلان اختبار مجرد TTCN-3.
Return Value	`void`	
Constraint	يطلب فقط بعد ابتداء إما جزء التحكم للوحدة أو الاختبار المجرد باستخدام عملية *required* لـــ `tciStartControl` أو `tciStartTestCase`.	
Effect	يطلب TE هذه العملية ليدل إدارة الاختبار أن الاختبار المجرد الذي تم تنفيذه حاليا على MTC قد انتهي وأن الحكم النهائي كان *verdict*. وعند تنفيذ عملية `tciTestCaseTerminated`، تحتوي جميع معلمات اختبار *out* على *inout*. وتحتوي جميع معلمات اختبار مجرد على القيمة المميزة *Values* بسبب أنها ذات علاقة فقط ببدء اختبار مجرد وليست في الإجابة على النداء.	

3.2.1.3.7 tciControlTerminated

Signature	`void tciControlTerminated()`
Return Value	`void`
Constraint	يطلب فقط عندما يبدأ تنفيذ الوحدة باستخدام عملية `tciStartControl`.
Effect	يطلب TE هذه العملية ليدل إدارة الاختبار بأن جزء التحكم للوحدة المختارة قد أنهي التنفيذ.

4.2.1.3.7 tciGetModulePar

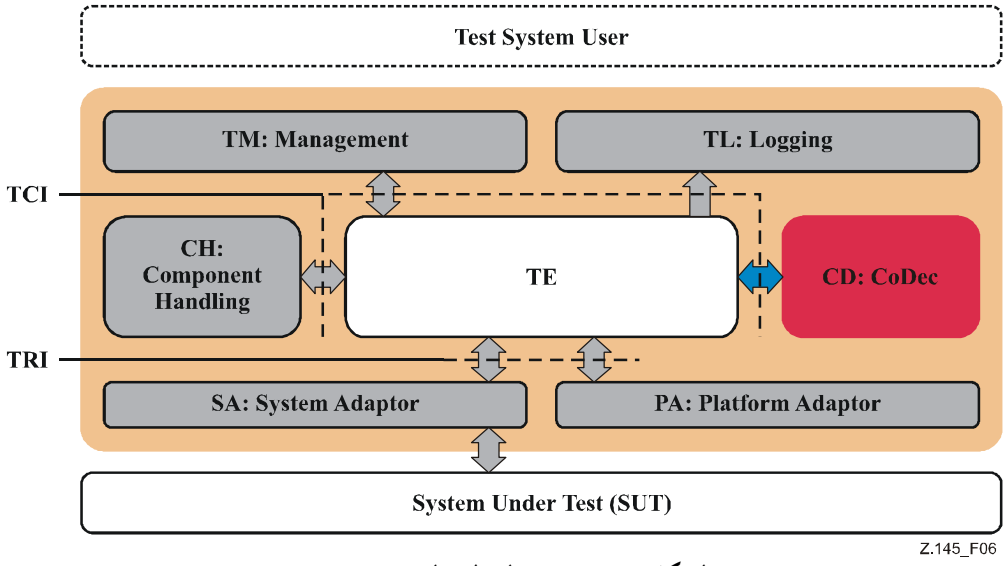
Signature	`Value tciGetModulePar(in TciModuleParameterIdType parameterId)`	
In Parameters	`parameterId`	معرف معلمة الوحدة كما عرف في TTCN-3.
Return Value	`A value.`	
Constraint	تطلب هذه العملية عندما يحتاج TE إلى نفاذ قيمة معلمة الوحدة. وتتم استبانة كل معلمة وحدة نافذة مرة واحدة فقط بين زوجي `tciStartTestCase` و`tciTestCaseTerminated` إذا بدأ اختبار المجرد بوضوح أو بين زوجي `tciStartControl` و`tciControlTerminated`إذا كان جزء التحكم لوحدة قد بدأ.	
Effect	توفر الإدارة إلى TE `Value` للدلالة على `parameterId`. يعيد كل نداء `()tciGetModulePar` نفس القيمة طوال تنفيذ اختبار مجرد بدأ بوضوح أو طوال تنفيذ جزء التحكم. وإذا لم تتمكن الإدارة من توفير قيمة TTCN-3، تعاد قيمة `null` المميزة.	

tciError **5.2.1.3.7**

Signature	`void tciError(in TString message)`	
In Parameters	`message`	قيمة سلسلة، أي، خطأ الرسالة.
Return Value	`void`	
Constraint	يمكن أن يطلبه TE في أي وقت ليدل على حالة خطأ لا يمكن استرجاعه. وتكون حالة الخطأ هذه إما بواسطة CH أو CD الدالة عليه أو يمكن أن تحدث داخل TE.	
Effect	يدل TE على حدوث حالة خطأ لا يمكن استرجاعه. وتحتوي `message` على عبارة السبب الذي قد أرسل إلى مستعمل نظام الاختبار. ويرجع الأمر إلى إدارة الاختبار تحديد تنفيذ اختبارات مجردة أو أجزاء تحكم إذا كانت تمرر. ويتعين على إدارة الاختبار أن تتخذ تدابير واضحة لإنهاء تنفيذ الاختبار فوراً.	

2.3.7 السطح البيني TCI-CD

يصف السطح البيني لكودك TCI عمليات TTCN-3 قابل للتنفيذ المطلوبة للتنفيذ وتوفير عمليات تنفيذ كودك لمخطط تشفير معين إلى TE (الشكل 6).



الشكل Z.145/6 – السطح البيني TCI-CD

يشفر تنفيذ كودك قيم TTCN-3 طبقاً لنعت تشفير في سلسلة ثنائية ويفكك تشفير سلسلة ثنائية طبقاً لافتراض تفكيك تشفير. وللتمكن من تفكيك تشفير سلسلة ثنائية إلى قيمة TTCN-3، يطلب CD بعض الوظائف من TE. وبدوره، يوفر CD وظائف تشفير وفك تشفير إلى TTCN-3 قابل للتنفيذ.

يوضح القسم 10 الاستخدام والترتيب التتابعي لنداءات العمليات بواسطة إما TE أو CD.

1.2.3.7 TCI-CD مطلوب

يصف هذا القسم العمليات التي يطلبها CD من TE. وجميع العمليات الواردة في هذا القسم مطلوبة أيضاً عند السطوح البينية لـ TCI-TM وTCI-CH.

getTypeForName 1.1.2.3.7

Signature	`Type getTypeForName(in TString typeName)`	
In Parameters	`typeName`	يعرف اسم TTCN-3 نمط كما عرف في وحدة TTCN-3. وما يلي هي أسماء أنماط محجوزة تعيد نمط معرف مسبقا: "integer" "float" "bitstring" "hexstring" "octetstring" "charstring" "universal charstring" "boolean" "verdicttype" "objid" typeName has to be the fully qualified type name, i.e., module.typeName
Return Value	نمط يمثل نمط TTCN-3 المطلوب.	
Constraint	---	
Effect	يعيد نمط يمثل نمط TTCN-3. يمكن استرداد أنماط TTCN-3 المعرفة مسبقا من TE باستخدام كلمات افتتاحية TTCN-3 لأنماط معرفة مسبقاً. وفي هذه الحالة يدل `typeName` على نمط TTCN-3 الأساسي مثل "`charstring`". يعيد القيمة المميزة `null` إذا لم يكن من الممكن إعادة النمط المطلوب. ولاحظ أن `anytype` لا يمكن الحصول عليهما مع وحدة مضبوطة على `address`. وبالرغم من أنهما نمطين معرفين مسبقاً، يكونان مميزان بين الوحدات. فمثلاً، يمكن أن يكون `null` إما نمطاً معرفاً مسبقاً أو نمطاً معرفاً لمستعمل في وحدة. ولا يمكن إعادة تعريف أنماط معرفة مسبقاً أخري.	

getInteger 2.1.2.3.7

`Type getInteger()`	**Signature**
An instance of `Type` representing a TTCN-3 integer type.	**Return Value**
Constructs and returns a basic TTCN-3 integer type.	**Effect**

getFloat 3.1.2.3.7

`Type getFloat()`	**Signature**
An instance of `Type` representing a TTCN-3 float type.	**Return Value**
Constructs and returns a basic TTCN-3 float type.	**Effect**

getBoolean 4.1.2.3.7

`Type getBoolean()`	**Signature**
An instance of `Type` representing a TTCN-3 boolean type.	**Return Value**
Constructs and returns a basic TTCN-3 boolean type.	**Effect**

getObjid 5.1.2.3.7

`Type getObjid()`	**Signature**
An instance of `Type` representing a TTCN-3 object id type.	**Return Value**
Constructs and returns a basic TTCN-3 object id type.	**Effect**

getCharstring 6.1.2.3.7

`Type getCharstring()`	**Signature**
An instance of `Type` representing a TTCN-3 charstring type.	**Return Value**
Constructs and returns a basic TTCN-3 charstringtype.	**Effect**

getUniversalCharstring 7.1.2.3.7

`Type getUniversalCharstring()`	**Signature**
An instance of `Type` representing a TTCN-3 universal charstring type.	**Return Value**
Constructs and returns a basic TTCN-3 universal charstring type.	**Effect**

getHexstring 8.1.2.3.7

`Type getHexstring()`	**Signature**
An instance of `Type` representing a TTCN-3 hexstring type.	**Return Value**
Constructs and returns a basic TTCN-3 hexstring type.	**Effect**

getBitstring 9.1.2.3.7

`Type getBitstring()`	**Signature**
An instance of `Type` representing a TTCN-3 bitstring type.	**Return Value**
Constructs and returns a basic TTCN-3 bitstring type.	**Effect**

getOctetstring 10.1.2.3.7

`Type getOctetstring()`	**Signature**
An instance of `Type` representing a TTCN-3 octetstring type.	**Return Value**
Constructs and returns a basic TTCN-3 octetstring type.	**Effect**

getVerdict 11.1.2.3.7

`Type getVerdict()`	**Signature**
An instance of `Type` representing a TTCN-3 verdict type.	**Return Value**
Constructs and returns a basic TTCN-3 verdict type.	**Effect**

tciErrorReq 12.1.2.3.7

`void tciErrorReq(in TString message)`		**Signature**
A string value, i.e., the error phrase describing the problem.	`message`	**In Parameters**
`void`		**Return Value**
يطلب عندما تحدث حالة خطأ.		**Constraint**
يبلغ TE عن حالة خطأ لا يمكن استرداده في CD ويحيل دلالة الخطأ إلى إدارة الاختبار.		**Effect**

2.2.3.7 موفر TCI-CD

يصف هذا القسم العمليات التي توفرها TM إلى TE.

1.2.2.3.7 فك تشفير

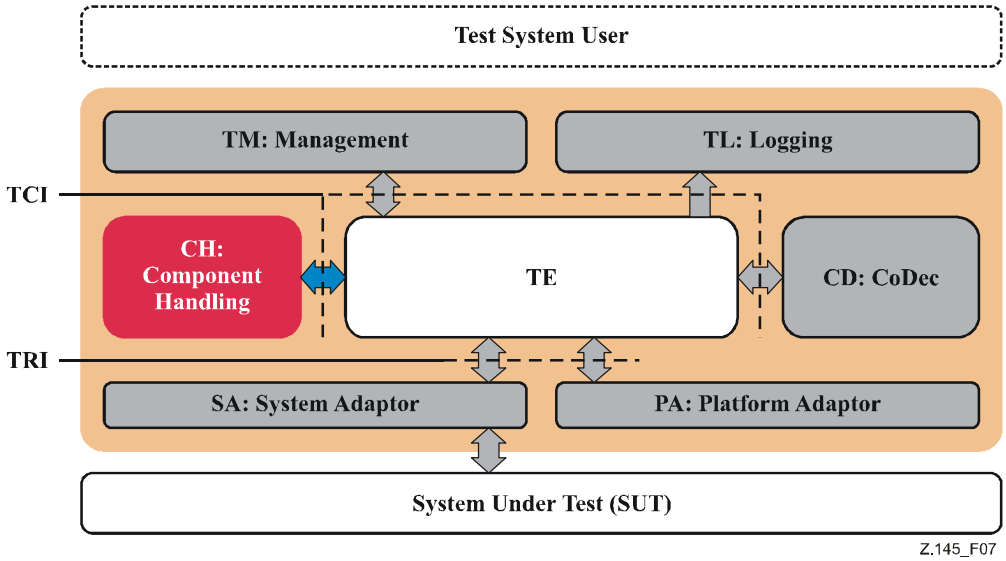
Signature	`Value decode(in TriMessageType message, in Type decodingHypothesis)`	
In Parameters	`message`	الرسالة المشفرة التي يفكك تشفيرها.
	`decodingHypothesis`	الافتراض الذي يمكن أن يقوم عليه فك تشفير.
Return Value	يعيد القيمة المفكك تشفيرها، إذا كانت القيمة هي نمط متوائم مثل `decodingHypothesis`، وإلا القيمة المميزة هي `null`.	
Constraint	تطلب هذه العملية عندما يتعين على TE فك تشفير قيمة مشفرة. وقد يفكك TE التشفير فوراً بعد إستقبال القيمة المشفرة، أو قد يؤجل، لاعتبارات الأداء، فك التشفير حتى النفاذ الفعلي للقيمة المشفرة.	
Effect	تقوم هذه العملية بفك تشفير `message` طبقاً لقواعد التشفير وتعيد قيمة TTCN-3. ويستخدم `decodingHypothesis` لتحديد ما إذا كانت القيمة المشفرة يمكن تفكيك تشفيرها. وإذا لم تكن قاعدة التشفير مكتفية ذاتياً، أي، إذا لم تكن الرسالة المشفرة تحتوى ضمنياً نمطها `decodingHypothesis` تستخدم. وإذا كان من الممكن فك تشفير القيمة المشفرة دون فك شفرة الافتراض، تعاد القيمة `null` المميزة إذا حدد النمط من الرسالة المشفرة أنه لا يتواءم مع افتراض فك التشفير.	

2.2.2.3.7 تشفير

Signature	`TriMessageType encode(in Value value)`	
In Parameters	`value`	القيمة التي تشفر.
Return Value	يعيد `TriMessage` المشفرة لقاعدة التشفير المحددة.	
Constraint	تطلب هذه العملية عندما يتعين على TE أن يشفر قيمة.	
Effect	يعيد `TriMessage` مشفرة طبقاً لقواعد التشفير.	

3.3.7 السطح البيني TCI-CH

يصف السطح البيني لمناولة مكون TCI العمليات التي تطلب من TTCN-3 قابل للتنفيذ أن ينفذها وعمليات تنفيذ مناولة مكون لتوفر إلى TE (الشكل 7).



الشكل Z.145/7 – السطح البيني TCI-CH

يوزع تنفيذ مناولة مكون عمليات تشكيل TTCN-3 مثل `create` و`connect` و`start` والاتصالات بين مكونات مثل send على منفذ موصول فيما بين واحد أو أكثر من TTCN-3 قابل للتنفيذ مشارك في دورة اختبار. ولاحظ أنه بالرغم من أمثلة متعددة قد يشارك فيها TE في دورة اختبار، إلا أن هذا ليس إلزامياً.

إن المبدأ الأساسي هو أن TCI-CH ليس *أداة* من أي نوع لوظيفية TTCN-3. وبدلاً من ذلك، يبلغ TE بأنه سينشئ مثلاً مكون اختبار. وعلى أساس المعرفة الداخلية لـــ CH، يرسل طلب إنشاء مكون اختبار إلى TE مشارك آخر (بعيد). وينشئ TE المشارك (البعيد) مكون TTCN-3 ويوفر مناولة إلى TE الطالب (المحلي). ويمكن الآن لـــ TE الطالب (المحلي) العمل على مكون الاختبار الذي أنشأ عبر مناولة المكون هذا.

وفي تعريف العملية، يستخدم المصطلحان "TE محلي" و"TE بعيد" للتأكيد على حقيقة أن تنفيذ نظام اختبار قد يوزع على أجهزة اختبار عديدة، يحتوي كل واحد منها على TE مكتمل. ويشير المصطلحان "محلي" و"بعيد" إلى السطوح البينية التي يجرى وصفها حالياً. ومن أجل التيسير، يشير المصطلح "محلي" دائما إلى TE الذي يكون إما مطلوباً لعملية (لعمليات *مطلوبة*) أو طالباً لعملية (لعمليات *موفرة*). وبينما يعتبر TE مفهومياً على أنه موزع، تعتبر CH على أنها غير موزعة. ويمكن تحقيق هذا إما باستخدام معمارية مركزية أو باستخدام منصة برمجيات متوسطة تستوعب من جوانب توزيع. وبالرغم من أن TE يمكن توزيعه عبر أجهزة مادية مختلفة، قد توجد تشكيلات حيث واحد منها فقط، TE غير موزع يشارك في دورة اختبار. وفي هذه الحالة، يشير المصطلحان "محلي" و"بعيد" إلى نفس TE مطابق.

يوضح القسم 10 الاستخدام والترتيب التتابعي لنداءات العمليات بواسطة إما TE أو CH.

وبالرغم من أن جميع TTCN-3 القابلة للتنفيذ في دورة اختبار متساوية، هناك TE* مميز. وTE* هذا هو TE حيث `tciStartTestCase()` أو `tciStartControl()` قد تمت معالجة بوضوح. والسبب في هذا التمييز هو أن TE* يحسب الحكم الشامل. ويبلغ TE* إدارة الاختبار عن إنتهاء تنفيذ الاختبار ثم يوفر حكماً شاملاً للاختبار المجرد.

1.3.3.7 TCI-CH مطلوبة

يصف هذا القسم العمليات التي تطلبها CH من TE. وبالإضافة إلى العمليات المحددة في هذا القسم، تطلب جميع العمليات *المطلوبة* للسطح البيني TCI-CD.

1.1.3.3.7 tciEnqueueMsgConnected

Signature	`void tciEnqueueMsgConnected(in TriPortIdType sender,`
	`in TriComponentIdType receiver,`
	`in Value rcvdMessage)`	

In Parameters	`sender`	Port identifier at the sending component via which the message is sent.
	`receiver`	Identifier of the receiving component.
	`rcvdMessage`	The value to be enqueued.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciSendConnected` عند TE البعيد.	
Effect	يضع TE في صف انتظار القيمة المستقبلة في صف انتظار منفذ محلي للمكون الدال عليه للمستقبل.	

2.1.3.3.7 tciEnqueueCallConnected

Signature	`void tciEnqueueCallConnected(in TriPortIdType sender,`
	`in TriComponentIdType receiver,`
	`in TriSignatureIdType signature,`
	`in TciParameterListType parameterList)`	

In Parameters	`sender`	Port identifier at the sending component via which the message is sent.
	`receiver`	Identifier of the receiving component.
	`signature`	Identifier of the signature of the procedure call.
	`parameterList`	قائمة معلمات القيمة التي هي جزء من التوقيع الدال عليه. وترتب المعلمات في `parameterList` كما تظهر في إعلان توقيع TTCN-3.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciCallConnected` عند TE البعيد. تحتوي معلمات إجراء جميع على قيم *out* و*inout*. وتحتوي معلمات إجراء جميع *out* على قيمة مميزة `null` بسبب أنها ذات علاقة فقط في إجابة على نداء إجراء ولكن ليست في نداء الإجراء نفسه. ومعلمات الإجراء هي معلمات محددة في مقاس توقيع TTCN-3.	
Effect	يضع TE في صف انتظار القيمة المستقبلة في صف انتظار منفذ محلي للمكون الدال عليه للمستقبل.	

tciEnqueueReplyConnected 3.1.3.3.7

Signature	`void tciEnqueueReplyConnected(in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue)`	
In Parameters	`sender`	Identifier of the port sending the reply.
	`receiver`	Identifier of the component receiving the reply.
	`signature`	Identifier of the signature of the procedure call.
	`parameterList`	قائمة معلمات القيمة التي هي جزء من التوقيع الدال عليه. وترتب المعلمات في parameterList كما تظهر في إعلان توقيع TTCN-3.
	`returnValue`	يعيد (خيارياً) قيمة نداء الإجراء.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciReplyConnected` عند TE البعيد. تحتوي معلمات إجراء جميع *out* و*inout* والقيمة العائدة على قيم. وتحتوي معلمات إجراء جميع *in* على قيمة مميزة `null` بسبب أنها ذات علاقة فقط بنداء إجراء ولكن ليست في الإجابة على نداء. وتحتوي `parameterList` على معلمات نداء إجراء. وهذه المعلمات هي معلمات محددة في مقاس توقيع TTCN-3. وإذا لم يعرف نمط إعادة لتوقيع الإجراء في TTCN-3 ATS ، تمرر القيمة المميزة `null` لـــ `returnValue`.	
Effect	يضع TE الإجابة في صف انتظار عند صف انتظار منفذ محلي للمكون الدال عليه للمستقبل.	

tciEnqueueRaiseConnected 4.1.3.3.7

Signature	`void tciEnqueueRaiseConnected(in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in Value exception)`	
In Parameters	`sender`	Identifier of the port sending the reply.
	`receiver`	Identifier of the component receiving the reply.
	`signature`	Identifier of the signature of the procedure call.
	`exception`	The exception.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciRaiseConnected` عند TE بعيد.	
Effect	يضع TE في صف انتظار استثناء عند صف انتظار منفذ محلي لمكون دال عليه لمستقبل.	

tciCreateTestComponent 5.1.3.3.7

Signature	`TriComponentIdType tciCreateTestComponent(in TciTestComponentKindType kind, in Type componentType), in TString name)`	
In Parameters	`kind`	نوع المكون الذي ينشأ، سواء `MTC, PTC or CONTROL`.
	`componentType`	معرف نمط مكون TTCN-3 الذي ينشأ.
	`name`	اسم المكون الذي ينشأ.
Return Value	قيمة `TriComponentIdType` للمكون المنشأ.	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* *tciCreateTestComponentReq* عند TE البعيد. ويضبط `componentType` على القيمة المميزة `null` إذا أنشأ مكون اختبار من نوع `control`. ويضبط `name` على القيمة المميزة `null` إذا لم يعطي اسم في بيان إنشاء TTCN-3.	
Effect	ينشئ TE مكون اختبار TTCN-3 لـــ `componentType` ويمرر مرجع `TriComponentIdType` إلى CH. وترسل CH المرجع إلى TE البعيد.	

tciStartTestComponent 6.1.3.3.7

Signature	`void tciStartTestComponent(in TriComponentIdType component, in TciBehaviourIdType behaviour, in TciParamaterListType parameterList)`	
In Parameters	`component`	معرف المكون الذي يبدأ. يشير إلى معرف أنشئ في السابق بواسطة نداء `.tciCreateTestComponent`
	`behaviour`	معرف سلوك يبدأ على مكون.
	`parameterList`	قائمة قيم حيث كل قيمة تعرف معلمة من قائمة معلمات كما تعرف في إعلان وظيفة لوظيفة يجرى بدأها. وترتب معلمات `parameterList` كما تظهر في توقيع TTCN-3 لاختبار مجرد. وإذا لم تمرر معلمات، سواء القيمة `null` أو `parameterList` فارغة، أي، تمرر قائمة طولها صفر.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciStartTestComponentReq` عند TE بعيد. ونظرا لأن معلمات *in* يسمح لها فقط لوظائف يجرى قد بدأت [2] ITU-T Z.140، تحتوي `parameterList` على معلمات *in* فقط.	
Effect	يبدأ TE السلوك المشار إليه على مكون دال عليه.	

tciStopTestComponent 7.1.3.3.7

Signature	`void tciStopTestComponent(in TriComponentIdType component)`	
In Parameters	`component`	معرف مكون يتعين وقفه.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciStopTestComponentReq` عند TE بعيد.	
Effect	يوقف TE السلوك المشار إليه على مكون دال عليه.	

tciConnect 8.1.3.3.7

Signature	`void tciConnect(in TriPortIdType fromPort, in TriPortIdType toPort)`	
In Parameters	`fromPort`	Identifier of the test component port to be connected from.
	`toPort`	Identifier of the test component port to be connected to.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciConnect` عند TE بعيد.	
Effect	يوصل TE المنافذ الدالة كل بالآخر.	

tciDisconnect 9.1.3.3.7

Signature	`void tciDisconnect(in TriPortIdType fromPort, in TriPortIdType toPort)`	
In Parameters	`fromPort`	Identifier of the test component port to be disconnected.
	`toPort`	Identifier of the test component port to be disconnected.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciDisconnect` عند TE بعيد.	
Effect	يفكك TE توصيل المنافذ الدالة عليها.	

tciMap 10.1.3.3.7

Signature	`void tciMap(in TriPortIdType fromPort, in TriPortIdType toPort)`	
In Parameters	`fromPort`	Identifier of the test component port to be mapped from.
	`toPort`	Identifier of the test component port to be mapped to.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciMapReq` عند TE بعيد.	
Effect	يتقابل TE مع المنافذ الدالة عليه الواحد بالآخر.	

tciUnmap 11.1.3.3.7

Signature	`void tciUnmap(in TriPortIdType fromPort, in TriPortIdType toPort)`	
In Parameters	`fromPort`	Identifier of the test component port to be unmapped.
	`toPort`	Identifier of the test component port to be unmapped.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciUnmapReq` عند TE بعيد.	
Effect	يفكك TE تقابل المنافذ الدالة عليه.	

tciTestComponentTerminated 12.1.3.3.7

Signature	`void tciTestComponentTerminated(in TriComponentIdType component, in VerdictValue verdict)`	
In Parameters	`component`	Identifier of the component that has terminated.
	`verdict`	Verdict after termination of the component.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciTestComponentTerminatedReq` عند TE بعيد.	
Effect	يبلغ TE المحلي بإنتهاء مكون الاختبار الدال عليه على TE بعيد. وطالما يجرى تنفيذ وظيفة على مكون اختبار يمكن أن يكون لها معلمات *in* ITU-T Rec. Z.140 [2]، لا يكون لعملية `tciTestComponentTerminated` معلمة `parameterList`.	

tciTestComponentRunning 13.1.3.3.7

Signature	`TBoolean tciTestComponentRunning(in TriComponentIdType component)`	
In Parameters	`component`	معرف المكون الذي يتم التأكد منه للتمرير.
Return Value	`true` إذا كان المكون الدال عليه مازال ينفذ سلوك، وإلا `false`.	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciTestComponentRunningReq` عند TE بعيد.	
Effect	يحدد TE المحلي ما إذا كان المكون الدال عليه ينفذ سلوك اختبار. وإذا كان المكون ينفذ سلوك، يعاد `true`. وفي أي حالة أخرى، مثلاً، أنهي مكون اختبار تنفيذ أو أن مكون الاختبار لم يبدأ وما إلى ذلك، يعاد `false`. وبعدما تعود العملية، ترسل CH القيمة إلى TE البعيد.	

tciTestComponentDone 14.1.3.3.7

Signature	`TBoolean tciTestComponentDone(in TriComponentIdType comp)`	
In Parameters	`comp`	معرف المكون الذي يتم التأكد من للتنفيذ.
Return Value	`true` إذا كان المكون الدال عليه قد أكمل تنفيذ سلوكه، وإلا `false`.	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciTestComponentDoneReq` عند TE البعيد.	
Effect	يحدد TE المحلي ما إذا كان المكون الدال عليه ينفذ سلوك اختباره. وإذا كان المكون قد أكمل سلوكه، يعاد `true`. وفي أي حالة أخرى، مثلاً، لم يبدأ مكون الاختبار التنفيذ أو مازال مكون الاختبار يقوم بالتنفيذ وما إلى ذلك، يعاد `false`. وبعدما تعود العملية، ترسل CH القيمة إلى TE البعيد.	

tciGetMTC 15.1.3.3.7

Signature	`TriComponentIdType tciGetMTC()`	
Return Value	قيمة `TriComponentIdType` لـــ MTC إذا كان MTC ينفذ على TE المحلي، وإلا تكون القيمة المميزة `null`.	
Constraint	تطلب CH هذه العملية عند TE محلي ملائم عندما يطلب *provided* `tciGetMTCReq` عند TE بعيد.	
Effect	يحدد TE المحلي ما إذا كان MTC ينفذ على MTC محلي. وإذا كان MTC ينفذ على TE محلي، تعاد هوية مكون MTC وإذا لم ينفذ MTC على TE محلي ، تعاد القيمة المميزة `null`. ولن يكون للعملية أي تأثير على تنفيذ MTC . وبعدما تعود العملية، ترسل CH القيمة إلى TE البعيد.	

tciExecuteTestCase 16.1.3.3.7

Signature	`void tciExecuteTestCase(in TciTestCaseIdType testCaseId, in TriPortIdListType tsiPortList)`	
In Parameters	`testCaseId`	معرف اختبار مجرد كما عرف في وحدة TTCN-3.
	`tsiPortList`	يحتوي على جميع منافذ التي تم الإعلان عن أنها في تعريف مكون النظام لاختبار مجرد، أي، منافذ TSI. وإذا لم يعرف مكون النظام بوضوح لاختبار مجرد، فإن `tsiPortList` تحتوي على جميع منافذ الاتصالات لـــ MTC. ويجرى ترتيب المنافذ في `tsiPortList` كما تظهر في إعلان نمط مكون TTCN-3 المعني. وإذا لم تمرر منافذ سواء القيمة `null` أو `tsiPortList` فارغة، أي، تمرر قائمة طولها صفراً.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي الملائم عندما يطلب *provided* `tciExecuteTestCaseReq` عند TE بعيد.	
Effect	يحدد TE محلي ما إذا كانت التوصيلات السكونية إلى SUT وتدميث وسائل الاتصالات لمنافذ TSI ينبغي القيام بها.	

tciReset 17.1.3.3.7

Signature	`void tciReset()`
Return Value	`void`
Constraint	تطلب CH هذه العملية عند TE المحلي الملائم عندما يطلب *provided* `tciResetReq` عند TE بعيد.
Effect	يمكن أن يقرر TE اتخاذ أي وسائل لإعادة تدميث نظام الاختبار محلياً.

tciKillTestComponent 18.1.3.3.7

Signature	`void tciKillTestComponent(in TriComponentIdType comp)`	
In Parameters	`comp`	معرف المكون الذي يتعين كبته.
Return Value	`void`	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciKillTestComponentReq` عند TE بعيد.	
Effect	يوقف TE السلوك على المكون الدال عليه إذا لزم الأمر وينقله إلى حالة كبت.	

tciTestComponentAlive 19.1.3.3.7

Signature	`TBoolean tciTestComponentAlive(in TriComponentIdType comp)`	
In Parameters	`comp`	معرف المكون للتأكد من أنه نشط.
Return Value	`true` إذا كان المكون الدال عليه نشط، وإلا `false`.	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciTestComponentAliveReq` عند TE بعيد.	
Effect	يحدد TE المحلي ما إذا كان المكون الدال عليه نشطاً. وبعد إعادة العملية، ترسل CH القيمة إلى TE البعيد.	

tciTestComponentKilled 20.1.3.3.7

Signature	`TBoolean tciTestComponentKilled(in TriComponentIdType comp)`	
In Parameters	`comp`	معرف المكون للتأكد منه مكبوت.
Return Value	`true` إذا كان المكون الدال عليه مكبوتاً، وإلا `false`.	
Constraint	تطلب CH هذه العملية عند TE المحلي عندما يطلب *provided* `tciTestComponentKilledReq` عند TE بعيد.	
Effect	يحدد TE المحلي ما إذا كان المكون الدال عليه في حالة كبت. وإذا كان، يعاد `true`. وفي أي حالة أخرى، يعاد `false`. وبعد إعادة العملية، ترسل CH القيمة إلى TE البعيد.	

TCI-CH موفرة 2.3.3.7

يصف هذا القسم العمليات التي توفرها CH إلى TE.

tciSendConnected 1.2.3.3.7

Signature	`void tciSendConnected(in TriPortIdType sender,`
	`in TriComponentIdType receiver,`
	`in Value sendMessage)`	

In Parameters	`sender`	معرف منفذ عند المكون المرسل الذي ترسل عبره الرسالة.
	`receiver`	معرف مكون مستقبل.
	`sendMessage`	الرسالة التي ترسل.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية ترسل أحادياً على منفذ مكون TTCN-3، الذي تم توصيله بمنفذ مكون آخر.	
Effect	يرسل إرسالاً لا تزامنياً إلى مكون مستقبل معين فقط. وترسل CH الرسالة إلى TE بعيد ينفذ مستقبل عليه ويضع المعطيات في صف انتظار في TE البعيد.	

tciSendConnectedBC 2.2.3.3.7

Signature	`void tciSendConnectedBC(in TriPortIdType sender,` `in Value sendMessage)`	
In Parameters	`sender`	معرف منفذ عند المكون المرسل الذي ترسل عبره الرسالة.
	`sendMessage`	الرسالة التي ترسل.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية ترسل إذاعياً على منفذ مكون TTCN-3، الذي تم توصيله بمنفذ مكون آخر.	
Effect	يرسل إرسالاً لا تزامنياً إلى جميع المكونات التي تنفذ على هذا المنفذ. وترسل CH الرسالة إلى جميع TEs البعيدين حيث يجرى تنفيذ المستقبلات عليه ويضع المعطيات في صف الانتظار في TE البعيد.	

tciSendConnectedMC 3.2.3.3.7

Signature	`void tciSendConnectedMC(in TriPortIdType sender,`
	`in TriComponentIdListType receivers,`
	`in Value sendMessage)`	

In Parameters	`sender`	معرف منفذ عند المكون المرسل الذي ترسل عبره الرسالة.
	`receivers`	معرفات المكونات المستقبلة.
	`sendMessage`	الرسالة التي ترسل.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية ترسل إرسالاً متعدداً على منفذ مكون TTCN-3، الذي تم توصيله بمنفذ مكون آخر.	
Effect	يرسل إرسالاً لا تزامنياً إلى جميع مكونات مستقبل. وترسل CH الرسالة إلى جميع TEs البعيدة تنفذ مستقبلات عليها ويضع المعطيات في صف انتظار في TEs البعيدة.	

tciCallConnected **4.2.3.3.7**

Signature	`void tciCallConnected(in TriPortIdType sender,`
	`in TriComponentIdType receiver,`
	`in TriSignatureIdType signature,`
	`in TciParameterListType parameterList)`	

In Parameters	`sender`	معرف منفذ عند المكون المرسل الذي ترسل عبره الرسالة.
	`receiver`	معرف مكون مستقبل.
	`signature`	معرف توقيع نداء إجراء.
	`parameterList`	قائمة معلمات قيمة هي جزء من توقيع دال عليه. وترتب المعطيات في `parameterList` كما تظهر في إعلان توقيع TTCN-3.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية نداء أحادية على منفذ مكون TTCN-3، تم توصيله بمنفذ مكون آخر. وتحتوي جميع معلمات إجراء *in* و*inout* على قيم. وتحتوي جميع معلمات إجراء *out* على قيمة مميزة `null` بسبب أنها تتعلق فقط بالإجابة على نداء إجراء ولكن ليس في نداء الإجراء نفسه. ومعلمات الإجراء هي المعلمات المحددة في مقاس توقيع TTCN-3.	
Effect	عند تنفيذ هذه العملية، يمكن أن يمهد TE نداء الإجراء المتوافق مع معرف التوقيع `signature` عند مستقبل المكون المطلوب. وتعيد عملية `tciCallConnected` دون انتظار لعودة نداء إجراء صادر. ولاحظ أن قيمة إمهال خيارية، التي يمكن تحديدها في TTCN-3 ATS لعملية نداء، لا تتضمن في توقيع عملية `tciCallConnected`. ويكون TE مسؤولا عن تناول هذه المسألة بواسطة بدأ مؤقت عملية نداء TTCN-3 في PA مع نداء عملية TRI منفصل، أي، `triStartTimer`. وترسل CH النداء إلى TE بعيد على `receiver` ينفذ ووضع النداء في صف انتظار في TE بعيد.	

tciCallConnectedBC **5.2.3.3.7**

Signature	`void tciCallConnectedBC(in TriPortIdType sender,`
	`in TriSignatureIdType signature,`
	`in TciParameterListType parameterList)`	

In Parameters	`sender`	معرف منفذ عند المكون المرسل الذي ترسل عبره الرسالة.
	`signature`	معرف توقيع نداء إجراء.
	`parameterList`	قائمة معلمات قيمة هي جزء من توقيع دال عليه. وترتب المعطيات في `parameterList` كما تظهر في إعلان توقيع TTCN-3.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية نداء إذاعية على منفذ مكون TTCN-3، تم توصيله بمنفذ مكون آخر. وتحتوي جميع معلمات إجراء *in* و*inout* على قيم. وتحتوي جميع معلمات إجراء *out* على قيمة مميزة `null` بسبب أنها تتعلق فقط بالإجابة على نداء إجراء ولكن ليس نداء الإجراء نفسه. ومعلمات الإجراء هي المعلمات المحددة في مقاس توقيع TTCN-3.	
Effect	عند تنفيذ هذه العملية، يمكن أن يمهد TE نداء الإجراء المتوافق مع معرف التوقيع `signature` عند مستقبل المكون المطلوب. ويعيد عملية `tciCallConnected` دون انتظار لعودة نداء إجراء صادر. ولاحظ أن قيمة إمهال خيارية، التي يمكن تحديدها في TTCN-3 ATS لعملية نداء، لا تتضمن في توقيع عملية `tciCallConnected`. ويكون TE مسؤولاً عن تناول هذه المسألة بواسطة بدء مؤقت عملية نداء TTCN-3 في PA مع نداء عملية TRI منفصل، أي، `triStartTimer`. وترسل CH النداء إلى TE بعيد على `receiver` ينفذ ووضع النداء في صف انتظار في TE بعيد.	

tciCallConnectedMC 6.2.3.3.7

Signature	`void tciCallConnectedMC(in TriPortIdType sender,`
	`in TriComponentIdListType receivers,`
	`in TriSignatureIdType signature,`
	`in TciParameterListType parameterList)`	

In Parameters	`sender`	معرف منفذ عند المكون المرسل الذي ترسل عبره الرسالة.
	`receivers`	معرف مكونات مستقبلة.
	`signature`	معرف توقيع نداء إجراء.
	`parameterList`	قائمة معلمات قيمة هي جزء من توقيع دال عليه. وترتب المعطيات في parameterList كما تظهر في إعلان توقيع TTCN-3
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية نداء متعددة على منفذ مكون TTCN-3، الذي تم توصيله بمنافذ مكون آخر. وتحتوي جميع معلمات إجراء *in* و*inout* على قيم. وتحتوي معلمات جميع *out* على قيمة مميزة `null` بسبب أنها تتعلق فقط بالإجابة على نداء إجراء ولكن ليس في نداء الإجراء نفسه. ومعلمات الإجراء هي المعلمات المحددة في مقاس توقيع TTCN-3.	
Effect	عند تنفيذ هذه العملية، يمكن أن يمهد TE نداء الإجراء المتوافق مع معرف التوقيع `signature` عند مستقبل المكون المطلوب. وتعاد عملية `tciCallConnected` دون انتظار لعودة نداء إجراء صادر. ولاحظ أن قيمة إمهال خيارية، التي يمكن تحديدها في TTCN-3 ATS لعملية نداء، لا تتضمن في توقيع عملية `tciCallConnected`. ويكون TE مسؤولاً عن تناول هذه المسألة بواسطة بدء مؤقت عملية نداء TTCN-3 في PA مع نداء عملية TRI منفصل، أي، `triStartTimer`. وترسل CH النداء إلى جميع TEs البعيدة على `receiver` الذي ينفذ ووضع النداء في صف انتظار TEs البعيدة.	

tciReplyConnected 7.2.3.3.7

Signature	`void tciReplyConnected(in TriPortIdType sender,`
	`in TriComponentIdType receiver,`
	`in TriSignatureIdType signature,`
	`in TciParameterListType parameterList,`
	`in Value returnValue)`	

In Parameters	`sender`	معرف المنفذ المرسل للإجابة.
	`receiver`	معرف المكون المستقبل للإجابة.
	`signature`	معرف توقيع نداء الإجراء.
	`parameterList`	قائمة معلمات مشفرة هي جزء من التوقيع الدال عليه. وترتب المعطيات في `parameterList` كما تظهر في إعلان توقيع TTCN-3.
	`returnValue`	إعادة (خيارية) لقيمة نداء الإجراء.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية إجابة أحادية على منفذ مكون TTCN-3، الذي تم توصيله بمنفذ مكون آخر. وتحتوي جميع معلمات إجراء *in* و*inout* والقيمة العائدة على قيم. وتحتوي معلمات جميع *in* على قيمة مميزة `null` بسبب أنها تتعلق بنداء إجراء ولكن ليس في الإجابة على النداء. وتحتوي `parameterList` على معلمات نداء إجراء. وهذه المعلمات هي المعلمات المحددة في مقاس توقيع TTCN-3. وإذا لم يعرف نمط إعادة لتوقيع الإجراء في TTCN-3، تمرر القيمة المميزة `null` للقيمة المعادة.	
Effect	عند تنفيذ هذه العملية، يمكن أن تصدر CH الإجابة على نداء إجراء يتوافق مع معرف توقيع `signature` ومعرف مكون `receiver`. وترسل CH الإجابة إلى TE بعيد يجرى فيه تنفيذ `receiver` ويضع الإجابة في صف انتظار TE البعيد.	

tciReplyConnectedBC 8.2.3.3.7

Signature	`void tciReplyConnectedBC(in TriPortIdType sender,`
	`in TriSignatureIdType signature,`
	`in TciParameterListType parameterList,`
	`in Value returnValue)`	

In Parameters	`sender`	معرف المنفذ المرسل للإجابة.
	`signature`	معرف توقيع نداء الإجراء.
	`parameterList`	قائمة معلمات مشفرة هي جزء من التوقيع الدال عليه. وترتب المعطيات في `parameterList` كما تظهر في إعلان توقيع TTCN-3.
	`returnValue`	إعادة (خيارية) لقيمة نداء الإجراء.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية إجابة إذاعية على منفذ مكون TTCN-3، الذي تم توصيله بمنفذ مكون آخر. وتحتوي جميع معلمات إجراء *in* و*inout* والقيمة العائدة على قيم. وتحتوي معلمات جميع *in* على قيمة مميزة `null` بسبب أنها تتعلق بنداء إجراء ولكن ليس في الإجابة على النداء. وتحتوي `parameterList` على معلمات نداء إجراء. وهذه المعلمات هي المعلمات المحددة في مقاس توقيع TTCN-3. وإذا لم يعرف نمط إعادة لتوقيع الإجراء في TTCN-3، تمرر القيمة المميزة `null` للقيمة المعادة.	
Effect	عند تنفيذ هذه العملية، يمكن أن تصدر CH الإجابة على نداء إجراء يتوافق مع معرف توقيع `signature` وجميع المكونات الموصولة بـــ `sender`. وترسل CH الاستثناء إلى جميع TEs البعيدة التي ينفذ فيها مستقبلات ويضع الاستثناء في صف انتظار TEs البعيدة.	

tciReplyConnectedMC 9.2.3.3.7

Signature	`void tciReplyConnectedMC(in TriPortIdType sender,`
	`in TriComponentIdListType receivers,`
	`in TriSignatureIdType signature,`
	`in TciParameterListType parameterList,`
	`in Value returnValue)`	

In Parameters	`sender`	معرف المنفذ المرسل للإجابة.
	`receivers`	معرف المكونات المستقبلة للإجابة.
	`signature`	معرف توقيع نداء الإجراء.
	`parameterList`	قائمة معلمات مشفرة هي جزء من التوقيع الدال عليه. وترتب المعطيات في parameterList كما تظهر في إعلان توقيع TTCN-3.
	`returnValue`	إعادة (خيارية) لقيمة نداء الإجراء.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية إجابة متعددة على منفذ مكون TTCN-3، الذي تم توصيله بمنفذ مكون آخر تم توصيله بمنافذ مكون آخر. وتحتوي جميع معلمات إجراء *in*، *inout* والقيمة العائدة على قيم. وتحتوي معلمات جميع *in* على قيمة مميزة `null` بسبب أنها تتعلق بنداء إجراء ولكن ليس في الإجابة على النداء. وتحتوي `parameterList` على معلمات نداء إجراء. وهذه المعلمات هي المعلمات المحددة في مقاس توقيع TTCN-3. وإذا لم يعرف نمط إعادة لتوقيع الإجراء في TTCN-3، تمرر القيمة المميزة `null` للقيمة المعادة.	
Effect	عند تنفيذ هذه العملية، يمكن أن تصدر CH الإجابة على نداء إجراء يتوافق مع معرف توقيع `signature` ومعرف مكون واحد في `receivers`. وترسل CH الإجابة إلى جميع TEs البعيدة التي ينفذ فيها `receivers` ويضع الإجابة في صف انتظار TEs البعيدة.	

tciRaiseConnected 10.2.3.3.7

Signature	`void tciRaiseConnected(in TriPortIdType sender,`
	`in TriComponentIdType receiver,`
	`in TriSignatureIdType signature,`
	`in Value exception)`	

In Parameters	`sender`	معرف المنفذ المرسل للإجابة.
	`receiver`	معرف المكونات المستقبلة للإجابة.
	`signature`	معرف توقيع نداء الإجراء.
	`exception`	The exception value.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية انتشار أحادية على منفذ مكون TTCN-3، الذي تم توصيله بمنفذ مكون آخر.	
Effect	عند تنفيذ هذه العملية، يمكن أن تنشر CH الاستثناء إلى نداء إجراء يتوافق مع معرف توقيع `signature` ومعرف مكون `receiver`. وترسل CH الاستثناء إلى TE البعيد الذي يجرى فيه تنفيذ `receiver` ويضع الاستثناء في صف انتظار TE البعيد.	

tciRaiseConnectedBC 11.2.3.3.7

Signature	`void tciRaiseConnectedBC(in TriPortIdType sender,`
	`in TriSignatureIdType signature,`
	`in Value exception)`	

In Parameters	`sender`	Identifier of the port sending the reply.
	`signature`	Identifier of the signature of the procedure call.
	`exception`	The exception value.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية انتشار إذاعية على منفذ مكون TTCN-3، الذي تم توصيله بمنافذ مكون آخر.	
Effect	عند تنفيذ هذه العملية، يمكن أن تصدر CH الاستثناء إلى نداء إجراء يتوافق مع معرف توقيع `signature` وجميع المكونات الموصولة بـــ `sender`. وترسل CH الاستثناء إلى جميع TEs البعيدة التي يجرى فيها تنفيذ المستقبلات ويضع الاستثناء في صف انتظار TEs البعيدة.	

tciRaiseConnectedMC 12.2.3.3.7

Signature	`void tciRaiseConnectedMC(in TriPortIdType sender,`
	`in TriComponentIdListType receiver,`
	`in TriSignatureIdType signature,`
	`in Value exception)`	

In Parameters	`sender`	Identifier of the port sending the reply.
	`receivers`	Identifiers of the component receiving the reply.
	`signature`	Identifier of the signature of the procedure call.
	`exception`	The exception value.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية انتشار متعددة على منفذ مكون TTCN-3، الذي تم توصيله بمنفذ مكون آخر.	
Effect	عند تنفيذ هذه العملية، يمكن أن تصدر CH الاستثناء إلى نداء إجراء يتوافق مع معرف توقيع `signature` ومعرف مكون واحد `receivers`. وترسل CH الاستثناء إلى جميع TEs البعيدة التي يجرى فيها تنفيذ `receivers` ويضع الاستثناء في صف انتظار TEs البعيدة.	

tciCreateTestComponentReq 13.2.3.3.7

Signature	`TriComponentIdType tciCreateTestComponentReq(in TciTestComponentKindType kind, in Type componentType, in TString name)`	
In Parameters	`kind`	نوع المكون الذي ينشأ، سواء `MTC` ,`PTC` أو `CONTROL`.
	`componentType`	معرف نمط مكون TTCN-3 الذي ينشأ.
Return Value	قيمة `TriComponentIdType` للمكون المنشأ.	
Constraint	تطلب هذه العملية من TE عندما يتعين إنشاء مكون سواء صراحة عندما يطلب إنشاء عملية TTCN-3 أو ضمنيا عندما يتعين إنشاء مكون اختبار رئيسي (MTC) أو مكون تحكم. ويضبط `name` على القيمة المميزة `null` إذا لم يعط اسم في بيان إنشاء TTCN-3.	
Effect	ترسل CH طلب إنشاء مكون إلى TE البعيد وتطلب هناك عملية `tciCreateTestComponent` للحصول على معرف مكون لهذا المكون.	

tciStartTestComponentReq 14.2.3.3.7

Signature	`void tciStartTestComponentReq(in TriComponentIdType component, in TciBehaviourIdType behaviour, in TciParamaterListType parameterList)`	
In Parameters	`component`	معرف المكون الذي يبدأ.
	`behaviour`	معرف السلوك الذي يبدأ على المكون.
	`parameterList`	قائمة قيم حيث كل قيمة تعرف معلمة من قائمة المعلمات كما تعرف في إعلان وظيفة لوظيفة تبدأ. وترتب معلمات `parameterList` كما تظهر في توقيع TTCN-3 لاختبار مجرد. وإذا لم تمرر معلمات، سواء القيمة `null` أو `parameterList` فارغة، أي، تمرر قائمة طولها صفر.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية بدء TTCN-3. ونظراً لأن معلمات *in* يسمح لها ببدء وظائف [2] ITU-T Z.140، تحتوي `parameterList` على معلمات *in* فقط.	
Effect	ترسل CH طلب مكون بدء إلى TE البعيد وتطلب هناك عملية `tciStartTestComponent`.	

tciStopTestComponentReq 15.2.3.3.7

Signature	`void tciStopTestComponentReq(in TriComponentIdType component)`	
In Parameters	`component`	معرف مكون يتعين وقفه.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية توقف TTCN-3.	
Effect	ترسل CH طلب مكون وقف إلى TE وتطلب هناك عملية `tciStopTestComponent`.	

tciConnectReq 16.2.3.3.7

Signature	`void tciConnectReq(in TriPortIdType fromPort, in TriPortIdType toPort)`	
In Parameters	`fromPort`	معرف منفذ مكون اختبار يتم التوصيل منه.
	`toPort`	معرف منفذ مكون اختبار يتم التوصيل إليه.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية توصيل TTCN-3.	
Effect	ترسل CH طلب التوصيل إلى TE البعيد حيث يطلب عملية `tciConnect` لإنشاء توصيل منطقي بين منفذين. ولاحظ أن كلا المنفذين يمكن أن يكونا على TEs. وفي هذه الحالة، تعاد العملية فقط بعد طلب عملية `tciConnect` على كلا TEs البعيدين.	

tciDisconnectReq 17.2.3.3.7

Signature	`void tciDisconnectReq(in TriPortIdType fromPort, in TriPortIdType toPort)`	
In Parameters	`fromPort`	Identifier of the test component port to be disconnected.
	`toPort`	Identifier of the test component port to be disconnected.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية فك توصيل TTCN-3.	
Effect	ترسل CH طلب فك توصيل إلى TE البعيد حيث تطلب عملية `tciConnect` لإلغاء توصيل منطقي بين منفذين. ولاحظ أن كلا المنفذين يمكن أن يكونا على TEs. وفي هذه الحالة، تعاد العملية فقط بعد طلب عملية `tciConnect` على كلا TEs البعيدين.	

tciMapReq 18.2.3.3.7

Signature	`void tciMapReq(in TriPortIdType fromPort, in TriPortIdType toPort)`	
In Parameters	`fromPort`	معرف منفذ مكون اختبار يتم التوصيل منه.
	`toPort`	معرف منفذ مكون اختبار يتم التوصيل إليه.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية تقابل TTCN-3.	
Effect	ترسل CH طلب تقابل إلى TE البعيد حيث تطلب عملية `tciMap` لإنشاء توصيل منطقي بين منفذين.	

tciUnmapReq 19.2.3.3.7

Signature	`void tciUnmapReq(in TriPortIdType fromPort, in TriPortIdType toPort)`	
In Parameters	`fromPort`	Identifier of the test component port to be unmapped.
	`toPort`	Identifier of the test component port to be unmapped.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية فك تقابل TTCN-3.	
Effect	ترسل CH طلب فك تقابل إلى TE البعيد حيث تطلب عملية `tciUnmap` لإلغاء توصيل منطقي بين منفذين.	

tciTestComponentTerminatedReq 20.2.3.3.7

Signature	`void tciTestComponentTerminatedReq(in TriComponentIdType component, in VerdictValue verdict)`	
In Parameters	`component`	معرف المكون الذي تم انتهاؤه.
	`verdict`	الحكم بعد انتهاء مكون.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينهي مكون اختبار تنفيذ، سواء صراحة مع عملية وقف TTCN-3، أو ضمنياً، إذا كان قد توصل إلى آخر بيان.	
Effect	تبلغ CH بانتهاء مكون اختبار دال عليه. ونظراً لتنفيذ وظيفة على مكون اختبار يمكنه فقط أن يكون له معلمات *in* ([2] ITU-T Z.140)، لا يكون لعملية `tciTestComponentTerminateReq` معلمة `parameterList`. وترسل CH انتهاء المكون الدال عليه إلى جميع TEs المشاركين وإلى TE* خاص، الذي يتتبع الحكم الشامل.	

tciTestComponentRunningReq 21.2.3.3.7

Signature	`TBoolean tciTestComponentRunningReq(in TriComponentIdType component)`	
In Parameters	`component`	معرف مكون للتأكد من تمريره.
Return Value	`true` إذا كان المكون الدال عليه مازال ينفذ سلوك، وإلا يكون `false`.	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية تمرير TTCN-3.	
Effect	ترسل CH طلب تمرير إلى TE بعيد له مكون اختبار يتعين التأكد منه، حيث يطلب عملية `tciTestComponentRunning` للتأكد من حالة تنفيذ مكون الاختبار الدال عليه.	

tciTestComponentDoneReq 22.2.3.3.7

Signature	`TBoolean tciTestComponentDoneReq(in TriComponentIdType comp)`	
In Parameters	`comp`	معرف مكون للتأكد من أنه نفذ.
Return Value	`true` إذا كان المكون الدال عليه قد أكمل تنفيذ سلوكه، وإلا يكون `false`.	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية `done` TTCN-3.	
Effect	ترسل CH طلب تنفيذ إلى TE بعيد له مكون اختبار يتعين التأكد منه، حيث يطلب عملية `tciTestComponentDone` للتأكد من حالة تنفيذ مكون الاختبار الدال عليه.	

tciGetMTCReq 23.2.3.3.7

Signature	`TriComponentIdType tciGetMTCReq()`
Return Value	قيمة `TriComponentIdType` لـــ MTC.
Constraint	يطلب TE هذه العملية عندما ينفذ عملية `mtc` TTCN-3.
Effect	تحدد CH هوية مكون MTC.

tciExecuteTestCaseReq 24.2.3.3.7

Signature	`void tciExecuteTestCaseReq(in TciTestCaseIdType testCaseId,` `in TriPortIdListType tsiPortList)`	
In Parameters	`testCaseId`	معرف اختبار مجرد كما عرف في وحدة TTCN-3.
	`tsiPortList`	تحتوي `tsiPortList` على جميع المنافذ التي تم الإعلان عنها في تعريف مكون النظام لاختبار مجرد، أي، منافذ TSI. وإذا لم يعرف مكون نظام بوضوح لاختبار مجرد، فإن `tsiPortList` تحتوي على جميع منافذ الاتصالات لـــ MTC. وترتب جميع المنافذ في `tsiPortList` كما تظهر في إعلان نمط مكون TTCN-3 المعني. وإذا لم تمرر أي منافذ سواء بقيمة `null` أو `tsiPortList` فارغة، أي، تمرر قائمة طولها صفر.
Return Value	`void`	
Constraint	يمكن لـــ TE أن يطلب هذه العملية مباشرة قبل أن يبدأ سلوك الاختبار المجرد على MTC (خلال عملية تنفيذ TTCN-3).	
Effect	ترسل CH طلب تنفيذ اختبار مجرد إلى TEs بعيدين لها منافذ نظام للاختبار المجرد الدالة عليها. ويمكن إنشاء توصيلات سكونية إلى SUT وتدميث اتصالات لمنافذ TSI.	

tciResetReq 25.2.3.3.7

Signature `void tciResetReq()`
Return Value `void`
Constraint يمكن لـــ TE أن يطلب هذه العملية في أي وقت لإعادة تدميث نظام الاختبار.
Effect ترسل CH طلب إعادة تدميث إلى جميع TEs المتضمنة.

tciKillTestComponentReq 26.2.3.3.7

Signature	`void tciKillTestComponentReq(in TriComponentIdType comp)`	
In Parameters	`comp`	معرف المكون الذي يكبت.
Return Value	`void`	
Constraint	يطلب TE هذه العملية عندما ينفذ عملية `kill` TTCN-3.	
Effect	ترسل CH طلب مكون كبت إلى TE بعيد وتطلب هناك عملية `tciKillTestComponent`.	

tciTestComponentAliveReq 27.2.3.3.7

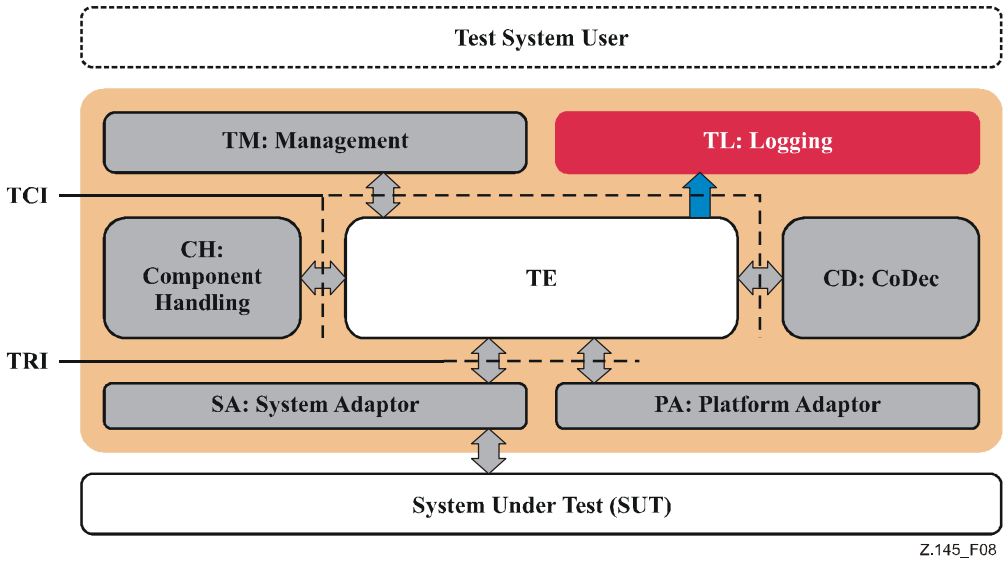
`TBoolean tciTestComponentAliveReq(in TriComponentIdType comp)`		**Signature**
`comp`	معرف مكون للتأكد من أنه نشط.	**In Parameters**
`true` إذا كان المكون الدال عليه نشطا، وإلا يكون `false`.		**Return Value**
يطلب TE هذه العملية عندما ينفذ عملية `alive` TTCN-3.		**Constraint**
ترسل CH الطلب إلى TE البعيد الذي أنشأ مكون الاختبار قيد التنفيذ، حيث يطلب عملية `tciTestComponentAlive` للتأكد من حالة التنفيذ لمكون الاختبار الدال عليه.		**Effect**

tciTestComponentKilledReq 28.2.3.3.7

`TBoolean tciTestComponentKilledReq(in TriComponentIdType comp)`		**Signature**
`comp`	معرف مكون للتأكد من أنه كبت.	**In Parameters**
`true` إذا كان المكون الدال عليه تم كبته، وإلا يكون `false`.		**Return Value**
يطلب TE هذه العملية عندما ينفذ عملية `killed` TTCN-3.		**Constraint**
ترسل CH الطلب إلى TE البعيد الذي أنشأ مكون الاختبار قيد التنفيذ، حيث يطلب عملية `tciTestComponentKilled` للتأكد من حالة التنفيذ لمكون الاختبار الدال عليه.		**Effect**

4.3.7 السطح البيني TCI-TL

يصف السطح البيني لتسجيل اختبار TCI العمليات التي يطلبها TTCN-3 قابل للتنفيذ لينفذها وعمليات تنفيذ تسجيل اختبار يوفرها إلى TE (الشكل 8).



الشكل Z.145/8 – السطح البيني TCI-TL

يوفر التسجيل لجميع عمليات مستوى TTCN-3 لعملية تسجيل الحدث المعني الذي يؤدى بواسطة TE أو SAأو PA أوCH أو CD إلى المستعمل.

1.4.3.7 موفر TCI-TL

يصف هذا القسم العمليات التي يوفرها TL إلى TE.

tliTcExecute 1.1.4.3.7

`void tliTcExecute(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TciTestCaseIdType tcId, in TriParameterListType pars,`
`in TriTimerDurationType dur)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The testcase to be executed.	`tcId`	
The list of parameters required by the testcase.	`pars`	
Duration of the execution.	`dur`	
	`void`	**Return Value**
Shall be called by TE to log the execute test case request.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTcStart 2.1.4.3.7

`void tliTcStart(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TciTestCaseIdType tcId, in TriParameterListType pars,`
`in TriTimerDurationType dur)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The testcase to be executed.	`tcId`	
The list of parameters required by the testcase.	`pars`	
Duration of the execution.	`dur`	
	`void`	**Return Value**
Shall be called by TE to log the start of a testcase. This event occurs before the testcase is started.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTcStop 3.1.4.3.7

`void tliTcStop(in TString am, in TInteger ts,`
`in TString src, in TInteger line,`
`in TriComponentIdType c)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
	`void`	**Return Value**
Shall be called by TE to log the stop of a testcase.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTcStarted 4.1.4.3.7

`void tliTcStarted(in TString am, in TInteger ts,`
`in TString src, in TInteger line,`
`in TriComponentIdType c, in TciTestCaseIdType tcId,`
`in TriParameterListType pars, in TriTimerDurationType dur)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The testcase to be executed.	`tcid`	
The list of parameters required by the testcase.	`pars`	
Duration of the execution.	`dur`	
	`void`	**Return Value**
Shall be called by TM to log the start of a testcase. This event occurs after the testcase was started.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTcTerminated 5.1.4.3.7

`void tliTcTerminated(in TString am, in TInteger ts,`
`in TString src, in TInteger line,`
`in TriComponentIdType c, in TciTestCaseIdType tcId,`
`in TriParameterListType pars, in VerdictValue outcome)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The testcase to be executed.	`tcId`	
The list of parameters required by the testcase.	`pars`	
The verdict of the testcase.	`outcome`	
	`void`	**Return Value**
Shall be called by TM to log the termination of a testcase. This event occurs after the testcase terminated.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCtrlStart 6.1.4.3.7

`void tliCtrlStart(in TString am, in TInteger ts,`
`in TString src, in TInteger line,`
`in TriComponentIdType c)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
	`void`	**Return Value**
Shall be called by TE to log the start of the control part. This event occurs before the control is started. If the control is not represented by a TRI component, c is null.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCtrlStop 7.1.4.3.7

`void tliCtrlStop(in TString am, in TInteger ts,`
`in TString src, in TInteger line,`
`in TriComponentIdType c)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
`void`		**Return Value**
Shall be called by TE to log the stop of the control part. This event occurs before the control is stopped. If the control is not represented by a TRI component, c is null.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCtrlTerminated 8.1.4.3.7

`void tliCtrlTerminated(in TString am, in TInteger ts,`
`in TString src, in TInteger line,`
`in TriComponentIdType c)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
	`void`	**Return Value**
Shall be called by TM to log the termination of the control part. This event occurs after the control has terminated. If the control is not represented by a TRI component, c is null.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMSend_m 9.1.4.3.7

`void tliMSend_m(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in Value msgValue,`
`in TriAddressType address,`
`in TriStatusType encoderFailure,`
`in TriMessageType msg,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is sent.	`port`	
The value to be encoded and sent.	`msgValue`	
The address of the destination within the SUT.	`address`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded message.	`msg`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a unicast send operation. This event occurs after sending. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMSend_m_BC 10.1.4.3.7

`void tliMSend_m_BC(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in Value msgValue,`
`in TriStatusType encoderFailure,`
`in TriMessageType msg,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is sent.	`port`	
The value to be encoded and sent.	`msgValue`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded message.	`msg`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a broadcast send operation. This event occurs after sending. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMSend_m_MC 11.1.4.3.7

`void tliMSend_m_MC(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in Value msgValue,`
`in TriAddressListType addresses,`
`in TriStatusType encoderFailure,`
`in TriMessageType msg,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is sent.	`port`	
The value to be encoded and sent.	`msgValue`	
The addresses of the destinations within the SUT.	`addresses`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded message.	`msg`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a multicast send operation. This event occurs after sending. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMSend_c 12.1.4.3.7

`void tliMSend_c(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in Value msgValue,`
`in TriComponentIdType to,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is sent.	`port`	
The value to be encoded and sent.	`msgValue`	
The component which will receive the message.	`to`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a unicast send operation. This event occurs after sending. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMSend_c_BC 13.1.4.3.7

`void tliMSend_c_BC(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in Value msgValue,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is sent.	`port`	
The value to be encoded and sent.	`msgValue`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a broadcast send operation. This event occurs after sending. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMSend_c_MC 14.1.4.3.7

`void tliMSend_c_MC(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in Value msgValue,`
`in TriComponentIdListType toList,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is sent.	`port`	
The value to be encoded and sent.	`msgValue`	
The components which will receive the message.	`toList`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a multicast send operation. This event occurs after sending. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMDetected_m 15.1.4.3.7

`void tliMDetected_m(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriMessageType msg,`
`in TriAddressType address)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is received.	`port`	
The received encoded message.	`msg`	
The address of the source within the SUT.	`address`	
	`void`	**Return Value**
Shall be called by SA to log the enqueuing of a message. This event occurs after the message is enqueued. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMDetected_c 16.1.4.3.7

`void tliMDetected_c(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in Value msgValue,`
`in TriComponentIdType from)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is received.	`port`	
The received message.	`msgValue`	
The component which sent the message.	`from`	
	`void`	**Return Value**
Shall be called by CH to log the enqueuing of a message. This event occurs after the message is enqueued. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMMismatch_m 17.1.4.3.7

`void tliMMismatch_m(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in Value msgValue,`
`in TciValueTemplate msgTmpl,`
`in TciValueDifferenceList diffs,`
`in TriAddressType address,`
`in TciValueTemplate addressTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is received.	`port`	
The message which is checked against the template.	`msgValue`	
The template used to check the message match.	`msgTmpl`	
The difference/the mismatch between message and template.	`diffs`	
The address of the source within the SUT.	`address`	
The expected address of the source within the SUT.	`addressTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the mismatch of a template. This event occurs after checking a template match. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMMismatch_c 18.1.4.3.7

`void tliMMismatch_c(in TString am, in TInteger ts,`
`in TString src, in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in Value msgValue,`
`in TciValueTemplate msgTmpl,`
`in TciValueDifferenceList diffs,`
`in TriComponentIdType from,`
`in TciNonValueTemplate fromTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is received.	`port`	
The message which is checked against the template.	`msgValue`	
The template used to check the message match.	`msgTmpl`	
The difference/the mismatch between message and template.	`diffs`	
The component which sent the message.	`from`	
The expected sender component.	`fromTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the mismatch of a template. This event occurs after checking a template match. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMReceive_m 19.1.4.3.7

`void tliMReceive_m(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port,`
`in Value msgValue, in TciValueTemplate msgTmpl,`
`in TriAddressType address,`
`in TciValueTemplate addressTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is received.	`port`	
The message which is checked against the template.	`msgValue`	
The template used to check the message match.	`msgTmpl`	
The address of the source within the SUT.	`address`	
The expected address of the source within the SUT.	`addressTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the receive of a message. This event occurs after checking a template match. This event is used for logging the communication with SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliMReceive_c 20.1.4.3.7

`void tliMReceive_c(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port,`
`in Value msgValue, in TciValueTemplate msgTmpl,`
`in TriComponentIdType from,`
`in TciNonValueTemplate fromTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the message is received.	`port`	
The message which is checked against the template.	`msgValue`	
The template used to check the message match.	`msgTmpl`	
The component which sent the message.	`from`	
The expected sender component.	`fromTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the receive of a message. This event occurs after checking a template match. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCall_m 21.1.4.3.7

`void tliPrCall_m(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port,`
`in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TriAddressType address,`
`in TriStatusType encoderFailure,`
`in TriParameterListType pars,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is invoked.	`port`	
The signature of the called operation.	`signature`	
The parameters of the called operation.	`parsValue`	
The address of the destination within the SUT.	`address`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded parameters.	`pars`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a unicast call operation. This event occurs after call execution. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCall_m_BC 22.1.4.3.7

`void tliPrCall_m_BC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port,`
`in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TriStatusType encoderFailure,`
`in TriParameterListType pars,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is invoked.	`port`	
The signature of the called operation.	`signature`	
The parameters of the called operation.	`parsValue`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded parameters.	`pars`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a broadcast call operation. This event occurs after call execution. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCall_m_MC 23.1.4.3.7

`void tliPrCall_m_MC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port,`
`in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TriAddressListType addresses,`
`in TriStatusType encoderFailure,`
`in TriParameterListType pars,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is invoked.	`port`	
The signature of the called operation.	`signature`	
The parameters of the called operation.	`parsValue`	
The addresses of the destinations within the SUT.	`addresses`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded parameters.	`pars`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a multicast call operation. This event occurs after call execution. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCall_c 24.1.4.3.7

`void tliPrCall_c(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port,`
`in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TriComponentIdType to,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is invoked.	`port`	
The signature of the called operation.	`signature`	
The parameters of the called operation.	`parsValue`	
The component which will receive the message.	`to`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a unicast call operation. This event occurs after call execution. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCall_c_BC 25.1.4.3.7

`void tliPrCall_c_BC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port,`
`in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is invoked.	`port`	
The signature of the called operation.	`signature`	
The parameters of the called operation.	`parsValue`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a broadcast call operation. This event occurs after call execution. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCall_c_MC 26.1.4.3.7

`void tliPrCall_c_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is invoked.	`port`	
The signature of the called operation.	`signature`	
The parameters of the called operation.	`parsValue`	
The component which will receive the message.	`toList`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a multicast call operation. This event occurs after call execution. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetCallDetected_m 27.1.4.3.7

`void tliPrGetCallDetected_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriParameterListType pars, in TriAddressType address)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is received.	`port`	
The signature of the detected call.	`signature`	
The encoded parameters of detected call.	`pars`	
The address of the destination within the SUT.	`address`	
	`void`	**Return Value**
Shall be called by SA to log the getcall enqueue operation. This event occurs after call is enqueued. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetCallDetected_c 28.1.4.3.7

`void tliPrGetCallDetected_c(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TriComponentIdType from)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is received.	`port`	
The signature of the called operation.	`signature`	
The encoded parameters of detected call.	`parsValue`	
The component which called the operation.	`from`	
	`void`	**Return Value**
Shall be called by CH to log the getcall enqueue operation. This event occurs after call is enqueued. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetCallMismatch_m 29.1.4.3.7

`void tliPrGetCallMismatch_m(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TciValueTemplate parsTmpl,`
`in TciValueDifferenceList diffs,`
`in TriAddressType address,`
`in TciValueTemplate addressTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is received.	`port`	
The signature of the detected call.	`signature`	
The parameters of detected call.	`parsValue`	
The template used to check the parameter match.	`parsTmpl`	
The difference/the mismatch between call and template	`diffs`	
The address of the source within the SUT.	`address`	
The expected address of the source within the SUT.	`addressTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the mismatch of a getcall. This event occurs after getcall is checked against a template. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetCallMismatch_c 30.1.4.3.7

`void tliPrGetCallMismatch_c(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TciValueTemplate parsTmpl,`
`in TciValueDifferenceList diffs,`
`in TriComponentIdType from,`
`in TciNonValueTemplate fromTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is received.	`port`	
The signature of the detected call.	`signature`	
The parameters of detected call.	`parsValue`	
The template used to check the parameter match.	`parsTmpl`	
The difference/the mismatch between message and template.	`diffs`	
The component which called the operation.	`from`	
The expected calling component.	`fromTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the mismatch of a getcall. This event occurs after getcall is checked against a template. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetCall_m 31.1.4.3.7

`void tliPrGetCall_m(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TciValueTemplate parsTmpl,`
`in TriAddressType address,`
`in TciValueTemplate addressTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is received.	`port`	
The signature of the detected call.	`signature`	
The parameters of detected call.	`parsValue`	
The template used to check the parameter match.	`parsTmpl`	
The address of the source within the SUT.	`address`	
The expected address of the source within the SUT.	`addressTmpl`	
	`void`	**Return Value**
Shall be called by TE to log getting a call. This event occurs after getcall has matched against a template. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetCall_c 32.1.4.3.7

`void tliPrGetCall_c(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue,`
`in TciValueTemplate parsTmpl,`
`in TriComponentIdType from,`
`in TciNonValueTemplate fromTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the call is received.	`port`	
The signature of the detected call.	`signature`	
The parameters of detected call.	`parsValue`	
The template used to check the parameter match.	`parsTmpl`	
The component which called the operation.	`from`	
The expected calling component.	`fromTmpl`	
	`void`	**Return Value**
Shall be called by TE to log getting a call. This event occurs after getcall has matched against a template. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrReply_m 33.1.4.3.7

`void tliPrReply_m(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in Value parsValue, in Value replValue,`
`in TriAddressType address,`
`in TriStatusType encoderFailure,`
`in TriParameterType repl,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is sent.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The reply to be sent.	`replValue`	
The address of the destination within the SUT.	`address`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded reply.	`repl`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a unicast reply operation. This event occurs after reply execution. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrReply_m_BC 34.1.4.3.7

`void tliPrReply_m_BC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in Value parsValue, in Value replValue,`
`in TriStatusType encoderFailure,`
`in TriParameterType repl,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is sent.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The reply to be sent.	`replValue`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded reply.	`repl`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a broadcast reply operation. This event occurs after reply execution. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrReply_m_MC 35.1.4.3.7

`void tliPrReply_m_MC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in Value parsValue, in Value replValue,`
`in TriAddressListType addresses,`
`in TriStatusType encoderFailure,`
`in TriParameterType repl,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is sent.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The reply to be sent.	`replValue`	
The addresses of the destinations within the SUT.	`addresses`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded reply.	`repl`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a multicast reply operation. This event occurs after reply execution. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrReply_c 36.1.4.3.7

`void tliPrReply_c(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in Value parsValue, in Value replValue,`
`in TriComponentIdType to,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is sent.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The reply to be sent.	`replValue`	
The component which will receive the reply.	`to`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a unicast reply operation. This event occurs after reply execution. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrReply_c_BC 37.1.4.3.7

`void tliPrReply_c_BC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in Value parsValue, in Value replValue,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is sent.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The reply to be sent.	`replValue`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a broadcast reply operation. This event occurs after reply execution. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrReply_c_MC 38.1.4.3.7

`void tliPrReply_c_MC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in Value parsValue, in Value replValue,`
`in TriComponentIdListType toList,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is sent.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The reply to be sent.	`replValue`	
The components which will receive the reply.	`toList`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a multicast reply operation. This event occurs after reply execution. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetReplyDetected_m 39.1.4.3.7

`void tliPrGetReplyDetected_m(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType`
`signature,`
`in TriParameterType repl, in TriAddressType address)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is received.	`port`	
The signature relating to the reply.	`signature`	
The received encoded reply.	`repl`	
The address of the source within the SUT.	`address`	
	`void`	**Return Value**
Shall be called by SA to log the getreply enqueue operation. This event occurs after getreply is enqueued. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetReplyDetected_c 40.1.4.3.7

`void tliPrGetReplyDetected_c(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in Value replValue, in TriComponentIdType from)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is received.	`port`	
The signature relating to the reply.	`signature`	
The received reply.	`replValue`	
The component which sent the reply.	`from`	
	`void`	**Return Value**
Shall be called by CH to log the getreply enqueue operation. This event occurs after getreply is enqueued. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetReplyMismatch_m 41.1.4.3.7

`void tliPrGetReplyMismatch_m(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value replValue,`
`in TciValueTemplate replyTmpl,`
`in TciValueDifferenceList diffs,`
`in TriAddressType address,`
`in TciValueTemplate addressTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is received.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The received reply.	`replValue`	
The template used to check the reply match.	`replyTmpl`	
The difference/the mismatch between reply and template	`diffs`	
The address of the source within the SUT.	`address`	
The expected address of the source within the SUT.	`addressTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the mismatch of a getreply operation. This event occurs after getreply is checked against a template. This event is used for logging the communication with SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetReplyMismatch_c 42.1.4.3.7

`void tliPrGetReplyMismatch_c(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value replValue,`
`in TciValueTemplate replyTmpl,`
`in TciValueDifferenceList diffs,`
`in TriComponentIdType from,`
`in TciNonValueTemplate fromTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is received.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The received reply.	`replValue`	
The template used to check the reply match.	`replyTmpl`	
The difference/the mismatch between reply and template	`diffs`	
The component which sent the reply.	`from`	
The expected replying component.	`fromTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the mismatch of a getreply operation. This event occurs after getreply is checked against a template. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetReply_m 43.1.4.3.7

`void tliPrGetReply_m(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value replValue,`
`in TciValueTemplate replyTmpl,`
`in TriAddressType address,`
`in TciValueTemplate addressTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is received.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The received reply.	`replValue`	
The template used to check the reply match.	`replyTmpl`	
The address of the source within the SUT.	`address`	
The expected address of the source within the SUT.	`addressTmpl`	
	`void`	**Return Value**
Shall be called by TE to log getting a reply. This event occurs after getreply is checked against a template. This event is used for logging the communication with SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrGetReply_c 44.1.4.3.7

`void tliPrGetReply_c(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value replValue,`
`in TciValueTemplate replyTmpl,`
`in TriComponentIdType from,`
`in TciNonValueTemplate fromTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the reply is received.	`port`	
The signature relating to the reply.	`signature`	
The signature parameters relating to the reply.	`parsValue`	
The received reply.	`replValue`	
The template used to check the reply match.	`replyTmpl`	
The component which sent the reply.	`from`	
The expected replying component.	`fromTmpl`	
	`void`	**Return Value**
Shall be called by TE to log getting a reply. This event occurs after getreply is checked against a template. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrRaise_m 45.1.4.3.7

`void tliPrRaise_m(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value excValue,`
`in TriAddressType address,`
`in TriStatusType encoderFailure,`
`in TriExceptionType exc,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is sent.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The exception to be sent.	`excValue`	
The address of the destination within the SUT.	`address`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded exception.	`exc`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a unicast raise operation. This event occurs after reply execution. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrRaise_m_BC 46.1.4.3.7

`void tliPrRaise_m_BC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value excValue,`
`in TriStatusType encoderFailure,`
`in TriExceptionType exc,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is sent.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The exception to be sent.	`excValue`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded exception.	`exc`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a broadcast raise operation. This event occurs after reply execution. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrRaise_m_MC 47.1.4.3.7

`void tliPrRaise_m_MC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value excValue,`
`in TriAddressListType addresses,`
`in TriStatusType encoderFailure,`
`in TriExceptionType exc,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is sent.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The exception to be sent.	`excValue`	
The addresses of the destinations within the SUT.	`addresses`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded exception.	`exc`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by SA to log a multicast raise operation. This event occurs after reply execution. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrRaise_c 48.1.4.3.7

`void tliPrRaise_c(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value excValue,`
`in TriComponentIdType to,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is sent.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The exception to be sent.	`excValue`	
The component which will receive the reply.	`to`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a unicast raise operation. This event occurs after reply execution. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrRaise_c_BC 49.1.4.3.7

`void tliPrRaise_c_BC(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value excValue,`
`in TriStatusType transmissionFailure)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is sent.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The exception to be sent.	`excValue`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a broadcast raise operation. This event occurs after reply execution. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrRaise_c_MC 50.1.4.3.7

`void tliPrRaise_c_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is sent.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The exception to be sent.	`excValue`	
The components which will receive the reply.	`toList`	
The failure message which might occur at transmission.	`transmissionFailure`	
	`void`	**Return Value**
Shall be called by CH to log a multicast raise operation. This event occurs after reply execution. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCatchDetected_m 51.1.4.3.7

`void tliPrCatchDetected_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriExceptionType exc, in TriAddressType address)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is received.	`port`	
The signature relating to the exception.	`signature`	
The catched exception.	`exc`	
The address of the source within the SUT.	`address`	
	`void`	**Return Value**
Shall be called by SA to log the catch enqueue operation. This event occurs after catch is enqueued. This event is used for logging the communication with the SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCatchDetected_c 52.1.4.3.7

`void tliPrCatchDetected_c(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in Value excValue, in TriAddressType address)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is received.	`port`	
The signature relating to the exception.	`signature`	
The catched exception.	`excValue`	
The address of the source within the SUT.	`address`	
	`void`	**Return Value**
Shall be called by CH to log the catch enqueue operation. This event occurs after catch is enqueued. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCatchMismatch_m 53.1.4.3.7

`void tliPrCatchMismatch_m(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value excValue,`
`in TciValueTemplate excTmpl,`
`in TciValueDifferenceList diffs,`
`in TriAddressType address,`
`in TciValueTemplate addressTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is received.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The received exception.	`excValue`	
The template used to check the exception match.	`excTmpl`	
The difference/the mismatch between exception and template	`diffs`	
The address of the source within the SUT.	`address`	
The expected address of the source within the SUT.	`addressTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the mismatch of a catch operation. This event occurs after catch is checked against a template. This event is used for logging the communication with SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCatchMismatch_c 54.1.4.3.7

`void tliPrCatchMismatch_c(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value excValue,`
`in TciValueTemplate excTmpl,`
`in TciValueDifferenceList diffs,`
`in TriComponentIdType from,`
`in TciNonValueTemplate fromTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is received.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The received exception.	`excValue`	
The template used to check the exception match.	`excTmpl`	
The difference/the mismatch between exception and template	`diffs`	
The component which sent the reply.	`from`	
The expected replying component.	`fromTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the mismatch of a catch operation. This event occurs after catch is checked against a template. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCatch_m 55.1.4.3.7

`void tliPrCatch_m(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value excValue,`
`in TciValueTemplate excTmpl,`
`in TriAddressType address,`
`in TciValueTemplate addressTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is received.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The received exception.	`excValue`	
The template used to check the exception match.	`excTmpl`	
The address of the source within the SUT.	`address`	
The expected address of the source within the SUT.	`addressTmpl`	
	`void`	**Return Value**
Shall be called by SA to log catching an exception. This event occurs after catch is checked against a template. This event is used for logging the communication with SUT.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCatch_c 56.1.4.3.7

`void tliPrCatch_c(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType signature,`
`in TciParameterListType parsValue, in Value excValue,`
`in TciValueTemplate excTmpl,`
`in TriComponentIdType from,`
`in TciNonValueTemplate fromTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is received.	`port`	
The signature relating to the exception.	`signature`	
The signature parameters relating to the exception.	`parsValue`	
The received exception.	`excValue`	
The template used to check the exception match.	`excTmpl`	
The component which sent the reply.	`from`	
The expected replying component.	`fromTmpl`	
	`void`	**Return Value**
Shall be called by CH to log catching an exception. This event occurs after catch is checked against a template. This event is used for logging the intercomponent communication.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCatchTimeoutDetected 57.1.4.3.7

`void tliPrCatchTimeoutDetected(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriPortIdType port, in TriSignatureIdType`
`signature)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is received.	`port`	
The signature relating to the exception.	`signature`	
	`void`	**Return Value**
Shall be called by PA to log the detection of a catch timeout. This event occurs after the timeout is enqueued.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPrCatchTimeout 58.1.4.3.7

`void tliPrCatchTimeout(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port via which the exception is received.	`port`	
The signature relating to the exception.	`signature`	
	`void`	**Return Value**
Shall be called by TE to log catching a timeout. This event occurs after the catch timeout has been performed.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCCreate 59.1.4.3.7

`void tliCCreate(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TString name)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component which is created.	`comp`	
The name of the component which is created.	`name`	
	`void`	**Return Value**
Shall be called by TE to log the create component operation. This event occurs after component creation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCStart 60.1.4.3.7

`void tliCStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TciBehaviourIdType beh, in TciParameterListType pars)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component which is started.	`comp`	
The behaviour being started on the component.	`beh`	
The parameters of the started behaviour.	`pars`	
	`void`	**Return Value**
Shall be called by TE to log the start component operation. This event occurs after component start.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCRunning 61.1.4.3.7

`void tliCRunning(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component which is checked to be running.	`comp`	
The status of this component.	`status`	
	`void`	**Return Value**
Shall be called by TE to log the running component operation. This event occurs after component running.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCAlive 62.1.4.3.7

`void tliCAlive(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component which is checked to be running.	`comp`	
The status of this component.	`status`	
	`void`	**Return Value**
Shall be called by TE to log the alive component operation. This event occurs after component alive.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCStop 63.1.4.3.7

`void tliCStop(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component which is stopped.	`comp`	
`void`		**Return Value**
Shall be called by TE to log the stop component operation. This event occurs after component stop.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCKill 64.1.4.3.7

`void tliCKill(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component which is stopped.	`comp`	
	`void`	**Return Value**
Shall be called by TE to log the kill component operation. This event occurs after component kill.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCDoneMismatch 65.1.4.3.7

`void tliCDoneMismatch(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTmpl)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The template used to check the done match.	`compTmpl`	
`void`		**Return Value**
Shall be called by TE to log the mismatch of a done component operation. This event occurs after done is checked against a template.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCDone 66.1.4.3.7

`void tliCDone(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TciNonValueTemplate compTmpl)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The template used to check the done match.	`compTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the done component operation. This event occurs after the done operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCKilledMismatch 67.1.4.3.7

`void tliCKilledMismatch(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTmpl)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The template used to check the done match.	`compTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the mismatch of a killed component operation. This event occurs after killed is checked against a template.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCKilled 68.1.4.3.7

`void tliCKilled(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTmpl)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The template used to check the done match.	`compTmpl`	
	`void`	**Return Value**
Shall be called by TE to log the killed component operation. This event occurs after the killed operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliCTerminated 69.1.4.3.7

`void tliCTerminated(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The verdict of the component.	`verdict`	
	`void`	**Return Value**
Shall be called by TE to log the termination of a component. This event occurs after the termination of the component.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPConnect 70.1.4.3.7

`void tliPConnect(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component of the first port to be connected.	`c1`	
The first port to be connected.	`port1`	
The component of the second port to be connected.	`c2`	
The second port to be connected.	`port2`	
	`void`	**Return Value**
Shall be called by CH to log the connect operation. This event occurs after the connect operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPDisconnect 71.1.4.3.7

`void tliPDisconnect(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component of the first port to be disconnected.	`c1`	
The first port to be disconnected.	`port1`	
The component of the second port to be disconnected.	`c2`	
The second port to be disconnected.	`port2`	
	`void`	**Return Value**
Shall be called by CH to log the disconnect operation. This event occurs after the disconnect operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPMap 72.1.4.3.7

`void tliPMap(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component of the first port to be mapped.	`c1`	
The first port to be mapped.	`port1`	
The component of the second port to be mapped.	`c2`	
The second port to be mapped.	`port2`	
	`void`	**Return Value**
Shall be called by SA to log the map operation. This event occurs after the map operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPUnmap 73.1.4.3.7

`void tliPUnmap(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The component of the first port to be unmapped.	`c1`	
The first port to be unmapped.	`port1`	
The component of the second port to be unmapped.	`c2`	
The second port to be unmapped.	`port2`	
	`void`	**Return Value**
Shall be called by SA to log the unmap operation. This event occurs after the unmap operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPClear 74.1.4.3.7

`void tliPClear(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port to be cleared.	`port`	
	`void`	**Return Value**
Shall be called by TE to log the port clear operation. This event occurs after the port clear operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPStart 75.1.4.3.7

`void tliPStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port to be started.	`port`	
	`void`	**Return Value**
Shall be called by TE to log the port start operation. This event occurs after the port start operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPStop 76.1.4.3.7

`void tliPStop(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port to be stopped.	`port`	
	`void`	**Return Value**
Shall be called by TE to log the port stop operation. This event occurs after the port stop operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliPHalt 77.1.4.3.7

`void tliPHalt(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The port to be stopped.	`port`	
	`void`	**Return Value**
Shall be called by TE to log the port halt operation. This event occurs after the port halt operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliEncode 78.1.4.3.7

`void tliEncode(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value val, in TriStatusType encoderFailure, in TriMessageType msg, in TString codec)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The value to be encoded.	`value`	
The failure message which might occur at encoding.	`encoderFailure`	
The encoded value.	`msg`	
The used encoder.	`codec`	
	`void`	**Return Value**
Shall be called by CD to log the encode operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliDecode 79.1.4.3.7

`void tliDecode(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value val, in TriStatusType decoderFailure, in TriMessageType msg, in TString codec)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The value to be decoded.	`msg`	
The failure message which might occur at decoding.	`decoderFailure`	
The decoded value.	`value`	
The used decoder.	`codec`	
	`void`	**Return Value**
Shall be called by CD to log the decode operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTTimeoutDetected 80.1.4.3.7

`void tliTTimeoutDetected(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The timer that timed out.	`timer`	
	`void`	**Return Value**
Shall be called by PA to log the detection of a timeout. This event occurs after timeout is enqueued.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTTimeoutMismatch 81.1.4.3.7

`void tliTTimeoutMismatch(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TciNonValueTemplate timerTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The timer template that did not match.	`timerTmpl`	
	`void`	**Return Value**
Shall be called by TE to log a timeout mismatch. This event occurs after a timeout match failed.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTTimeoutMismatch 82.1.4.3.7

`void tliTTimeoutMismatch(in TString am, in TInteger ts,`
`in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TciNonValueTemplate timerTmpl)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The timer template that matched.	`timerTmpl`	
	`void`	**Return Value**
Shall be called by TE to log a timeout match. This event occurs after a timeout matched.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTStart 83.1.4.3.7

`void tliTStart(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TriTimerIdType timer, in TriTimerDurationType dur)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The timer that is started.	`timer`	
The timer duration.	`dur`	
	`void`	**Return Value**
Shall be called by PA to log the start of a timer. This event occurs after the start timer operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTStop 84.1.4.3.7

`void tliTStop(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The timer that is stopped.	`timer`	
	`void`	**Return Value**
Shall be called by PA to log the stop of a timer. This event occurs after the stop timer operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTRead 85.1.4.3.7

`void tliTRead(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType elapsed)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The timer that is started.	`timer`	
The elapsed time of the timer.	`elapsed`	
	`void`	**Return Value**
Shall be called by PA to log the reading of a timer. This event occurs after the read timer operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliTRunning 86.1.4.3.7

`void tliTRunning(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TBoolean status)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The timer which is checked to be running.	`timer`	
The status of this component.	`status`	
	`void`	**Return Value**
Shall be called by PA to log the running timer operation. This event occurs after the running timer operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliSEnter 87.1.4.3.7

`void tliSEnter(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TString name, in TciParameterListType parsValue,`
`in TString kind)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The name of the scope.	`name`	
The parameters of the scope.	`parsValue`	
The kind of the scope.	`kind`	
	`void`	**Return Value**
Shall be called by TE to log the entering of a scope. This event occurs after the scope has been entered.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliSLeave 88.1.4.3.7

`void tliSLeave(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TString name, in Value val, in TString kind)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The name of the scope.	`name`	
The return value of the scope.	`val`	
The kind of the scope.	`kind`	
	`void`	**Return Value**
Shall be called by TE to log the leaving of a scope. This event occurs after the scope has left.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliVar 89.1.4.3.7

`void tliVar(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TString name, in Value varValue)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The name of the variable.	`name`	
The new value of the variable.	`varValue`	
	`void`	**Return Value**
Shall be called by TE to log the modification of the value of a variable. This event occurs after the values have been changed.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliModulePar 90.1.4.3.7

`void tliModulePar(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value parValue)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The name of the module parameter.	`name`	
The value of the module parameter.	`parValue`	
	`void`	**Return Value**
Shall be called by TE to log the value of a module parameter. This event occurs after the access to the value of a module parameter.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliGetVerdict 91.1.4.3.7

`void tliGetVerdict(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The current value of the local verdict.	`verdict`	
	`void`	**Return Value**
Shall be called by TE to log the getverdict operation. This event occurs after the getverdict operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliSetVerdict 92.1.4.3.7

`void tliSetVerdict(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The value to be set to the local verdict.	`verdict`	
	`void`	**Return Value**
Shall be called by TE to log the setverdict operation. This event occurs after the setverdict operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliLog 93.1.4.3.7

`void tliLog(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TciValueList log)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The string to be logged.	`log`	
	`void`	**Return Value**
Shall be called by TM to log the TTCN-3 statement log. This event occurs after the TTCN-3 log operation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliAEnter 94.1.4.3.7

`void tliAEnter(in TString am, in TInteger ts, in TString src,` `in TInteger line, in TriComponentIdType c)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
	`void`	**Return Value**
Shall be called by TE to log entering an alt. This event occurs after an alt has been entered.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliALeave 95.1.4.3.7

`void tliALeave(in TString am, in TInteger ts, in TString src,` `in TInteger line, in TriComponentIdType c)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
	`void`	**Return Value**
Shall be called by TE to log leaving an alt. This event occurs after the alt has left.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliANomatch 96.1.4.3.7

`void tliANomatch(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
	`void`	**Return Value**
Shall be called by TE to log the nomatch of an alt. This event occurs after the alt has not matched.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliARepeat 97.1.4.3.7

`void tliARepeat(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
	`void`	**Return Value**
Shall be called by TE to log repeating an alt. This event occurs when the alt is being repeated.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliADefaults 98.1.4.3.7

`void tliADefaults(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
	`void`	**Return Value**
Shall be called by TE to log entering the default section. This event occurs after the default section has been entered.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliAActivate 99.1.4.3.7

`void tliAActivate(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in TString name, in TciParameterListType pars,`
`in Value ref)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The name of the default.	`name`	
The parameter of the default.	`pars`	
The resulting default reference.	`ref`	
	`void`	**Return Value**
Shall be called by TE to log the activation of a default. This event occurs after the default activation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliADeactivate 100.1.4.3.7

`void tliADeactivate(in TString am, in TInteger ts, in TString src,`
`in TInteger line, in TriComponentIdType c,`
`in Value ref)`
		Signature

An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
The resulting default reference.	`ref`	
	`void`	**Return Value**
Shall be called by TE to log the deactivation of a default. This event occurs after the default deactivation.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

tliAWait 101.1.4.3.7

`void tliAWait(in string am, in long ts, in string src, in long line,` `in TriComponentId c)`		**Signature**
An additional message.	`am`	**In Parameters**
The time when the event is produced.	`ts`	
The source file of the test specification.	`src`	
The line number where the request is performed.	`line`	
The component which produces this event.	`c`	
	`void`	**Return Value**
Shall be called by TE to log that the component awaits events for a new snapshot.		**Constraint**
The TL presents all the information provided in the parameters of this operation to the user; how this is done is not within the scope of this Recommendation.		**Effect**

8 تقابل لغة Java

1.8 مقدمة

يقدم هذا القسم تقابل لغة TCI Java. ولأسباب الدقة، يقدم تقابل لغة مكرس بدلاً من استخدام OMG IDL مع لغة Java.

يعرف تقابل لغة Java لسطح بيني تحكم TTCN-3 كيف تتقابل تعاريف IDL الواردة في القسم 7 مع لغة Java. وتقابل اللغة مستقل عن استخدام صيغة Java حيث تستخدم تركيبات لغة Java الأساسية فقط.

2.8 الأسماء والمنظورات

1.2.8 الأسماء

بالرغم من عدم وجود تعارضات بين المعرفات المستخدمة في تعريف IDL ولغة Java، تطبق بعض قواعد ترجمة تسمية على معرفات IDL.

تحذف السطوح البينية Java أو معرفات صنف أثر `Type` المستخدم في تعريف IDL.

مثـــال: يتقابل `TciTestCaseIdType` مع `TciTestCaseId` في Java.

ويتوافق التقابل الناتج مع مصطلحات تشفير Java المعيارية.

2.2.8 المنظورات

يتقابل **`tciInterface`** لوحدة IDL مع مجموعة `org.etsi.ttcn3.tci Java`. وتتقابل جميع إعلانات نمط IDL في هذه الوحدة مع أصناف Java أو إعلانات سطح بيني في هذه المجموعة.

تقابل نمط

تقابل نمط أساسي

يلقي الجدول 2 نظرة شاملة على كيف تتقابل أنماط محلية TBoolean وTFloat وTInteger وTString وTStringSeq مع أنماط Java.

الجدول Z.145/2 – تقابلات نمط أساسي

IDL type	Java type
`TBoolean`	`boolean`
`TFloat`	`float`
`TInteger`	`int`
`TString`	`java.lang.String`
`TStringSeq`	`java.lang.String[]`

يعرف النمط المحلي TObjId في القسم المعني بالسطح البيني ObjidValue.

بولاني

يتقابل نمط **`TBoolean`** مع نمط Java الأساسي `boolean`.

طليق

يتقابل نمط **`TFloat`** مع نمط Java الأساسي `float`.

سمة

يتقابل نمط **`TChar`** مع نمط Java الأساسي `char`.

صحيح

يتقابل نمط **`TInteger`** مع نمط Java الأساسي `int`.

سلسلة

يتقابل نمط **TString** مع صنف java.lang.String دون تأكد من مدى أو موثقات السمات في السلسلة. ويمكن تحويل جميع السلاسل الممكنة المعرفة في TTCN-3 إلى java.lang.String.

السلسلة[]

يتقابل **TStringSeq** مع أي صفيف من صنف java.lang.String.

سمة عالمية

يتقابل نمط **TUniversalChar** مع نمط Java الأساسي int. ويستخدم الصحيح الشكل الشرعي كما عرف في 6.2/X.290.

تقابل نمط مركب

يعرف وصف TCI IDL أنماط معرفة لمستعمل باعتبارها أنماطا محلية. وفي تقابل لغة Java ، تتقابل هذه الأنماط مع السطوح البينية Java. وتعرف السطوح البينية الطرق والنعوت المتاحة لأشياء تنفذ السطح البيني هذا.

TciParameterType 1.2.2.8

يتقابل **TciParameterType** مع السطح البيني التالي:

```
// TCI IDL TciParameterType
package org.etsi.ttcn.tci;
public interface TciParameter {
    public String   getParameterName();
    public void     setParameterName(String name);
    public int      getParameterPassingMode();
    public void     setParameterPassingMode(TciParameterPassingMode mode);
    public Value    getParameter();
    public void     setParameter(Value parameter);
}
```

الطرق:

- getParameterName() يعيد اسم المعلمة كما عرف في مواصفة TTCN-3؛
- setParameterName(String name) يضبط اسم معلمة TciParameter هذه على name؛
- getParameterPassingMode() يعيد أسلوب تمرير المعلمة لهذه المعلمة؛
- setParameterPassingMode(TciParameterPassingMode mode) يضبط أسلوب المعلمة هذه المعلمة TriParameter على mode؛
- getParameter() يعيد معلمة قيمة TciParameter هذه أو الشيء null إذا كانت المعلمة تحتوي على قيمة مميزة null؛
- setParameter(Value parameter) يضبط معلمة القيمة لهذا TciParameter على معلمة. وإذا كانت القيمة المميزة هي null تضبط لتدل على أن هذه المعلمة لا تحتفظ بقيمة، وتمرر null Java باعتبارها معلمة.

TciParameterPassingModeType 2.2.2.8

يتقابل **TciParameterPassingModeType** مع السطح البيني التالي:

```
// TCI IDL TciParameterPassingModeType
package org.etsi.ttcn.tci;
public interface TciParameterPassingMode {
    public final static int TCI_IN      = 0;
    public final static int TCI_INOUT   = 1;
    public final static int TCI_OUT     = 2;
}
```

الثوابت:

- `TCI_IN` يستخدم ليدل على أن `TciParameter` هو معلمة `in`؛
- `TCI_INOUT` يستخدم ليدل على أن `TciParameter` هو معلمة `inout`؛
- `TCI_OUT` يستخدم ليدل على أن `TciParameter` هو معلمة `out`؛

3.2.2.8 TciParameterListType

يتقابل **`TciParameterListType`** مع السطح البيني التالي:

```
// TCI IDL TciParameterListType
package org.etsi.ttcn.tci;
public interface TciParameteList {
    public int                      size() ;
    public boolean                  isEmpty() ;
    public java.util.Enumeration    getParameters() ;
    public TciParameter             get(int index) ;
    public void                     clear() ;
    public void                     add(TciParameter parameter) ;
    public void                     setParameters(TciParameter[] parameters) ;
}
```

الطرق:

- `size()` يعيد عدد المعلمات في هذه القائمة؛
- `isEmpty()` يعيد `true` إذا كانت القائمة لا تحتوي على معلمات؛
- `getParameters()` يعيد `Enumeration` عبر المعلمات في القائمة. ويوفر العد المعلمات بنفس الترتيب كما تظهر في القائمة؛
- `get(int index)` يعيد `TciParameter` عند الموضع المحدد؛
- `clear()` يسحب جميع المعلمات من `TciParameterList` هذه؛
- `add(TciParameter parameter)` يضيف `TciParameterList` إلى طرف `TciParameterList` هذه؛
- `setParameter(TciParameter[] parameters)` يملء `TciParameterList` هذه بـــ `parameters`.

4.2.2.8 TciTypeClassType

يتقابل **`TciTypeClassType`** مع السطح البيني التالي:

```
// TCI IDL TciTypeClassType
package org.etsi.ttcn.tci;
public interface TciTypeClass {
    public final static int ADDRESS                 = 0 ;
    public final static int ANYTYPE                 = 1 ;
    public final static int BITSTRING               = 2 ;
    public final static int BOOLEAN                 = 3 ;
    public final static int CHARSTRING              = 5 ;
    public final static int COMPONENT               = 6 ;
    public final static int ENUMERATED              = 7 ;
    public final static int FLOAT                   = 8 ;
    public final static int HEXSTRING               = 9 ;
    public final static int INTEGER                 = 10 ;
    public final static int OBJID                   = 11 ;
    public final static int OCTETSTRING             = 12 ;
    public final static int RECORD                  = 13 ;
    public final static int RECORD_OF               = 14 ;
    public final static int SET                     = 15 ;
    public final static int SET_OF                  = 16 ;
    public final static int UNION                   = 17 ;
    public final static int UNIVERSAL_CHARSTRING    = 19 ;
    public final static int VERDICT                 = 20 ;
}
```

TciTestComponentKindType 5.2.2.8

يتقابل **TciTestComponentKindType** مع السطح البيني التالي:

```
// TCI IDL TciTestComponentKindType
public interface TciTestComponentKind {
    public final static int TCI_CTRL_COMP       = 0;
    public final static int TCI_MTC_COMP        = 1;
    public final static int TCI_PTC_COMP        = 2;
    public final static int TCI_SYSTEM_COMP     = 3;
}
```

TciBehaviourIdType 6.2.2.8

يتقابل **TciBehaviourIdType** مع السطح البيني التالي:

```
// TCI IDL TciBehaviourIdType
package org.etsi.ttcn.tci;
public interface TciBehaviourId extends QualifiedName {
}
```

TciTestCaseIdType 7.2.2.8

يتقابل **TciTestCaseIdType** مع السطح البيني التالي:

```
// TCI IDL TciTestCaseIdType
package org.etsi.ttcn.tci;
public interface TciTestCaseId  extends QualifiedName {
}
```

TciModuleIdType 8.2.2.8

يتقابل **TciModuleIdType** مع السطح البيني التالي:

```
// TCI IDL TciModuleIdType
package org.etsi.ttcn.tci;
public interface TciModuleId  extends QualifiedName {
}
```

TciModuleParameterIdType 9.2.2.8

يتقابل **TciModuleParameterIdType** مع السطح البيني التالي:

```
// TCI IDL TciModuleParameterIdType
package org.etsi.ttcn.tci;
public interface TciModuleParameterId  extends QualifiedName {
}
```

TciModuleParameterListType 10.2.2.8

يتقابل **TciModuleParameterListType** مع السطح البيني التالي:

```
// TCI IDL TciModuleParameterListType
package org.etsi.ttcn.tci;
public interface TciModuleParameterList {
      public int                   size() ;
      public boolean               isEmpty() ;
      public java.util.Enumeration getModuleParameters() ;
      public TciModuleParameter    get(int index) ;
}
```

الطرق:

- `size()` يعيد عدد المعلمات في هذه القائمة؛
- `isEmpty()` يعيد `true` إذا كانت القائمة لا تحتوي على معلمات وحدة؛
- `getModuleParameters()` يعيد `Enumeration` عبر معلمات الوحدة في القائمة. ويوفر العد معلمات الوحدة بنفس الترتيب كما تظهر في القائمة؛
- `get(int index)` يعيد `TciModuleParameter` عند الموضع المحدد.

11.2.2.8 TciModuleParameterType

يتقابل **TciModuleParameterType** مع السطح البيني التالي:

```
// TCI IDL TciModuleParameterType
package org.etsi.ttcn.tci;
public interface TciModuleParameter {
    public String       getModuleParameterName();
    public Value        getDefaultValue();
}
```

الطرق:

- `getModuleParameterName()` يعيد اسم معلمة الوحدة كما عرف في مواصفة TTCN-3؛
- `getDefaultValue()` يعيد معلمة وحدة قيمة بالتغيب لـــ `TciModuleParameter` هذه أو شيء `null` إذا كانت معلمة الوحدة تحتوي على قيمة مميزة `null`؛

12.2.2.8 TciParameterTypeListType

يتقابل **TciParameterTypeListType** مع السطح البيني التالي:

```
// TCI IDL TciParameterTypeListType
package org.etsi.ttcn.tci;
public interface TciParameterTypeList {
    public int                      size() ;
    public boolean                  isEmpty() ;
    public java.util.Enumeration    getParameterTypes() ;
    public TciParameterType         get(int index) ;
}
```

الطرق:

- `size()` يعيد عدد أنماط معلمات في هذه القائمة؛
- `isEmpty()` يعيد `true` إذا كانت القائمة لا تحتوي على أنماط معلمة؛
- `getParameterTypes()` يعيد `Enumeration` عبر أنماط المعلمات في القائمة. ويوفر العد أنماط المعلمات بنفس الترتيب كما تظهر في القائمة؛
- `get(int index)` يعيد `TciParameterType` عند الموضع المحدد.

13.2.2.8 TciModuleIdListType

يتقابل **TciModuleIdListType** مع السطح البيني التالي:

```
// TCI IDL TciModuleIdListType
package org.etsi.ttcn.tci;
public interface TciModuleIdList {
    public int                      size() ;
    public boolean                  isEmpty() ;
    public java.util.Enumeration    getImportedModules() ;
    public TciModuleId              get(int index) ;
}
```

الطرق:

- `size()` يعيد عدد الوحدات في هذه القائمة؛
- `isEmpty()` يعيد `true` إذا لم تحتوي هذه القائمة على وحدات؛
- `getImportedModules()` يعيد `Enumeration` عبر الوحدات في القائمة. ويوفر العد الوحدات بنفس الترتيب كما تظهر في القائمة؛
- `get(int index)` يعيد `TciModuleId` عند الموضع المحدد.

3.2.8 تقابل نمط مجرد

يوضع نموذج أنماط معطيات TTCN-3 في Java باستخدام تقابل نمط مجرد كما عرف في هذا القسم. ويعرف السطح البيني `Type` العمليات التي تستخدم فقط للاسترداد في الأنماط المعرفة TTCN-3. ولا يمكن تركيب أنماط TTCN-3 باستخدام السطح البيني `Type`. وتوضع نماذج الأنماط باستخدام سطح بيني `Type` وحيد، يوفر طرق لتعريف أنماط واسترداد قيم لنمط معين.

1.3.2.8 النمط

يتقابل Type مع السطح البيني التالي:

```
// TCI IDL Type
package org.etsi.ttcn.tci;
public interface Type {
    public TciModuleId  getDefiningModule();
    public String       getName();
    public int          getTypeClass();
    public Value        newInstance();
    public String       getTypeEncoding();
    public String       getTypeEncodingVariant();
    public String[]     getTypeExtension();
}
```

الطرق:

- `getDefiningModule()` يعيد معرف الوحدة لنمط الوحدة المعرف. وإذا مثل النمط نمط قاعدة TTCN-3، تعاد القيمة المميزة willnull؛
- `getName()` يعيد اسم النمط المعرف في وحدة TTCN-3؛
- `getTypeClass()` يعيد صنف النمط للنمط المعني. ويمكن أن يكون لقيمة TciTypeClassType واحد من الثوابت التالية: ADDRESS، ANYTYPE، BITSTRING، BOOLEAN، CHARSTRING، COMPONENT، ENUMERATED، FLOAT، HEXSTRING، INTEGER، OBJID، OCTETSTRING، RECORD، RECORD_OF، SET، SET_OF، UNION، UNIVERSAL_CHARSTRING، VERDICT؛
- `newInstance()` يعيد القيمة المنشأة حديثا لنمط معين. ولا تكون القيمة الأولية للقيمة المنشأة معرفة.
- `getTypeEncoding()` يعيد نعت تشفير النمط كما عرف في وحدة TTCN-3؛
- `getTypeEncodingVariant()` تعيد هذه العملية نعت متغير تشفير القيمة كما عرف في TTCN-3، إن وجد. وإذا لم يعرف نعت متغير تشفير تعاد القيمة المميزة null؛
- `getTypeEncoding()` يعيد نعت تمديد النمط كما عرف في وحدة TTCN-3.

4.2.8 تقابل قيمة مجردة

يمكن استرداد قيم TTCN-3 من TE وتركيبها باستخدام السطح البيني Value interface. ويركب السطح البيني لتقابل القيمة تراتبيا مع Value باعتباره السطح البيني الأساسي. وتم تعريف سطوح بينية متخصصة لأنماط مختلفة لقيم.

1.4.2.8 القيمة

تتقابل Value مع السطح البيني التالي:

```
// TCI IDL Value
package org.etsi.ttcn.tci;
public interface Value {
    public Type     getType() ;
    public boolean  notPresent() ;
    public String   getValueEncoding() ;
    public String   getValueEncodingVariant();
}
```

الطرق:

- `getType()` يعيد النمط للقيمة المحددة؛
- `notPresent()` يعيد true إذا كانت القيمة المحددة هي omit، false وإلا؛
- `getValueEncoding()` تعيد هذه العملية نعت تشفير القيمة كما عرف في TTCN-3، إن وجد. وإذا لم يعرف نعت تشفير تعاد القيمة المميزة null؛
- `getValueEncodingVariant()` تعيد هذه العملية نعت متغير تشفير القيمة كما عرف في TTCN-3، إن وجد. وإذا لم يعرف نعت متغير تشفير تعاد القيمة المميزة null.

IntegerValue 2.4.2.8

يتقابل نمط **IntegerValue** مع السطح البيني التالي:

```
// IntegerValue
package org.etsi.ttcn.tci;
public interface IntegerValue {
    public void     setInteger(int value);
    public int      getInteger();
}
```

الطرق:

- `setInteger(int value)` يضبط IntegerValue هذه على قيمة الصحيح `value`؛
- `getInteger()` يعيد قيمة الصحيح التي تمثلها IntegerValue هذه.

FloatValue 3.4.2.8

يتقابل نمط **FloatValue** مع السطح البيني التالي:

```
// FloatValue
package org.etsi.ttcn.tci;
public interface FloatValue {
    public void     setFloat(float value);
    public float    getFloat();
}
```

الطرق:

- `setFloat(float value)` يضبط FloatValue هذه على قيمة مطلق `value`؛
- `getFloat()` يعيد قيمة المطلق التي تمثلها FloatValue هذه.

BooleanValue 4.4.2.8

يتقابل نمط **BooleanValue** مع السطح البيني التالي:

```
// BooleanValue
package org.etsi.ttcn.tci;
public interface BooleanValue {
    public void     setBoolean(boolean value);
    public boolean  getBoolean();
}
```

الطرق:

- `setBoolean(boolean value)` يضبط BooleanValue هذه على قيمة بولاني `value`؛
- `getBoolean()` يعيد قيمة بولاني التي تمثلها BooleanValue هذه.

ObjidValue 5.4.2.8

يتقابل **ObjidValue** مع السطح البيني التالي:

```
// TCI IDL ObjidValue
package org.etsi.ttcn.tci;
public interface ObjidValue {
    TciObjId    getObjid();
    void        setObjid(TciObjId value);
}
```

الطرق:

- `getObjid()` يعيد قيمة هوية شيء لـــ `objid` TTCN-3؛
- `setObjid(TciObjId value)` يضبط `ObjidValue` هذه على `value`.

6.4.2.8 ObjidValue

يتقابل **TciObjId** مع السطح البيني التالي. يتألف تمثيل Java محلي لـــ TTCN-3 ObjectId من تتابع مرتب لـــ TciObjIdElementS.

```
package org.etsi.ttcn.tci;
public interface TciObjId {
    public int              size() ;
    public void             setObjElement(TciObjIdElement[] objElements) ;
    public TciObjIdElement  getObjElement(int index) ;
}
```

الطرق:

- `size()` يعيد حجم Object هذا في TciObjIdElements؛
- `setObjElement(TciObjIdElement[] objElements)` يضبط ObjId هذا على قائمة objElements؛
- `getObjElement(int index)` يعيد TciObjIdElement إلى موضع index.

7.4.2.8 TciObjIdElement

يمثل TciObjIdElement مكون شيء وحيد في قيمة TTCN-3 ObjId. ويمكن ضبطه باستخدام تمثيل مثل تمثيل ASCII أو كصحيح.

يتقابل **TciObjIdElement** مع السطح البيني التالي:

```
package org.etsi.ttcn.tci;
public interface TciObjIdElement {
    public void     setElementAsAscii(String element) ;
    public void     setElementAsNumber(int element) ;
    public String   getElementAsAscii() ;
    public int      getElementAsNumber() ;
}
```

الطرق:

- `setElementAsAscii(String element)` يضبط التمثيل الداخلي لـــ ObjIdElement على قيمة سلسلة element؛
- `setElementAsNumber(int element)` يضبط التمثيل الداخلي هذا لـــ ObjIdElement على قيمة الصحيح element؛
- `getElementAsAscii()` يعيد التمثيل الداخلي لـــ ObjIdElement هذا باعتباره سلسلة؛
- `getElementAsNumber()` يعيد التمثيل الداخلي لـــ ObjIdElement هذا باعتباره صحيح.

8.4.2.8 CharstringValue

يتقابل **CharstringValue** مع السطح البيني التالي:

```
// TCI IDL CharstringValue
package org.etsi.ttcn.tci;
public interface CharstringValue {
    String  getString();
    void    setString(String value);
    char    getChar(int position);
    void    setChar(int  position, char value);
    int     getLength();
    void    setLength(int len);
}
```

الطرق:

- `getString()` يعيد سلسلة charstring TTCN-3. ويكون التمثيل النصي لـــ charstring TTCN-3 فارغ هو '' بينما طوله هو صفر؛
- `setString(String value)` يضبط CharstringValue هذا على value؛
- `getChar(int position)` يعيد قيمة السمة لسلسلة سمات TTCN-3 عند position. ويدل موضع 0 على السمة الأولى لـــ charstring TTCN-3. وتكون القيم الصالحة لموضع 0 هي 0 إلى length - 1؛

- `setChar(int position, char value)` تضبط السمة عند `position` على `value`. وتكون القيم الصالحة لـــ `position` هي 0 إلى `length - 1`؛
- `getLength()` يعيد طول `CharstringValue` هذا في سمات، ويكون صفر إذا كانت قيمة `CharstringValue` هذا هي `omit`؛
- `setLength(int len)` يضبط طول `CharstringValue` هذا في سمات على `len`.

9.4.2.8 BitstringValue

يتقابل **`BitstringValue`** مع السطح البيني التالي:

```
// TCI IDL BitstringValue
package org.etsi.ttcn.tci;
public interface BitstringValue {
    String  getString ();
    void    setString (String value);
    int     getBit (int position);
    void    setBit (int  position, int value);
    int     getLength ();
    void    setLength (int len);
}
```

الطرق:

- `getString()` يعيد التمثيل النصي لـــ `BitstringValue` كما عرف في TTCN-3. مثلاً، يكون التمثيل النصي لـــ `0101` هو `'0101'B`. ويكون التمثيل النصي لـــ `bitstring` TTCN-3 فارغ هو `''B`، بينما يكون طوله صفراً؛
- `setString(String value)` يضبط القيمة لـــ `BitstringValue` هذا طبقاً للتمثيل النصي كما عرفته `value`. مثلاً، تكون قيمة `BitstringValue` هذا هي `0101` إذا كان التمثيل النصي في قيمة هو `'0101'B`؛
- `getBit(int position)` يعيد القيمة `(1 | 0)` عند موضع السلسلة الثنائية TTCN-3 هذه. ويدل موضع 0 على البتة الأولى لسلسلة ثنائية TTCN-3. وتكون القيم الصالحة لموضع هي من `0` إلى `length - 1`؛
- `setBit(int position, int value)` يضبط البتة عند موضع على قيمة `(1 | 0)`. ويدل الموضع 0 على البتة الأولى في `BitstringValue`. وتكون القيم الصالحة هي من `0` إلى `length` - 1؛
- `getLength()` يعيد طول `BitstringValue` هذه في بتات، صفر إذا كانت القيمة لـــ `BitstringValue` هذه هي `omit`؛
- `setLength(int len)` يضبط الطول لـــ `BitstringValue` هذه في بتات على `len`.

10.4.2.8 OctetstringValue

يتقابل **`OctetstringValue`** مع السطح البيني التالي:

```
// TCI IDL OctetstringValue
package org.etsi.ttcn.tci;
public interface OctetstringValue {
    String  getString();
    void    setString(String value);
    int     getOctet(int position);
    void    setOctet(int  position, int value);
    int     getLength();
    void    setLength(int len);
}
```

الطرق:

- `getString()` يعيد التمثيل النصي لـــ `OctetstringValue` كما عرف في TTCN-3. مثلاً، يكون التمثيل النصي لـــ `0xCAFFEE` هو `'CAFFEE'O`. ويكون التمثيل النصي لـــ `octetstring` TTCN-3 فارغ هو `''O`، بينما يكون طوله صفراً؛
- `setString(String value)` يضبط القيمة لـــ `OctetstringValue` هذا طبقاً للتمثيل النصي كما عرفته `value`. مثلاً، تكون قيمة `OctetstringValue` هذا هي `0xCAFFEE` إذا كان التمثيل النصي في قيمة هو `'CAFFEE'O`؛
- `getBit(int position)` يعيد القيمة (0..255) عند موضع `position` سلسلة أثمونات TTCN-3 هذه. ويدل موضع 0 على الأثمون الأول لسلسلة أثمونات TTCN-3. وتكون القيم الصالحة لموضع هي من `0` إلى `length - 1`؛
- `setBit(int position, int value)` يضبط الأثمون عند `position` على قيمة (0.255). ويدل الموضع 0 على الأثمون الأول في سلسلة الأثمونات. وتكون القيم الصالحة هي من `0` إلى `length - 1`؛
- `getLength()` يعيد طول `OctetstringValue` هذه في أثمونات، صفر إذا كانت القيمة لـــ `OctetstringValue` هذه هي `omit`؛
- `setLength(int len)` يضبط الطول لـــ `OctetstringValue` هذه في أثمونات على `len`.

11.4.2.8 UniversalCharstringValue

يتقابل **`UniversalCharstringValue`** مع السطح البيني التالي:

```
// TCI IDL UniversalCharstringValue
package org.etsi.ttcn.tci;
public interface UniversalCharstringValue {
    String  getString();
    void    setString(String value);
    int     getChar(int position);
    void    setChar(int  position, int value);
    int     getLength();
    void    setLength(int len);
}
```

الطرق:

- `getString()` يعيد التمثيل النصي لـــ `UniversalCharstringValue` هذه، كما عرف في TTCN-3؛
- `setString(String value)` يضبط قيمة `UniversalCharstringValue` هذه طبقاً للتمثيل النصي كما عرفته `value`؛
- `getChar(int position)` يعيد قيمة `universalcharstring` لـــ `UniversalChar` TTCN-3 عند موضع. ويدل موضع 0 على `universal charstring` الأولي لـــ `universal charstring` TTCN-3. وتكون القيم الصالحة لـــ `position` هي من `0` إلى `length - 1`؛
- `setChar(int position, char value)` يضبط `UniversalChar` عند `position` على `value`. وتكون القيم الصالحة لـــ `position` هي من `0` إلى `length - 1`؛
- `getLength()` يعيد الطول لـــ `UniversalCharstringValue` هذه في `UniversalChars`، صفر إذا كانت القيمة لـــ `UniversalCharstringValue` هذه هي `omit`؛
- `setLength(int len)` يضبط طول `UniversalCharstringValue` هذه في `UniversalChars` على `len`.

HexstringValue 12.4.2.8

يتقابل **HexstringValue** مع السطح البيني التالي:

```
// TCI IDL HexstringValue
package org.etsi.ttcn.tci;
public interface HexstringValue {
    String  getString();
    void    setString(String value);
    int     getHex(int position);
    void    setHex(int  position, int value);
    int     getLength();
    void    setLength(int len);
}
```

الطرق:

- `getString()` يعيد التمثيل النصي لـــ `HextstringValue` كما عرف في TTCN-3. مثلاً، يكون التمثيل النصي لـــ `0xAFFEE` هو `H"AFFEE"`. ويكون التمثيل النصي لـــ `hexstring` TTCN-3 فارغ هو `H""`، بينما يكون طوله صفراً؛
- `setString(String value)` يضبط القيمة لـــ `HexstringValue` هذا طبقاً للتمثيل النصي كما عرفته `value`. مثلاً، تكون قيمة `HexstringValue` هذا هي `0xAFFEE` إذا كان التمثيل النصي في `value` هو `H"AFFEE"`؛
- `getHex(int position)` يعيد القيمة `(0..15)` عند موضع `position` سلسلة ستة عشري TTCN-3 هذا. ويدل موضع 0 على الأرقام الستة عشري الأولي لسلسلة ستة عشري TTCN-3. وتكون القيم الصالحة لموضع هي من `0` إلى `length - 1`؛
- `setHex(int position, int value)` يضبط الرقم ستة عشري عند موضع على قيمة `(0..16)`. ويدل الموضع 0 على الأثمون الأول في `hexstring`. وتكون القيم الصالحة هي من `0` إلى `length - 1`؛
- `getLength()` يعيد طول `HexstringValue` هذه في أثمونات، صفر إذا كانت القيمة لـــ `HexstringValue` هذه هي `omit`؛
- `setLength(int len)` يضبط الطول لـــ `HexstringValue` هذه في أرقام ستة عشري على len.

RecordValue 13.4.2.8

يتقابل **RecordValue** مع السطح البيني التالي:

```
// TCI IDL RecordValue
package org.etsi.ttcn.tci;
public interface RecordValue {
    public Value    getField(String fieldName) ;
    public void     setField(String fieldName, Value value) ;
    public String[] getFieldNames() ;
}
```

الطرق:

- `getString()` يعيد قيمة المجال المسمي `fieldName`. وتكون القيمة المعادة هي نمط مشترك لقاعدة مجردة `Value`، كمجال سجل يمكن أن يكون له أي نمط معرف في TTCN-3. وإذا لم يكن من الممكن الحصول على المجال من القيمة المميزة لسجل null يعاد؛
- `setString(String value)` يضبط المجال المسمي `fieldName` للسجل على `value`. ولا يمكن وضع افتراض عن كيفية تخزين مجال في سجل. وقد يختار تنفيذ داخلي أن يخزن مرجع لهذه القيمة أو ينسخ القيمة. ومن الأسلم افتراض أن القيمة سيجرى نسخها. ولهذا ينبغي افتراض أن التعديلات اللاحقة لـــ `value` لن تنعكس في السجل؛
- `getHex(int position)` يعيد صفيف لسلسلة أسماء مجالات، القيمة المميزة `null`، إذا لم يكن للسجل مجالات.

RecordOfValue 14.4.2.8

يتقابل **RecordOfValue** مع السطح البيني التالي:

```
// TCI IDL RecordOfValue
package org.etsi.ttcn.tci;
public interface RecordOfValue {
    public Value    getField(String fieldName) ;
```

```
    public void     setField(int position, Value value) ;
    public void     appendField(Value value) ;
    public Type     getElementType() ;
    public int      getLength() ;
    public void     setLength(int len) ;
}
```

الطرق:

- `getField(String fieldName)` يعيد قيمة record of عند position إذا كان position بين zero و length - 1، وإلا القيمة المميزة null. وتكون القيمة العائدة هي نمط مشترك لقاعدة مجردة Value، لأن a record of يمكن أن يكون له مجالات من أي نمط معرفة في TTCN-3؛
- `setField(int position, Value value)` يضبط المجال عند position على value. وإذا كان position أكبر من (length - 1)، يمدد السجل ليكون له طول (position + 1). ويضبط record of للمكونات بين الموضع الأصلي عن length و position - 1 على OMIT. ولا يمكن وضع افتراض عن كيفية تخزين مجال في record of. وقد يختار تنفيذ داخلي أن يخزن مرجع لهذه القيمة أو ينسخ القيمة. ومن الأسلم افتراض أن القيمة سيجرى نسخها. ولهذا، ينبغي افتراض أن التعديلات اللاحقة لـــ value لن تنعكس في record of؛
- `appendField(Value value)` يذيل القيمة عند طرف record of، أي، عند موضع length. ولا يمكن وضع افتراض عن كيفية تخزين مجال في record of. وقد يختار تنفيذ داخلي أن يخزن مرجع لهذه القيمة أو ينسخ القيمة. ومن الأسلم افتراض أن القيمة سيجرى نسخها. ولهذا ينبغي افتراض أن التعديلات اللاحقة لـــ value لن تنعكس في record of؛
- `getElementType()` تعيد هذه العملية Type لمكونات record of هذا؛
- `getLength()` يعيد الطول الفعلي لـــ record of لقيمة، صفر إذا كان record of لقيمة هو OMIT؛
- `setLength(int len)` يضبط طول record of على len. إذا كان len أكبر من الطول الأصلي، يكون للمكونات التي أنشأت جديداً القيمة OMIT. وإذا كان len أقل أو مساو للطول الأصلي لهذه العملية يتم تجاهلها.

UnionValue 15.4.2.8

يتقابل **UnionValue** مع السطح البيني التالي:

```
// TCI IDL UnionValue
package org.etsi.ttcn.tci;
public interface UnionValue {
    Value       getVariant(String variantName);
    void        setVariant(String variantName, Value value);
    String      getPresentVariantName();
    String[]    getVariantNames();
}
```

الطرق:

- `getVariant(String variantName)` يعيد قيمة variantName لاتحاد TTCN-3، إذا كان variantName) مساو لنتيجة getPresentVariantName، وإلا القيمة المميزة null. ويدل variantName على اسم متغير الاتحاد كما عرف في TTCN-3؛
- `setVariant(String variantName, Value value)`

  يضبط variantName للاتحاد على قيمة. وإذا لم يكن variantName معرفا لهذا الاتحاد، يتم تجاهل هذه العملية. وإذا تم اختيار متغير آخر، يختار المتغير الجيد بدلاً منه؛

- `getPresentVariantName()` يعيد اسم المتغير الذي له قيمة في هذا الاتحاد مضبوطة على String. وتعاد القيمة المميزة null إذا لم يتم اختيار متغير؛
- `getVariantNames()` يعيد صفيف من String لأسماء متغيرات، القيمة المميزة null، إذا لم يكن للاتحاد مجالات. وإذا مثل anytype UnionValue TTCN-3، أي، يكون صنف النمط لنمط حصل عليه (getType()) هو ANYTYPE، تعاد جميع أنماط TTCN-3 المعرفة مسبقاً والمعرفة لمستعمل.

16.4.2.8 EnumeratedValue

يتقابل **EnumeratedValue** مع السطح البيني التالي:

```
// TCI IDL EnumeratedValue
package org.etsi.ttcn.tci;
public interface EnumeratedValue {
    String  getEnum();
    void    setEnum(String enumValue);
}
```

الطرق:

- `getVerdict()` يعيد معرف سلسلة لـــ EnumeratedValue هذه. ويساوي هذا المعرف المعرف في مواصفة TTCN-3؛
- `setVerdict(int verdict)` يضبط العد على enumValue. وإذا لم يكن enumValue قيمة مسموح بها للعد هذا يتم تجاهل العملية.

17.4.2.8 VerdictValue

يتقابل **VerdictValue** مع السطح البيني التالي:

```
// TCI IDL VerdictValue
package org.etsi.ttcn.tci;
public interface VerdictValue {
    public static final int NONE    = 0;
    public static final int PASS    = 1;
    public static final int INCONC  = 2;
    public static final int FAIL    = 3;
    public static final int ERROR   = 4;

    public int      getVerdict() ;
    public void     setVerdict(int verdict) ;
}
```

الطرق:

- `getVerdict()` يعيد قيمة الصحيح لـــ VerdictValue هذا. ويكون integer واحد من الثوابت التالية: ERROR, FAIL, INCONC, NONE, PASS؛
- `setVerdict(int verdict)` يضبط VerdictValue هذا على verdict. ولاحظ أن VerdictValue يمكن ضبطه على أي من الأحكام الواردة أعلاه في أي وقت. ولا يؤدى VerdictValue أي حسابات لحكم كما عرف في TTCN-3. فمثلاً، يعتبر قانونياً ضبط VerdictValue أولاً على ERROR ثم على PASS.

AddressValue 18.4.2.8

يتقابل **AddressValue** مع السطح البيني التالي:

```
// TCI IDL VerdictValue
package org.etsi.ttcn.tci;
public interface AddressValue {
    public int      getAddress() ;
    public void     setAddress(Value value) ;
}
```

الطرق:

- `getAddress()` يعيد القيمة التي تمثلها `AddressValue` هذه؛
- `setAddress(Value value)` تضبط `AddressValue` هذه على `value`.

5.2.8 تقابل أنماط تسجيل مجرد

تعرف أنماط اضافية لتسهيل تسجيل مواءمات بين قيم ومقاسات.

TciValueTemplate 1.5.2.8

يتقابل **TciValueTemplate** مع السطح البيني التالي:

```
// TCI IDL TciValueTemplate
package org.etsi.ttcn.tci;
public interface TciValueTemplate {
    public boolean isOmit();
    public boolean isAny();
    public boolean isAnyOrOmit();
    public String getTemplateDef();
}
```

الطرق:

- `isOmit()` يعيد `true` إذا كان المقاس `omit`، `false` وإلا؛
- `isAny()` يعيد `true` إذا كان المقاس `any`، `false` وإلا؛
- `isAnyOrOmit()` يعيد `true` إذا كان المقاس `anyoromit`، `false` وإلا؛
- `getTemplateDef()` تعيد هذه العملية تعريف المقاس.

TciNonValueTemplate 2.5.2.8

يتقابل **TciNonValueTemplate** مع السطح البيني التالي:

```
// TCI IDL TciNonValueTemplate
package org.etsi.ttcn.tci;
public interface TciNonValueTemplate {
    public boolean isAny();
    public boolean isAll();
    public String getTemplateDef();
}
```

الطرق:

- `isAny()` يعيد `true` إذا كان المقاس `any, false` وإلا؛
- `isAll()` يعيد `true` إذا كان المقاس `all, false` وإلا؛
- `getTemplateDef()` تعيد هذه العملية تعريف المقاس.

TciValueList 3.5.2.8

يتقابل **TciValueList** مع السطح البيني التالي:

```
// TCI IDL TciValueList
package org.etsi.ttcn.tci;
public interface TciValueList{
    public int                      size() ;
    public boolean                  isEmpty() ;
    public TciValue                 get(int index) ;
}
```

الطرق:

- `size()` يعيد عدد القيم في هذه القائمة؛
- `isEmpty()` يعيد `true` إذا لم تحتوي القائمة على قيم؛
- `get(int index)` يعيد `Value` عند الموضع المحدد.

4.5.2.8 TciValueDifference

يتقابل **`TciValueDifference`** مع السطح البيني التالي:

```
// TCI IDL TciValueDifference
package org.etsi.ttcn.tci;
public interface TciValueDifference {
    public Value    getValue();
    public TciValueTemplate getTciValueTemplate();
    public String   getDescription()
}
```

الطرق:

- `getValue()` يعيد قيمة `TciValueDifference` هذه؛
- `getTciValueTemplate()` يعيد مقاس `TciValueDifference` هذا؛
- `getDescription()` يعيد وصف عدم المواءمة.

5.5.2.8 TciValueDifferenceList

يتقابل **`TciValueDifferenceList`** مع السطح البيني التالي:

```
// TCI IDL TciValueDifferenceList
package org.etsi.ttcn.tci;
public interface TciValueDifferenceList{
    public int                      size() ;
    public boolean                  isEmpty() ;
    public TciValueDifference       get(int index) ;
}
```

الطرق:

- `size()` يعيد عدد الفروق في هذه القائمة؛
- `isEmpty()` يعيد `true` إذا لم تحتوي القائمة على فروق؛
- `get(int index)` يعيد `TciValueDifference` عند الموضع المحدد.

3.8 الثوابت

في نطاق تقابل لغة Java، تم تحديد الثوابت. وتعرف جميع الثوابت `public final static` ويمكن النفاذ إليها من كل شيء من كل مجموعة. والثوابت المعرفة في هذا القسم ليست معرفة مع قسم IDL. وبدلاً من ذلك، تنتج من مواصفة أنماط TCI IDL معلمة مثل المحلية.

يمكن استخدام الثوابت لتحديد أسلوب مرور المعلمة لمعلمات TTCN-3 (انظر 2.2.2.8).

- `org.etsi.ttcn.tci.TciParameterPassingMode.TCI_IN`؛
- `org.etsi.ttcn.tci.TciParameterPassingMode.TCI_INOUT`؛
- `org.etsi.ttcn.tri.TciParameterPassingMode.TCI_OUT`.

بالنسبة لقيمة معلمة مميزة `null`، تضبط قيمة معلمة مشفرة على `null` Java.

تستخدم الثوابت التالية لمناولة قيمة (انظر أيضاً 4.2.2.8).

- `org.etsi.ttcn.tci.TciTypeClass.ADDRESS`؛
- `org.etsi.ttcn.tci.TciTypeClass.ANYTYPE`؛
- `org.etsi.ttcn.tci.TciTypeClass.BITSTRING`؛
- `org.etsi.ttcn.tci.TciTypeClass.BOOLEAN`؛
- `org.etsi.ttcn.tci.TciTypeClass.CHARSTRING`؛
- `org.etsi.ttcn.tci.TciTypeClass.COMPONENT`؛
- `org.etsi.ttcn.tci.TciTypeClass.ENUMERATED`؛
- `org.etsi.ttcn.tci.TciTypeClass.FLOAT`؛

- org.etsi.ttcn.tci.TciTypeClass.HEXSTRING؛
- org.etsi.ttcn.tci.TciTypeClass.INTEGER؛
- org.etsi.ttcn.tci.TciTypeClass.OBJID؛
- org.etsi.ttcn.tci.TciTypeClass.OCTETSTRING؛
- org.etsi.ttcn.tci.TciTypeClass.RECORD؛
- org.etsi.ttcn.tci.TciTypeClass.RECORD_OF؛
- org.etsi.ttcn.tci.TciTypeClass.SET؛
- org.etsi.ttcn.tci.TciTypeClass.SET_OF؛
- org.etsi.ttcn.tci.TciTypeClass.UNION؛
- org.etsi.ttcn.tci.TciTypeClass.UNIVERSAL_CHARSTRING؛
- org.etsi.ttcn.tci.TciTypeClass.VERDICT.

تستخدم الثوابت التالية لمناولة مكون (انظر أيضاً 8.2.2.5).

- org.etsi.ttcn.tci.TciTestComponentKind.TCI_CTRL_COMP؛
- org.etsi.ttcn.tci.TciTestComponentKind.TCI_MTC_COMP؛
- org.etsi.ttcn.tci.TciTestComponentKind.TCI_PTC_COMP.
- org.etsi.ttcn.tci.TciTestComponentKind.TCI_SYSTEM_COMP.

4.8 تقابل سطوح بينية

يعرف تعريف TCI IDL أربعة سطوح بينية، TCI-TM، TCI-CH، TCI-CD وTCI-TL. وتعرف العمليات لاتجاهات مختلفة في السطح البيني هذا، أي، يمكن لبعض العمليات أن تطلب فقط من قبل TTCN-3 قابل للتنفيذ أو SA أو PA أو TMC بينما آخري يمكن أن تطلبها TMC على TE. وينعكس هذا بواسطة تقسيم السطوح البينية لـــ TCI IDL إلى سطحين بينيين فرعيين، لكل واحد سابقة Required أو Provided.

الجدول Z.145/3 – سطوح بينية فرعية

TL	PA	SA	CH	CD	TM	TE	Calling/Called
TCI-TL Provided	See ITU-T Rec. Z.144 [1]	See ITU-T Rec. Z.144 [1]	TCI-CH Provided	TCI-CD Provided	TCI-TM Provided	–	**TE**
TCI-TL Provided	–	–	–	–	–	TCI-TM Required	**TM**
TCI-TL Provided	–	–	–	–	–	TCI-CD Required	**CD**
TCI-TL Provided	–	–	–	–	–	TCI-CH Required	**CH**
TCI-TL Provided	–	–	–	–	–	انظر التوصية ITU-T Z.144	**SA**
TCI-TL Provided	–	–	–	–	–	انظر التوصية ITU-T Z.144	**PA**
–	–	–	–	–	–	–	**TL**

ينبغي على جميع الطرق المعرفة في السطوح البينية هذه أن تسلك كما عرف في القسم 7.

1.4.8 السطح البيني TCI-TM

ينقسم السطح البيني **TCI-TM** إلى سطحين بينيين فرعيين، سطح بيني TCI-TM، يعرف نداءات من TE إلى TM، وسطح بيني Required، يعرف نداءات من TM إلى TE.

1.1.4.8 TCI-TM موفرة

يتقابل السطح البيني TCI-TM Provided مع السطح البيني التالي:

```
// TCI-TM
// TE -> TM
package org.etsi.ttcn.tci;
public interface TciTMProvided {
    public void tciTestCaseStarted(TciTestCaseId testCaseId, TciParameterList parameterList, Float
    timer);
    public void     tciTestCaseTerminated( VerdictValue verdict, TciParameterList parameterList);
    public void     tciControlTerminated();
    public Value    tciGetModulePar(TciModuleParameterId parameterId);
    public void     tciLog(TriComponentId testComponentId, String message);
    public void     tciError(String message);
}
```

2.1.4.8 TCI-TM مطلوبة

يتقابل السطح البيني TCI-TM Required مع السطح البيني التالي:

```
// TCI-TM
// TM -> TE
package org.etsi.ttcn.tci;
public interface TciTMRequired {
    public void                     tciRootModule(TciModuleId moduleName) ;
    public TciModuleParameterList   tciGetModuleParameters(TciModuleId moduleId);
    public TciTestCaseIdList        tciGetTestCases();
    public TciParameterTypeList     tciGetTestCaseParameters(TciTestCaseId TestCaseId);
    public TriPortIdList            tciGetTestCaseTSI(TciTestCaseId testCaseId);
    public void                     tciStartTestCase(String testCaseId,
                                    TciParameterList parameterList) ;
    public void                     tciStopTestCase() ;
    public TriComponentId           tciStartControl() ;
    public void                     tciStopControl() ;
}
```

2.4.8 السطح البيني TCI-CD

ينقسم السطح البيني **TCI-CD** إلى سطحين بينيين فرعيين، سطح بيني TCI-CD Provided، يعرف نداءات من TE إلى CD، وسطح بيني TCI-CD Required، يعرف نداءات من CD إلى TE.

1.2.4.8 TCI-CD موفر

يتقابل السطح البيني TCI-CD Provided مع السطح البيني التالي:

```
// TCI-CD
// TE -> CD
package org.etsi.ttcn.tci;
public interface TciCDProvided {
    public Value        decode(TriMessage message, Type decodingHypothesis );
    public TriMessage   encode(Value value);
}
```

2.2.4.8 TCI-CD مطلوب

يتقابل السطح البيني TCI-CD Required مع السطح البيني التالي:

```
// TCI-CD
// CD -> TE
package org.etsi.ttcn.tci;
public interface TciCDRequired {
    public Type     getTypeForName(String typeName);
    public Type     getInteger();
    public Type     getFloat();
    public Type     getBoolean();
    public Type     getObjid();
    public Type     getCharstring();
    public Type     getUniversalCharstring();
    public Type     getHexstring();
    public Type     getBitstring();
    public Type     getOctetstring();
    public Type     getVerdict();
    public void     tciErrorReq(String message);
}
```

3.4.8 السطح البيني TCI-CH

ينقسم السطح البيني **TCI-CH** إلى سطحين بينيين فرعيين، سطح بيني TCI-CH Provided، يعرف نداءات من TE إلى CH، وسطح بيني TCI-CH Required، يعرف نداءات من CH إلى TE.

1.3.4.8 TCI-CH موفرة

يتقابل السطح البيني TCI-CH Provided مع السطح البيني التالي:

```
// TciCHProvided
// TE -> CH
package org.etsi.ttcn.tci;
public interface TciCHProvided {
    public void tciSendConnected    (TriPortId sender, TriComponentId receiver, Value sendMessage) ;
    public void tciSendConnectedBC  (TriPortId sender, Value sendMessage) ;
    public void tciSendConnectedMC  (TriPortId sender, TriComponentIdList receivers,
                                     Value sendMessage) ;

    public void     tciCallConnected(TriPortId sender,
                                     TriComponentId receiver,
                                     TriSignatureId signature,
                                     TciParameterList parameterList) ;

    public void     tciCallConnectedBC(TriPortId sender,
                                       TriSignatureId signature,
                                       TciParameterList parameterList) ;

    public void     tciCallConnectedMC(TriPortId sender,
                                       TriComponentIdList receivers,
                                       TriSignatureId signature,
                                       TciParameterList parameterList) ;

    public void     tciReplyConnected(TriPortId sender,
                                      TriComponentId receiver,
                                      TriSignatureId signature,
                                      TciParameterList parameterList,
                                      Value returnValue) ;
    public void     tciReplyConnectedBC(TriPortId sender,
                                        TriSignatureId signature,
                                        TciParameterList parameterList,
                                        Value returnValue) ;

    public void     tciReplyConnectedMC(TriPortId sender,
                                        TriComponentIdList receivers,
                                        TriSignatureId signature,
                                        TciParameterList parameterList,
                                        Value returnValue) ;

    public void     tciRaiseConnected(TriPortId sender,
                                      TriComponentId receiver,
                                      TriSignatureId signature,
                                      Value except) ;
    public void     tciRaiseConnectedBC(TriPortId sender,
                                        TriSignatureId signature,
                                        alue except) ;

    public void     tciRaiseConnectedMC(TriPortId sender,
                                        TriComponentIdList receivers,
                                        TriSignatureId signature,
                                        Value except) ;

    public TriComponentId   tciCreateTestComponentReq(int kind, Type componentType, String name) ;

    public void     tciStartTestComponentReq(TriComponentId comp,
                                             TciBehaviourId behavior,
                                             TciParameterList parameterList) ;

    public void     tciStopTestComponentReq(TriComponentId comp) ;

    public void     tciConnectReq(TriPortId fromPort, TriPortId toPort) ;

    public void     tciDisconnectReq(TriPortId fromPort, TriPortId toPort) ;

    public void     tciTestComponentTerminatedReq(TriComponentId comp, VerdictValue verdict) ;

    public boolean  tciTestComponentRunningReq(TriComponentId comp) ;

    public TriComponentId   tciGetMTCReq() ;
```

```
    public void     tciMapReq(TriPortId fromPort, TriPortId toPort);

    public void     tciUnmapReq(TriPortId fromPort, TriPortId toPort);

    public void     tciExecuteTestCaseReq(TriComponentId testComponentId,
                                          TriPortIdList tsiPortList);

    public void     tciResetReq() ;

    public boolean  tciTestComponentDoneReq(TriComponentId testComponentId) ;

    public void     tciKillTestComponentReq(TriComponentId component)

    public boolean  tciTestComponentAliveReq(TriComponentId component)

    public boolean  tciTestComponentKilledReq(TriComponentId component)
}
```

2.3.4.8 TCI-CH مطلوبة

يتقابل السطح البيني TCI-CH Required مع السطح البيني التالي:

```
// TciCHRequired
// CH -> TE
package org.etsi.ttcn.tci;
public interface TciCHRequired extends TciCDRequired {
    public void     tciEnqueueMsgConnected(TriPortId sender,
                                           TriComponentId receiver,
                                           Value receivedMessage) ;

    public void     tciEnqueueCallConnected(TriPortId sender,
                                            TriComponentId receiver,
                                            TriSignatureId signature,
                                            TciParameterList parameterList) ;

    public void     tciEnqueueReplyConnected(TriPortId sender,
                                             TriComponentId receiver,
                                             TriSignatureId signature,
                                             TciParameterList parameterList,
                                             Value returnValue) ;

    public void     tciEnqueueRaiseConnected(TriPortId sender,
                                             TriComponentId receiver,
                                             TriSignatureId signature,
                                             Value except) ;

    public TriComponentId   tciCreateTestComponent(int kind, Type componentType, String name) ;

    public void     tciStartTestComponent(TriComponentId comp,
                                          TciBehaviourId behavior,
                                          TciParameterList parameterList) ;

    public void     tciStopTestComponent(TriComponentId comp) ;

    public void     tciConnect(TriPortId fromPort, TriPortId toPort) ;

    public void     tciDisconnect(TriPortId fromPort, TriPortId toPort) ;

    public void     tciTestComponentTerminated(TriComponentId comp, VerdictValue verdict);

    public boolean  tciTestComponentRunning(TriComponentId comp);

    public boolean  tciTestComponentDone(TriComponentId comp);

    public TriComponentId   tciGetMTC();

    public void     tciExecuteTestCase(TciTestCaseId TestCaseId, TriPortIdList tsiPortList);

    public void     tciReset();

    public void     tciMap(TriPortId fromPort, TriPortId toPort);

    public void     tciUnmap(TriPortId fromPort, TriPortId toPort);

    public void     tciKillTestComponent(TriComponentId component)

    public boolean  tciTestComponentAlive(TriComponentId component)

    public boolean  tciTestComponentKilled(TriComponentId component)
}
```

4.4.8 السطح البيني TCI-TL

يحتوي السطح البيني TCI-TL على سطح بيني فرعي واحد فقط، السطح البيني TCI-TL Provided، الذي يعرف نداءات من TE وTM وCH وCD وSA وPA إلى TL.

1.4.4.8 موفر TCI-TL

يتقابل السطح البيني TCI-TL Provided مع السطح البيني التالي:

```
// TCI-TL
// TE,TM,CH,CD,SA,PA -> TL
package org.etsi.ttcn.tci;
public interface TciTLProvided {
    public void tliTcExecute(String am, int ts, String src, int line, TriComponentId c,
            TciTestCaseId tcId, TriParameterList pars, TriTimerDuration dur);
    public void tliTcStart(String am, int ts, String src, int line, TriComponentId c,
            TciTestCaseId tcId, TriParameterList pars, TriTimerDuration dur);
    public void tliTcStop(String am, int ts, String src, int line, TriComponentId c);
    public void tliTcStarted(String am, int ts, String src, int line, TriComponentId c,
            TciTestCaseId tcId, TriParameterList pars, TriTimerDuration dur);
    public void tliTcTerminated(String am, int ts, String src, int line, TriComponentId c,
            TciTestCaseId tcId, TriParameterList pars, VerdictValue outcome);
    public void tliCtrlStart(String am, int ts, String src, int line, TriComponentId c);
    public void tliCtrlStop(String am, int ts, String src, int line, TriComponentId c);
    public void tliCtrlTerminated(String am, int ts, String src, int line, TriComponentId c);
    public void tliMSend_m(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TriAddress address,
            TriStatus encoderFailure, TriMessage msg, TriStatus transmissionFailure);
    public void tliMSend_m_BC(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue
            TriStatus encoderFailure, TriMessage msg, TriStatus transmissionFailure);
    public void tliMSend_m_MC(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TriAddressList addresses,
            TriStatus encoderFailure, TriMessage msg, TriStatus transmissionFailure);
    public void tliMSend_c(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TriComponentId to, TriStatus transmissionFailure);
    public void tliMSend_c_BC(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TriStatus transmissionFailure);
    public void tliMSend_c_MC(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TriComponentIdList toList,
            TriStatus transmissionFailure);
    public void tliMDetected_m(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, TriMessage msg, TriAddress address);
    public void tliMDetected_c(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TriComponentId from);
    public void tliMMismatch_m(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TciValueDifferenceList diffs,
            TriAddress address, TciValueTemplate addressTmpl);
    public void tliMMismatch_c(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TciValueDifferenceList diffs,
            TriComponentId from, TciNonValueTemplate fromTmpl);
    public void tliMReceive_m(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TciValueTemplate msgTmpl,
            TriAddress address, TciValueTemplate addressTmpl);
    public void tliMReceive_c(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, Value msgValue, TciValueTemplate msgTmpl,
            TriComponentId from, TciNonValueTemplate fromTmpl);
    public void tliPrCall_m(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, TriSignatureId signature, TciParameterList parsValue,
            TriAddress address, TriStatus encoderFailure, TriParameterList pars,
            TriStatus transmissionFailure);
    public void tliPrCall_m_BC(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, TriSignatureId signature, TciParameterList parsValue,
            TriStatus encoderFailure, TriParameterList pars,
            TriStatus transmissionFailure);
    public void tliPrCall_m_MC(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, TriSignatureId signature, TciParameterList parsValue,
            TriAddressList addresses, TriStatus encoderFailure, TriParameterList pars,
            TriStatus transmissionFailure);
    public void tliPrCall_c(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, TriSignatureId signature, TciParameterList parsValue, TriComponentId to,
            TriStatus transmissionFailure);
    public void tliPrCall_c_BC(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, TriSignatureId signature, TciParameterList parsValue,
            TriStatus transmissionFailure);
    public void tliPrCall_c_MC(String am, int ts, String src, int line, TriComponentId c,
            TriPortId port, TriSignatureId signature, TciParameterList parsValue,
            TriComponentIdList toList, TriStatus transmissionFailure);
```

```
public void tliPrGetCallDetected_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TriParameterList pars, TriAddress address);
public void tliPrGetCallDetected_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        TriComponentId from);
public void tliPrGetCallMismatch_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature,
        TciParameterList parsValue, TciValueTemplate parsTmpl, TciValueDifferenceList diffs,
        TriAddress address, TciValueTemplate addressTmpl);
public void tliPrGetCallMismatch_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature,
        TciParameterList parsValue, TciValueTemplate parsTmpl, TciValueDifferenceList diffs,
        TriComponentId from, TciNonValueTemplate fromTmpl);
public void tliPrGetCall_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature,
        TciParameterList parsValue, TciValueTemplate parsTmpl,
        TriAddress address, TciValueTemplate addressTmpl);
public void tliPrGetCall_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature,
        TciParameterList parsValue, TciValueTemplate parsTmpl,
        TriComponentId from, TciNonValueTemplate fromTmpl);
public void tliPrReply_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
        TriAddress address, TriStatus encoderFailure,
        TriParameter repl, TriStatus transmissionFailure);
public void tliPrReply_m_BC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
        TriStatus encoderFailure,
        TriParameter repl, TriStatus transmissionFailure);
public void tliPrReply_m_MC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
        TriAddressList addresses, TriStatus encoderFailure,
        TriParameter repl, TriStatus transmissionFailure);
public void tliPrReply_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
        TriComponentId to, TriStatus transmissionFailure);
public void tliPrReply_c_BC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
        TriStatus transmissionFailure);
public void tliPrReply_c_MC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
        TriComponentIdList toList, TriStatus transmissionFailure);
public void tliPrGetReplyDetected_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TriParameter repl, TriAddress address);
public void tliPrGetReplyDetected_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, Value replValue, TriComponentId from);
public void tliPrGetReplyMismatch_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value replValue, TciValueTemplate replyTmpl,  TciValueDifferenceList diffs,
        TriAddress address, TciValueTemplate addressTmpl);
public void tliPrGetReplyMismatch_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value replValue, TciValueTemplate replyTmpl,  TciValueDifferenceList diffs,
        TriComponentId from, TciNonValueTemplate fromTmpl);
public void tliPrGetReply_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value replValue, TciValueTemplate replyTmpl,
        TriAddress address, TciValueTemplate addressTmpl);
public void tliPrGetReply_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value replValue, TciValueTemplate replyTmpl,
        TriComponentId from, TciNonValueTemplate fromTmpl);
public void tliPrRaise_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriAddress address, TriStatus encoderFailure, TriException exc,
        TriStatus transmissionFailure);
public void tliPrRaise_m_BC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriStatus encoderFailure, TriException exc, TriStatus transmissionFailure);
public void tliPrRaise_m_MC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriAddressList addresses, TriStatus encoderFailure, TriException exc,
        TriStatus transmissionFailure);
public void tliPrRaise_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriComponentId to, TriStatus transmissionFailure);
public void tliPrRaise_c_BC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriStatus transmissionFailure);
public void tliPrRaise_c_MC(String am, int ts, String src, int line, TriComponentId c,
```

```
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriComponentIdList toList, TriStatus transmissionFailure);
public void tliPrCatchDetected_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TriException exc, TriAddress address);
public void tliPrCatchDetected_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, Value excValue, TriComponentId from);
public void tliPrCatchMismatch_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value excValue, TciValueTemplate excTmpl,  TciValueDifferenceList diffs,
        TriAddress address, TciValueTemplate addressTmpl);
public void tliPrCatchMismatch_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value excValue, TciValueTemplate excTmpl,  TciValueDifferenceList diffs,
        TriComponentId from, TciNonValueTemplate fromTmpl);
public void tliPrCatch_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value excValue, TciValueTemplate excTmpl,
        TriAddress address, TciValueTemplate addressTmpl);
public void tliPrCatch_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value excValue, TciValueTemplate excTmpl,
        TriComponentId from, TciNonValueTemplate fromTmpl);
public void tliPrCatchTimeoutDetected(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature);
public void tliPrCatchTimeout(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue);
public void tliCCreate(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp, String name);
public void tliCStart(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp, TciBehaviourId name, TciParameterList parsValue);
public void tliCRunning(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp, TBoolean status);
public void tliCAlive(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp, TBoolean status);
public void tliCStop(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp);
public void tliCKill(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp);
public void tliCDoneMismatch(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate compTmpl);
public void tliCDone(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate compTmpl);
public void tliCKilledMismatch(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate compTmpl);
public void tliCKilled(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate compTmpl);
public void tliCTerminated(String am, int ts, String src, int line, TriComponentId c,
        VerdictValue verdict);
public void tliPConnect(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2);
public void tliPDisconnect(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2);
public void tliPMap(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2);
public void tliPUnmap(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2);
public void tliPClear(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port);
public void tliPStart(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port);
public void tliPStop(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port);
public void tliPHalt(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port);
public void tliEncode(String am, int ts, String src, int line, TriComponentId c,
        Value val, TriStatus encoderFailure, TriMessage msg, String codec);
public void tliDecode(String am, int ts, String src, int line, TriComponentId c,
        TriMessage msg, TriStatus decoderFailure, Value val, String codec);
public void tliTTimeoutDetected(String am, int ts, String src, int line, TriComponentId c,
        TriTimerId timer);
public void tliTTimeoutMismatch(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate timerTmpl);
public void tliTTimeout(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate timerTmpl);
public void tliTStart(String am, int ts, String src, int line, TriComponentId c,
        TriTimerId timer, TriTimerDuration dur);
public void tliTStop(String am, int ts, String src, int line, TriComponentId c,
        TriTimerId timer);
public void tliTRead(String am, int ts, String src, int line, TriComponentId c,
        TriTimerId timer, TriTimerDuration elapsed);
```

```
    public void tliTRunning(String am, int ts, String src, int line, TriComponentId c,
            TriTimerId timer, TBoolean status);
    public void tliSEnter(String am, int ts, String src, int line, TriComponentId c,
            String name, TciParameterList parsValue, String kind);
    public void tliSLeave(String am, int ts, String src, int line, TriComponentId c,
            String name, Value returnValue, String kind);
    public void tliVar(String am, int ts, String src, int line, TriComponentId c,
            String name, Value varValue);
    public void tliModulePar(String am, int ts, String src, int line, TriComponentId c,
            String name, Value parValue);
    public void tliGetVerdict(String am, int ts, String src, int line, TriComponentId c,
            VerdictValue verdict);
    public void tliSetVerdict(String am, int ts, String src, int line, TriComponentId c,
            VerdictValue verdict);
    public void tliLog(String am, int ts, String src, int line, TriComponentId c,
            TciValueList log);
    public void tliAEnter(String am, int ts, String src, int line, TriComponentId c);
    public void tliALeave(String am, int ts, String src, int line, TriComponentId c);
    public void tliADefaults(String am, int ts, String src, int line, TriComponentId c);
    public void tliAActivate(String am, int ts, String src, int line, TriComponentId c,
            String name, TciParameterList pars, Value ref);
    public void tliADeactivate(String am, int ts, String src, int line, TriComponentId c,
            Value ref);
    public void tliANomatch(String am, int ts, String src, int line, TriComponentId c);
    public void tliARepeat(String am, int ts, String src, int line, TriComponentId c);
    public void tliAWait(String am, int ts, String src, int line, TriComponentId c);
}
```

5.8 معلمات خيارية

يعرف القسم 7–3 أن قيمة محجوزة تستخدم لتدل على غياب معلمة خيارية. ولتقابل لغة Java، تستخدم قيمة Java `null` لتدل على غياب قيمة خيارية. فمثلاً، إذا كانت معلمة قيمة في عملية `tciReplyConnected` محذوفة، يكون تنفيذ العملية هو `tciReplyConnected(sender, receiver, signature, parameterList, null)`.

6.8 تدميث TCI

إن جميع الطرق غير سكونية، أي، يمكن طلب العمليات على أشياء فقط. ونظرا لأن هذه التوصية لا تعرف استراتيجيات تنفيذ محسوسة لـــ TE وTM وCD وCH، فإن آلية كيفية معرفة تناول TE أو TM أو CD أو CH الأشياء المعنية هي خارج مدى هذه التوصية.

يوفر بائعو الأدوات الطرق إلى المطورين لـــ TM وCD وCH لتسجيل TE وTM وCD وCH لشركاء الاتصالات المعنيين.

7.8 مناولة خطأ

إن جميع العمليات المطلوبة من TM أو CH أو CD التي تعاد هي ناجحة. وإذا تعرف TE على حالة خاطئة، يرسل خطأ اختبار مجرد إلى المستعمل باستخدام الإجراءات المعرفة في 5.2.1.3.7 (`tciError`). إذا أدت عملية طلبها TE في TM أو CH أو CD أو TL إلى خطأ، ينبغي إرسال الحالة الخاطئة هذه إلى TE باستخدام الإجراءات المعرفة في 12.1.2.3.7 (`tciErrorReq`).

وبجانب مناولة الخطأ هذا، لا تعرف مناولة خطأ مع تقابل لغة Java. وخاصة، لم تعرف آليات مناولة استثنائية.

9 تقابل لغة ANSI-C

1.9 مقدمة

يعرف هذا القسم تقابل لغة TRI ANSI-C لمعطيات TCI المحددة في 7–2 ولعمليات TCI المحددة في 7–3.

2.9 السطوح البينية مع قيمة

ملاحظات وتعليقات	ANSI-C representation	TCI IDL Interface
Type		
	`TciModuleIdType tciGetDefiningModule(TciType inst)`	TciModuleIdType getDefiningModule()
نمط سلسلة معاد استخدامه من IDL (التوصية OMG)	`String tciGetName(TciType inst)`	TString getName()
	`TciTypeClassType tciGetTypeClass(TciType inst)`	TciTypeClassType getTypeClass()
	`TciValue tciNewInstance(TciType inst)`	Value newInstance()
	`String tciGetTypeEncoding(TciType inst)`	TString getTypeEncoding()
	`String* getTypeExtension(TciType inst)`	TStringSeq getTypeExtension()
	`String tciGetTypeEncodingVariant(TciType inst)`	TString getTypeEncodingVariant()
Value		
	`String tciGetValueEncoding(TciValue inst)`	TString getValueEncoding()
	`String tciGetValueEncodingVariant(TciValue inst)`	TString getValueEncodingVariant()
	`TciType tciGetType(TciValue inst)`	Type getType()
نمط بولاني معاد استخدامه من IDL (التوصية OMG)	`Boolean tciNotPresent(TciValue inst)`	TBoolean notPresent()
IntegerValue		
يعيد قيمة مطلقة لصحيح -10 (base) باعتبارها سلسلة ASCII	`String tciGetIntAbs(TciValue inst)`	TInteger getInt()
يعيد عدد الأرقام في قيمة صحيح	`unsigned long int tciGetIntNumberOfDigits (TciValue inst)`	
يعيد true إذا كان الرقم موجبا، وإلا false	`Boolean tciGetIntSign(TciValue inst)`	
يعيد قيمة الرقم عند موضع "position" حيث موضع 0 هو أقل رقم أهمية.	`char tciGetIntDigit (TciValue inst, unsigned long int position)`	
يضبط قيمة مطلقة لصحيح (10-base) عبر سلسلة ASCII. ولا ينبغي أن يكون الرقم الأول 0 ما لم تكون القيمة هي 0.	`void tciSetIntAbs(TciValue inst, String value)`	void setInt(in TInteger value)
يضبط عدد الأرقام في قيمة صحيح	`void tciSetIntNumberOfDigits (TciValue inst, unsigned long int dig_num)`	
يضبط العلامة + (true) أو – (false)	`void tciSetIntSign (TciValue inst, Boolean sign)`	
يضبط قيمة الرقم عند موضع "position" حيث موضع 0 هو أقل رقم أهمية	`void tciSetIntDigit (TciValue inst, unsigned long int position, char digit)`	
FloatValue		
	`double tciGetFloatValue(TciValue inst)`	TFloat getFloat()
	`void tciSetFloatValue(TciValue inst, double value)`	void setFloat (in TFloat value)
BooleanValue		
	`Boolean tciGetBooleanValue(TciValue inst)`	TBoolean getBoolean()
	`void tciSetBooleanValue (TciValue inst, Boolean value)`	void setBoolean (in TBoolean value)

ملاحظات وتعليقات	ANSI-C representation	TCI IDL Interface
	ObjidValue	
	`TciObjidValue tciGetTciObjidValue(TciValue inst)`	TObjid getObjid()
	`void tciSetObjidValue(TciValue inst, TciObjidValue value)`	void setObjid(in TObjid value)
	CharstringValue	
	`TciCharStringValue tciGetCStringValue(TciValue inst)`	TString getString()
	`void tciSetCStringValue (TciValue inst, TciCharString value)`	void setString(in TString value)
	`char tciGetCStringCharValue (TciValue inst, long int position)`	TChar getChar (in TInteger position)
	`void tciSetCStringCharValue (TciValue inst, long int position, char value)`	void setChar (in TInteger position, in char value)
	`unsigned long int tciGetCStringLength (TciValue inst)`	TInteger getLength()
	`void tciSetCStringLength (TciValue inst, unsigned long int len)`	void setLength(in TInteger len)
	UniversalCharstringValue	
	`TciUCStringValuetciGetUCStringValue(TciValue inst)`	TString getString()
	`void tciSetUCStringValue (TciValue inst, TciUCStringValue value)`	void setString (in TString value)
	`void tciGetUCStringCharValue (TciValue inst, unsigned long int position, TciUCValue result)`	TUniversalChar getChar (in TInteger position)
	`void tciSetUCStringCharValue (TciValue inst, unsigned long int position, TciUCValue value)`	void setChar (in TInteger position, in TUniversalChar value)
	`unsigned long int tciGetUCStringLength(TciValue inst)`	TInteger getLength()
	`void tciSetUCStringLength (TciValue inst, unsigned long int len)`	void setLength (in TInteger len)
	BitstringValue	
	`String tciGetBStringValue(TciValue inst)`	TString getString()
	`void tciSetBStringValue (TciValue inst, String value)`	void setString(in TString value)
	`int tciGetBStringBitValue (TciValue inst, long int position)`	TChar getBit (in integer position)
	`void tciSetBStringBitValue (TciValue inst, unsigned long int position, int value)`	void setBit (in TInteger position, in TInteger value)
	`unsigned long int tciGetBStringLength(TciValue inst)`	TInteger getLength()
	`void tciSetBStringLength (TciValue inst, long int len)`	void setLength (in TInteger len)
	HexstringValue	
	`String tciGetHStringValue(TciValue inst)`	TString getString()
	`void tciSetHStringValue (TciValue inst, String value)`	void setString (in TString value)

ملاحظات وتعليقات	ANSI-C representation	TCI IDL Interface
	`int tciGetHStringHexValue (TciValue inst, unsigned long int position)`	TChar getHex (in TInteger position)
	`void tciSetHStringHexValue (TciValue inst, unsigned long int position, int value)`	void setBit (in TInteger position, in TInteger value)
	`long int tciGetHStringLength(TciValue inst)`	TInteger getLength()
	`void tciSetHStringLength (TciValue inst, unsigned long int len)`	void setLength(in TInteger len)
OctetstringValue		
	`String tciGetOStringValue(TciValue inst)`	TString getString()
	`void tciSetOStringValue (TciValue inst, String value)`	void setString(in TString value)
	`int tciGetOStringOctetValue (TciValue inst, unsigned long int position)`	TChar getOctet(in TInteger position)
	`void tciSetOStringOctetValue (TciValue inst, unsigned long int position, int value)`	void setOctet (in TInteger position, in TInteger value)
	`unsigned long int tciGetOStringLength(TciValue inst)`	TInteger getLength()
	`void tciSetOStringLength (TciValue inst, unsigned long int len)`	void setLength(in TInteger len)
RecordValue		
	`TciValue tciGetRecFieldValue (TciValue inst, String fieldName)`	Value getField (in TString fieldName)
	`void tciSetRecFieldValue (TciValue inst, String fieldName, TciValue value)`	void setField (in TString fieldName, in Value value)
يعيد صفيف لأسماء مجال أنهاها NULL	`char** tciGetRecFieldNames(TciValue inst)`	TString[] getFieldNames()
RecordOfValue		
	`TciValue tciGetRecOfFieldValue (TciValue inst, unsigned long int position)`	Value getField (in TInteger position)
	`void tciSetRecOfFieldValue (TciValue inst, unsigned long int position, TciValue value)`	void setField (in TInteger position, in Value value)
	`void tciAppendRecOfFieldValue (TciValue inst, TciValue value)`	void appendField (in Value value)
	`TciType tciGetRecOfElementType(TciValue inst)`	Type getElementType()
	`unsigned long int tciGetRecOfLength(TciValue inst)`	TInteger getLength()
	`void tciSetRecOfLength (TciValue inst, unsigned long int len)`	void setLength (in TInteger len)
UnionValue		
	`TciValue tciGetUnionVariant (TciValue inst, String variantName)`	Value getVariant (in TString variantName)

ملاحظات وتعليقات	ANSI-C representation	TCI IDL Interface
	`void tciSetUnionVariant (TciValue inst, String variantName, TciValue value)`	void setVariant (in TString variantName, in Value value)
	`String tciGetUnionPresentVariantName (TciValue inst)`	TString getPresentVariantName()
يعيد صفيف لأسماء مجال ألفاها NULL	`char** tciGetUnionVariantNames(TciValue inst)`	TString[] getVariantNames()
EnumeratedValue		
	`String tciGetEnumValue(TciValue inst)`	TString getEnum()
	`void tciSetEnumValue (TciValue inst, String enumValue)`	void setEnum (in TString enumValue)
VerdictValue		
	`int tciGetVerdictValue(TciValue inst)`	TInteger getVerdict()
	`void tciSetVerdictValue(TciValue inst, int verdict)`	void setVerdict (in TInteger verdict)
		`AddressValue`
	`TciValue tciGetAddressValue(TciValue inst)`	Value getAddress()
	`void tciSetAddressValue(TciValue inst, TciValue value)`	void setAddress (in Value value)

3.9 سطح بيني مع تسجيل

Notes and comments	ANSI-C representation	TCI IDL Interface
TciValueTemplate		
نمط بولاني معاد استخدامه من IDL (التوصية OMG)	`Boolean tciIsOmit(TciValueTemplate inst)`	TBoolean isOmit()
نمط بولاني معاد استخدامه من IDL (التوصية OMG)	`Boolean tciIsAny(TciValueTemplate inst)`	TBoolean isAny()
نمط بولاني معاد استخدامه من IDL (التوصية OMG)	`Boolean tciIsAnyOrOmit(TciValueTemplate inst)`	TBoolean isAnyOrOmit()
نمط سلسلة معاد استخدامه من IDL (توصية OMG)	`String tciGetTemplateDef(TciValueTemplate inst)`	TString getTemplateDef()
TciNonValueTemplate		
نمط بولاني معاد استخدامه من IDL (التوصية OMG)	`Boolean tciIsAny(TciNonValueTemplate inst)`	TBoolean isAny()
نمط بولاني معاد استخدامه من IDL (التوصية OMG)	`Boolean tciIsAll(TciNonValueTemplate inst)`	TBoolean isAll()
نمط سلسلة معاد استخدامه من IDL (التوصية OMG)	`String tciGetTemplateDef(TciNonValueTemplate inst)`	TString getTemplateDef()

4.9 سطوح بينية مع عملية

TCI-TM required		
	`void tciRootModule(String moduleId)`	void tciRootModule (in TciModuleIdType moduleId)
	`TciModuleParameterListType tciGetModuleParameters (TciModuleIdType moduleName)`	TciModuleParameterListType tciGetModuleParameters (in TciModuleIdType moduleName)
	`TciTestCaseIdListType tciGetTestCases()`	TciTestCaseIdListType tciGetTestCases()

	`TciParameterTypeListType tciGetTestCaseParameters (TciTestCaseIdType testCaseId)`	TciParameterTypeListType tciGetTestCaseParameters (in TciTestCaseIdType testCaseId)
	`TriPortIdList tciGetTestCaseTSI (TciTestCaseIdType testCaseId)`	TriPortIdListType tciGetTestCaseTSI (in TciTestCaseIdType testCaseId)
	`void tciStartTestCase (TciTestCaseIdType testCaseId, TciParameterListType parameterlist)`	void tciStartTestCase (in TciTestCaseIdType testCaseId, in TciParameterListType parameterlist)
	`void tciStopTestCase()`	void tciStopTestCase()
	`TriComponentId tciStartControl()`	TriComponentId tciStartControl()
	`void tciStopControl()`	void tciStopControl()
TCI-TM provided		
	`void tciTestCaseStarted (TciTestCaseIdType testCaseId, TciParameterListType parameterList, double timer)`	void tciTestCaseStarted (in TciTestCaseIdType testCaseId, in TciParameterListType parameterList, in Tfloat timer)
	`void tciTestCaseTerminated (TciVerdictValue verdict, TciParameterListType parameterlist)`	void tciTestCaseTerminated (in VerdictValue verdict, in TciParameterListType parameterlist)
	`void tciControlTerminated()`	void tciControlTerminated()
	`tciValue tciGetModulePar (TciModuleParameterIdType parameterId)`	Value tciGetModulePar (in TciModuleParameterIdType parameterId)
	`void tciLog(String message)`	void tciLog(in TString message)
	`void tciError(String message)`	void tciError(in TString message)
TCI-CD required		
	`TciType tciGetTypeForName (String typeName)`	Type getTypeForName (in String typeName)
	`TciType tciGetIntegerType()`	Type getInteger()
	`TciType tciGetFloatType()`	Type getFloat()
	`TciType tciGetBooleanType()`	Type getBoolean()
	`TciType tciGetCharType()`	Type getChar()
	`TciType tciGetUniversalCharType()`	Type getUniversalChar()
	`TciType tciGetTciObjidType()`	Type getObjid()
	`TciType tciGetTciCharstringType()`	Type getCharstring()
	`TciType tciGetUniversalCharstringType()`	Type getUniversalCharstring()
	`TciType tciGetHexstringType()`	Type getHexstring()
	`TciType tciGetBitstringType()`	Type getBitstring()
	`TciType tciGetOctetstringType()`	Type getOctetstring()
	`TciType tciGetVerdictType()`	Type getVerdict()
	`void tciErrorReq(String message)`	void tciErrorReq(in String message)
TCI-CD provided		
يعاد استخدام BinaryString من TRI	`TciValue tciDecode (BinaryString message, TciType decHypothesis)`	Value decode (in TriMessageType message, in Type decodingHypothesis)
	`BinaryString tciEncode(TciValue value)`	TriMessageType encode (in Value value)
TCI-CH required		
	`void tciEnqueueMsgConnected (TriPortId sender, TriComponentId receiver, TciValue rcvdMessage)`	void tciEnqueueMsgConnected (in TriPortIdType sender, in TriComponentIdType receiver, in Value rcvdMessage)

	void tciEnqueueCallConnected (TriPortId sender, TriComponentId receiver, TriSignatureId signature, TciParameterListType parameterList)	void tciEnqueueCallConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList)
	void tciEnqueueReplyConnected (TriPortId sender, TriComponentId receiver, TriSignatureId signature, TciParameterListType parameterList, TciValue returnValue)	void tciEnqueueReplyConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue)
	void tciEnqueueRaiseConnected (TriPortId sender, TriComponentId receiver, TriSignatureIdType signature, TciValue exception)	void tciEnqueueRaiseConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in Value exception)
	TriComponentId tciCreateTestComponent (TciTestComponentKindType kind, TciType componentType, String name)	TriComponentIdType tciCreateTestComponent (in TciTestComponentKindType kind in Type componentType, in TString name)
	void tciStartTestComponent (TriComponentId component, TciBehaviourIdType behavior, TciParamaterListType parameterList)	void tciStartTestComponent (in TriComponentIdType component, in TciBehaviourIdType behavior, in TciParamaterListType parameterList)
	void tciStopTestComponent (TriComponentId component)	void tciStopTestComponent (in TriComponentIdType component)
	void tciConnect (TriPortId fromPort, TriPortId toPort)	void tciConnect (in TriPortIdType fromPort, in TriPortIdType toPort)
	void tciDisconnect (TriPortId fromPort, TriPortId toPort)	void tciDisconnect (in TriPortIdType fromPort, in TriPortIdType toPort)
	void tciMap (TriPortId fromPort, TriPortId toPort)	void tciMap (in TriPortIdType fromPort, in TriPortIdType toPort)
	void tciUnmap (TriPortId fromPort, TriPortId toPort)	void tciUnmap (in TriPortIdType fromPort, in TriPortIdType toPort)
	void tciTestComponentTerminated (TriComponentId component, TciVerdictValue verdict)	void tciTestComponentTerminated (in TriComponentIdType component, in VerdictValue verdict)
	Boolean tciTestComponentRunning (TriComponentId component)	boolean tciTestComponentRunning (in TriComponentIdType component)
	boolean tciTestComponentDone (TriComponentId component)	boolean tciTestComponentDone (in TriComponentIdType component)
	TriComponentId tciGetMTC()	TriComponentIdType tciGetMTC()
	void tciExecuteTestCase (TciTestCaseIdType testCaseId, TriPortIdList tsiPortList)	void tciExecuteTestCase (in TciTestCaseIdType testCaseId, in TriPortIdListType tsiPortList)
	void tciReset()	void tciReset()
	void tciKillTestComponent (TriComponentId component)	void tciKillTestComponent (in TriComponentIdType component)
	Boolean tciTestComponentAlive (TriComponentId component)	TBoolean tciTestComponentAlive (in TriComponentIdType component)
	Boolean tciTestComponentKilled (TriComponentId component)	TBoolean tciTestComponentKilled (in TriComponentIdType component)
TCI-CH provided		
	void tciSendConnected (TriPortId sender, TriComponentId receiver, TciValue sendMessage)	void tciSendConnected (in TriPortIdType sender, in TriComponentIdType receiver, in Value sendMessage)

	void tciSendConnectedBC (TriPortId sender, TciValue sendMessage)	void tciSendConnectedBC (in TriPortIdType sender, in Value sendMessage)
	void tciSendConnectedMC (TriPortId sender, TriComponentIdList receivers, TciValue sendMessage)	void tciSendConnectedMC (in TriPortIdType sender, in TriComponentIdListType receivers, in Value sendMessage)
	void tciCallConnected (TriPortId sender, TriComponentId receiver, TriSignatureId signature, TciParameterListType parameterList)	void tciCallConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList)
	void tciCallConnectedBC (TriPortId sender, TriSignatureId signature, TciParameterListType parameterList)	void tciCallConnectedBC (in TriPortIdType sender, in TriSignatureIdType signature, in TciParameterListType parameterList)
	void tciCallConnectedMC (TriPortId sender, TriComponentIdList receivers, TriSignatureId signature, TciParameterListType parameterList)	void tciCallConnectedMC (in TriPortIdType sender, in TriComponentIdListType receivers, in TriSignatureIdType signature, in TciParameterListType parameterList)
	void tciReplyConnected (TriPortId sender, TriComponentId receiver, TriSignatureIdType signature, TciParameterListType parameterList, TciValue returnValue)	void tciReplyConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue)
	void tciReplyConnectedBC (TriPortId sender, TriSignatureIdType signature, TciParameterListType parameterList, TciValue returnValue)	void tciReplyConnectedBC (in TriPortIdType sender, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue)
	void tciReplyConnectedMC (TriPortId sender, TriComponentIdList receivers, TriSignatureIdType signature, TciParameterListType parameterList, TciValue returnValue)	void tciReplyConnectedMC (in TriPortIdType sender, in TriComponentIdListType receivers, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue)
	void tciRaiseConnected (TriPortId sender, TriComponentId receiver, TriSignatureId signature, TciValue exception)	void tciRaiseConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in Value exception)
	void tciRaiseConnectedBC (TriPortId sender, TriSignatureId signature, TciValue exception)	void tciRaiseConnectedBC (in TriPortIdType sender, in TriSignatureIdType signature, in Value exception)
	void tciRaiseConnectedMC (TriPortId sender, TriComponentIdList receivers, TriSignatureId signature, TciValue exception)	void tciRaiseConnectedMC (in TriPortIdType sender, in TriComponentIdListType receivers, in TriSignatureIdType signature, in Value exception)
	TriComponentId tciCreateTestComponentReq (TciTestComponentKindType kind, TciType componentType, String name)	TriComponentIdType tciCreateTestComponentReq (in TciTestComponentKindType kind, in Type componentType, in TString name)
	void tciStartTestComponentReq (TriComponentId component, TciBehaviourIdType behavior, TciParamaterListType parameterList)	void tciStartTestComponentReq (in TriComponentIdType component, in TciBehaviourIdType behavior, in TciParamaterListType parameterList)
	void tciStopTestComponentReq (TriComponentId component)	void tciStopTestComponentReq (in TriComponentIdType component)

	void tciConnectReq (TriPortId fromPort, TriPortId toPort)	void tciConnectReq (in TriPortIdType fromPort, in TriPortIdType toPort)
	void tciDisconnectReq (TriPortId fromPort, TriPortId toPort)	void tciDisconnectReq (in TriPortIdType fromPort, in TriPortIdType toPort)
	void tciMapReq (TriPortId fromPort, TriPortId toPort)	void tciMapReq (in TriPortIdType fromPort, in TriPortIdType toPort)
	void tciUnmapReq (TriPortId fromPort, TriPortId toPort)	void tciUnmapReq (in TriPortIdType fromPort, in TriPortIdType toPort)
	void tciTestComponentTerminatedReq (TriComponentId component, TciVerdictValue verdict)	void tciTestComponentTerminatedReq (in TriComponentIdType component, in VerdictValue verdict)
	Boolean tciTestComponentRunningReq (TriComponentId component)	boolean tciTestComponentRunningReq (in TriComponentIdType component)
	Boolean tciTestComponentDoneReq (TriComponentId component)	boolean tciTestComponentDoneReq (in TriComponentIdType component)
	TriComponentId tciGetMTCReq()	TriComponentIdType tciGetMTCReq()
	void tciExecuteTestCaseReq (TciTestCaseIdType testCaseId, TriPortIdList tsiPortList)	void tciExecuteTestCaseReq (in TciTestCaseIdType testCaseId, in TriPortIdListType tsiPortList)
	void tciResetReq()	void tciResetReq()
	void tciKillTestComponentReq (TriComponentId component)	void tciKillTestComponentReq (in TriComponentIdType component)
	TBoolean tciTestComponentAliveReq (TriComponentId component)	TBoolean tciTestComponentAliveReq (in TriComponentIdType component)
	TBoolean tciTestComponentKilledReq (TriComponentId component)	TBoolean tciTestComponentKilledReq (in TriComponentIdType component)
TCI-TL provided		
	void tliTcExecute (String am, int ts, String src, int line, TriComponentId c, TciTestCaseIdType tcId, TriParameterList pars, TriTimerDuration dur)	void tliTcExecute (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in TriTimerDurationType dur)
	void tliTcStart (String am, int ts, String src, int line, TriComponentId c, TciTestCaseIdType tcId, TriParameterList pars, TriTimerDuration dur)	void tliTcStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in TriTimerDurationType dur)
	void tliTcStop (String am, int ts, String src, int line, TriComponentId c)	void tliTcStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)
	void tliTcStarted (String am, int ts, String src, int line, TriComponentId c, TciTestCaseIdType tcId, TriParameterList pars, TriTimerDuration dur)	void tliTcStarted (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in TriTimerDurationType dur)
	void tliTcTerminated (String am, int ts, String src, int line, TriComponentId c, TciTestCaseIdType tcId, TriParameterList pars, VerdictValue outcome)	void tliTcTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in VerdictValue outcome)

	void tliCtrlStart (String am, int ts, String src, int line, TriComponentId c)	void tliCtrlStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)
	void tliCtrlStop (String am, int ts, String src, int line, TriComponentId c)	void tliCtrlStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)
	void tliCtrlTerminated (String am, int ts, String src, int line, TriComponentId c)	void tliCtrlTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)
	void tliMSend_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriAddress address, TriStatus encoderFailure, TriMessage msg, TriStatus transmissionFailure)	void tliMSend_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriAddressType address, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure)
	void tliMSend_m_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriStatus encoderFailure, TriMessage msg, TriStatus transmissionFailure)	void tliMSend_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure)
	void tliMSend_m_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriAddressList addresses, TriStatus encoderFailure, TriMessage msg, TriStatus transmissionFailure)	void tliMSend_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure)
	void tliMSend_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriComponentId to, TriStatus transmissionFailure)	void tliMSend_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdType to, in TriStatusType transmissionFailure)
	void tliMSend_c_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriStatus transmissionFailure)	void tliMSend_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriStatusType transmissionFailure)
	void tliMSend_c_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriComponentIdList toList, TriStatus transmissionFailure)	void tliMSend_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)
	void tliMDetected_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriMessage msg, TriAddress address)	void tliMDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriMessageType msg, in TriAddressType address)
	void tliMDetected_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriComponentId from)	void tliMDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdType from)

	void tliMMismatch_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TciValueDifferenceList diffs, TriAddress address, TciValueTemplate addressTmpl)	void tliMMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)
	void tliMMismatch_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TciValueDifferenceList diffs, TriComponentId from, TciNonValueTemplate fromTmpl)	void tliMMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)
	void tliMReceive_m(String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TriAddress address, TciValueTemplate addressTmpl)	void tliMReceive_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)
	void tliMReceive_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TriComponentId from, TciNonValueTemplate fromTmpl)	void tliMReceive_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)
	void tliPrCall_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriAddress address, TriStatus encoderFailure, TriParameterList pars, TriStatus transmissionFailure)	void tliPrCall_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriAddressType address, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)
	void tliPrCall_m_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriStatus encoderFailure, TriParameterList pars, TriStatus transmissionFailure)	void tliPrCall_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)
	void tliPrCall_m_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriAddressList addresses, TriStatus encoderFailure, TriParameterList pars, TriStatus transmissionFailure)	void tliPrCall_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)

	void tliPrCall_c (String am, int ts, String src int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriComponentId to, TriStatus transmissionFailure)	void tliPrCall_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdType to, in TriStatusType transmissionFailure)
	void tliPrCall_c_BC (String am, int ts, String src int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriStatus transmissionFailure)	void tliPrCall_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriStatusType transmissionFailure)
	void tliPrCall_c_MC (String am, int ts, String src int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriComponentIdList toList, TriStatus transmissionFailure)	void tliPrCall_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)
	void tliPrGetCallDetected_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TriParameterList pars, TriAddress address)	void tliPrGetCallDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriParameterListType pars, in TriAddressType address)
	void tliPrGetCallDetected_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterList parsValue, TriComponentId from)	void tliPrGetCallDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdType from)
	void tliPrGetCallMismatch_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TciValueTemplate parsTmpl, TciValueDifferenceList diffs, TriAddress address, TciValueTemplate addressTmpl)	void tliPrGetCallMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)
	void tliPrGetCallMismatch_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TciValueTemplate parsTmpl, TciValueDifferenceList diffs, TriComponentId from, TciNonValueTemplate fromTmpl)	void tliPrGetCallMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)
	void tliPrGetCall_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TciValueTemplate parsTmpl, TriAddress address, TciValueTemplate addressTmpl)	void tliPrGetCall_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)

	void tliPrGetCall_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TciValueTemplate parsTmpl, TriComponentId from, TciNonValueTemplate fromTmpl)	void tliPrGetCall_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)
	void tliPrReply_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureIdType signature, TciParameterList parsValue, Value replValue, TriAddress address, TriStatus encoderFailure, TriParameter repl, TriStatus transmissionFailure)	void tliPrReply_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriAddressType address, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)
	void tliPrReply_m_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureIdType signature, TciParameterList parsValue, Value replValue, TriStatus encoderFailure, TriParameter repl, TriStatus transmissionFailure)	void tliPrReply_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)
	void tliPrReply_m_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureIdType signature, TciParameterList parsValue, Value replValue, TriAddressList addresses, TriStatus encoderFailure, TriParameter repl, TriStatus transmissionFailure)	void tliPrReply_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)
	void tliPrReply_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TriComponentId to, TriStatus transmissionFailure)	void tliPrReply_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriComponentIdType to, in TriStatusType transmissionFailure)
	void tliPrReply_c_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TriStatus transmissionFailure)	void tliPrReply_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriStatusType transmissionFailure)
	void tliPrReply_c_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TriComponentIdList toList, TriStatus transmissionFailure)	void tliPrReply_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)

	void tliPrGetReplyDetected_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TriParameter repl, TriAddress address)	void tliPrGetReplyDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriParameterType repl, in TriAddressType address)
	void tliPrGetReplyDetected_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, Value replValue, TriComponentId from)	void tliPrGetReplyDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value replValue, in TriComponentIdType from)
	void tliPrGetReplyMismatch_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TciValueTemplate replyTmpl, TciValueDifferenceList diffs, TriAddress address, TciValueTemplate addressTmpl)	void tliPrGetReplyMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)
	void tliPrGetReplyMismatch_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TciValueTemplate replyTmpl, TciValueDifferenceList diffs, TriComponentId from, TciNonValueTemplate fromTmpl)	void tliPrGetReplyMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)
	void tliPrGetReply_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TciValueTemplate replyTmpl, TriAddress address, TciValueTemplate addressTmpl)	void tliPrGetReply_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)
	void tliPrGetReply_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TciValueTemplate replyTmpl, TriComponentId from, TciNonValueTemplate fromTmpl)	void tliPrGetReply_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)
	void tliPrRaise_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TriAddress address, TriStatus encoderFailure, TriException exc, TriStatus transmissionFailure)	void tliPrRaise_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriAddressType address, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure)

	void tliPrRaise_m_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TriStatus encoderFailure, TriException exc, TriStatus transmissionFailure)	void tliPrRaise_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure)
	void tliPrRaise_m_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TriAddressList addresses, TriStatus encoderFailure, TriException exc, TriStatus transmissionFailure)	void tliPrRaise_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure)
	void tliPrRaise_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TriComponentId to, TriStatus transmissionFailure)	void tliPrRaise_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriComponentIdType to, in TriStatusType transmissionFailure)
	void tliPrRaise_c_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TriStatus transmissionFailure)	void tliPrRaise_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriStatusType transmissionFailure)
	void tliPrRaise_c_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TriComponentIdList toList, TriStatus transmissionFailure)	void tliPrRaise_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)
	void tliPrCatchDetected_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TriException exc, TriAddress address)	void tliPrCatchDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriExceptionType exc, in TriAddressType address)
	void tliPrCatchDetected_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, Value excValue, TriComponentId from)	void tliPrCatchDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value excValue, in TriComponentIdType from)

	void tliPrCatchMismatch_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TciValueTemplate excTmpl, TciValueDifferenceList diffs, TriAddress address, TciValueTemplate addressTmpl)	void tliPrCatchMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)
	void tliPrCatchMismatch_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TciValueTemplate excTmpl, TciValueDifferenceList diffs, TriComponentId from, TciNonValueTemplate fromTmpl)	void tliPrCatchMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)
	void tliPrCatch_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TciValueTemplate excTmpl, TriAddress address, TciValueTemplate addressTmpl)	void tliPrCatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)
	void tliPrCatch_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value excValue, TciValueTemplate excTmpl, TriComponentId from, TciNonValueTemplate fromTmpl)	void tliPrCatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)
	void tliPrCatchTimeoutDetected (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature)	void tliPrCatchTimeoutDetected(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature)
	void tliPrCatchTimeout (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue)	void tliPrCatchTimeout (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue)
	void tliCCreate (String am, int ts, String src, int line, TriComponentId c, TriComponentId comp, String name)	void tliCCreate (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TString name)
	void tliCStart (String am, int ts, String src, int line, TriComponentId c, TriComponentId comp, TciBehaviourIdType name, TciParameterListType parsValue)	void tliCStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TciBehaviourIdType name, in TciParameterListType parsValue)

	void tliCRunning (String am, int ts, String src, int line, TriComponentId c, TriComponentId comp, TBoolean status)	void tliCRunning (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status)
	void tliCStop (String am, int ts, String src, int line, TriComponentId c, TriComponentId comp)	void tliCStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp)
	void tliCDoneMismatch (String am, int ts, String src, int line, TriComponentId c, TciNonValueTemplate compTmpl)	void tliCDoneMismatch (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTmpl)
	void tliCDone (String am, int ts, String src, int line, TriComponentId c, TciNonValueTemplate compTmpl)	void tliCDone (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTmpl)
	void tliCTerminated (String am, int ts, String src, int line, TriComponentId c, VerdictValue verdict)	void tliCTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict)
	void tliPConnect (String am, int ts, String src, int line, TriComponentId c, TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2)	void tliPConnect (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)
	void tliPDisconnect (String am, int ts, String src, int line, TriComponentId c, TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2)	void tliPDisconnect (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)
	void tliPMap (String am, int ts, String src, int line, TriComponentId c, TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2)	void tliPMap (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)
	void tliPUnmap (String am, int ts, String src, int line, TriComponentId c, TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2)	void tliPUnmap (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)
	void tliPClear (String am, int ts, String src, int line, TriComponentId c, TriPortId port)	void tliPClear (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)
	void tliPStart (String am, int ts, String src, int line, TriComponentId c, TriPortId port)	void tliPStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)
	void tliPStop (String am, int ts, String src, int line, TriComponentId c, TriPortId port)	void tliPStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)

	void tliPHalt (String am, int ts, String src, int line, TriComponentId c, TriPortId port)	void tliPHalt (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)
	void tliEncode (String am, int ts, String src, int line, TriComponentId c, Value val, TriStatus encoderFailure, TriMessage msg, String codec)	void tliEncode (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value val, in TriStatusType encoderFailure, in TriMessageType msg, in TString codec)
	void tliDecode (String am, int ts, String src, int l ine, TriComponentId c, TriMessage msg, TriStatus decoderFailure, Value val, String codec)	void tliDecode (in TString am, in TInteger ts, in TString src, in TInteger l ine, in TriComponentIdType c, in TriMessageType msg, in TriStatusType decoderFailure, in Value val, in TString codec)
	void tliTTimeoutDetected (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer)	void tliTTimeoutDetected (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer)
	void tliTTimeoutMismatch (String am, int ts, String src, int line, TriComponentId c, TciNonValueTemplate timerTmpl)	void tliTTimeoutMismatch (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTmpl)
	void tliTTimeout (String am, int ts, String src, int line, TriComponentId c, TciNonValueTemplate timerTmpl)	void tliTTimeout (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTmpl)
	void tliTStart (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer, TriTimerDuration dur)	void tliTStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType dur)
	void tliTStop (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer)	void tliTStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer)
	void tliTRead (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer, TriTimerDuration elapsed)	void tliTRead (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType elapsed)
	void tliTRunning (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer, Boolean status)	void tliTRunning (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TBoolean status)
	void tliSEnter (String am, int ts, String src, int line, TriComponentId c, String name, TciParameterListType parsValue, String kind)	void tliSEnter (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in TciParameterListType parsValue, in TString kind)
	void tliSLeave (String am, int ts, String src, int line, TriComponentId c, String name, Value returnValue, String kind)	void tliSLeave (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value returnValue, in TString kind)
	void tliVar (String am, int ts, String src, int line, TriComponentId c, String name, Value varValue)	void tliVar (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value varValue)

	void tliModulePar (String am, int ts, String src, int line, TriComponentId c, String name, Value parValue)	void tliModulePar (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value parValue)
	void tliGetVerdict (String am, int ts, String src, int line, TriComponentId c, VerdictValue verdict)	void tliGetVerdict (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict)
	void tliSetVerdict (String am, int ts, String src, int line, TriComponentId c, VerdictValue verdict)	void tliSetVerdict (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict)
	void tliLog (String am, int ts, String src, int line, TriComponentId c, Value[] log)	void tliLog (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciValueList log)
	void tliAEnter (String am, int ts, String src, int line, TriComponentId c)	void tliAEnter (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)
	void tliALeave (String am, int ts, String src, int line, TriComponentId c)	void tliALeave (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)
	void tliADefaults (String am, int ts, String src, int line, TriComponentId c)	void tliADefaults (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)
	void tliAActivate (String am, int ts, String src, int line, TriComponentId c, String name, TciParameterListType pars, Value ref)	void tliAActivate (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in TciParameterListType pars, in Value ref)
	void tliADeactivate (String am, int ts, String src, int line, TriComponentId c, Value ref)	void tliADeactivate (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value ref)
	void tliANomatch (String am, int ts, String src, int line, TriComponentId c)	void tliANomatch (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)
	void tliARepeat (String am, int ts, String src, int line, TriComponentId c)	void tliARepeat (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)
	void tliAWait (String am, int ts, String src, int line, TriComponentId c)	void tliAWait (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)

5.9 معطيات

ملاحظات وتعليقات	ANSI-C representation (Type definition)	TCI IDL ADT
	QualifiedName	**TciModuleIdType**
	typedef struct TciModuleParameterType { String parName; TciValue defaultValue; } TciModuleParameterType;	TciModuleParameterType
	typedef struct TciModuleParameterListType { long int length; TciModuleParameterType *modParList; } TciModuleParameterListType;	TciModuleParameterListType
	typedef struct TciParameterType { String parName; TciParameterPassingModeType parPassMode; TciValue parValue; } TciParameterType;	TciParameterType

ملاحظات وتعليقات	ANSI-C representation (Type definition)	TCI IDL ADT
	QualifiedName	**TciModuleIdType**

	typedef enum
	{
	  TCI_IN_PAR = 0,
	  TCI_INOUT_PAR = 1,
	  TCI_OUT_PAR = 2
	} TciParameterPassingModeType;
		TciParameterPassingModeType

يفسر الطول 0 على أنه "قائمة فارغة"	typedef struct TciParameterListType
	{
	  long int length;
	  TciParameterType *parList;
	} TciParameterListType;
		TciParameterListType

يفسر الطول 0 على أنه "قائمة فارغة"	typedef struct TciParameterTypeListType
	{
	  long int length;
	  TciType *parList;
	} TciParameterTypeListType;
		TciParameterTypeListType

يفسر الطول 0 على أنه "قائمة فارغة"	typedef struct TciTestCaseIdListType
	{
	  long int length;
	  QualifiedName *idList;
	} TciTestCaseIdListType;
		TciTestCaseIdListType

	typedef enum
	{
	  TCI_ADDRESS_TYPE,
	  TCI_ANYTYPE_TYPE,
	  TCI_BITSTRING_TYPE,
	  TCI_BOOLEAN_TYPE,
	  TCI_CHAR_TYPE,
	  TCI_CHARSTRING_TYPE,
	  TCI_COMPONENT_TYPE,
	  TCI_ENUMERATED_TYPE,
	  TCI_FLOAT_TYPE,
	  TCI_HEXSTRING_TYPE,
	  TCI_INTEGER_TYPE,
	  TCI_OBJID_TYPE,
	  TCI_OCTETSTRING_TYPE,
	  TCI_RECORD_TYPE,
	  TCI_RECORD_OF_TYPE,
	  TCI_SET_TYPE,
	  TCI_SET_OF_TYPE,
	  TCI_UNION_TYPE,
	  TCI_UNIVERSAL_CHAR_TYPE,
	  TCI_UNIVERSAL_CHARSTRING_TYPE,
	  TCI_VERDICT_TYPE
	} TciTypeClassType;
		TciTypeClassType

	typedef enum
	{
	  TCI_CTRL_COMP,
	  TCI_MTC_COMP,
	  TCI_PTC_COMP,
	  TCI_SYS_COMP
	} TciTestComponentKindType;
		TciTestComponentKindType

	QualifiedName	TciBehaviourIdType

	typedef struct TciValueDifference
	{
	  TciValue val;
	  TciValueTemplate tmpl;
	  String desc;
	} TciValueDifference;
		TciValueDifference

يفسر الطول 0 على أنه "قائمة فارغة"	typedef struct TciValueDifferenceList
	{
	  long int length;
	  TciValueDifference[] diffList;
	} TciValueDifferenceList;
		TciValueDifferenceList

ملاحظات وتعليقات	ANSI-C representation	TCI concept
Verdict representation		
نظراً لأن السطح البيني TciVerdictValue معرف على أساس أعداد صحيحة، ينبغي إنشاء توافق في الآراء بشأن أي قيمة تعرف أي حكم.	`const int TCI_VERDICT_NONE = 0`	NONE
	`const int TCI_VERDICT_PASS = 1`	PASS
	`const int TCI_VERDICT_INCONC = 2`	INCONC
	`const int TCI_VERDICT_FAIL = 3`	FAIL
	`const int TCI_VERDICT_ERROR = 4`	ERROR
Objid representation		
نظراً لأن قيمة Objid تعاد "كما هي" عبر السطح البيني لقيمة Objid، ينبغي تعريف تمثيل.	`typedef struct TciObjidValue { long int length; TciObjidElem *elements; } TciObjidValue;`	Objid
	`typedef struct TciObjidElemValue { char* elem_as_ascii; long int elem_as_number; void* aux; } TciObjidElemValue;`	TciObjidElem
CharstringValue representation		
	`typedef struct TciCharStringValue { unsigned long int length; char* string; } TciCharStringValue`	TciCharString
`Universal Character[string] representation`		
	`typedef unsigned char[4] TciUCValue`	Universal Char
	`typedef struct TciUCStringValue { unsigned long int length; TciUCType *string; } TciUCStringValue;`	Universal Charstring

10 تقابل W3C XML

1.10 مقدمة

يقدم هذا القسم تقابل TCI XML [5] و[6] و[7] لسطح بيني لتسجيل TCI. ويعرف تقابل XML لسطح بيني لتسجيل كيفية تقابل تعاريف IDL الواردة في القسم 7 مع XML. وترد تعاريف تخطيط التقابل هذا في الملحق B.

2.10 منظورات

يتقابل **`tciInterface`** مع تخطيط XML مع مكان اسم `http://uri.etsi.org/ttcn-3/3.0.0/tci/TLI`.

يستخدم التخطيط هذا تخطيطات اضافية:

- http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes لتقابل أنماط بسيطة مع XML.
- http://uri.etsi.org/ttcn-3/3.0.0/tci/Types لتقابل أنماط مركبة مع XML.
- http://uri.etsi.org/ttcn-3/3.0.0/tci/Values لتقابل قيم مع XML.
- http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates لتقابل مقاسات مع XML.
- http://uri.etsi.org/ttcn-3/3.0.0/tci/Events لتقابل أحداث تسجيل مع XML.

3.10 تقابل نمط

1.3.10 تقابل أنماط بسيطة

TBoolean 1.1.3.10

يتقابل نمط **TBoolean** مع نمط أساسي xsd boolean.

TString 2.1.3.8

يتقابل نمط **TString** مع نمط أساسي xsd string.

TInteger 3.1.3.10

يتقابل نمط **TInteger** مع نمط أساسي xsd integer.

TriTimerDurationType 4.1.3.10

يتقابل نمط **TriTimerDurationType** مع نمط أساسي xsd float.

TciParameterPassingModeType 5.1.3.10

يتقابل نمط **TciParameterPassingModeType** مع نمط أساسي xsd string مع قيم عد 'out', 'in' و'inout'.

TriStatusType 6.1.3.10

يتقابل نمط **TriStatusType** مع نمط أساسي xsd string مع قيم عد 'TRI_Ok' و'TRI_Error'.

TciStatusType 7.1.3.10

يتقابل نمط **TriStatusType** مع نمط أساسي xsd string مع قيم عد 'TCI_Ok' و'TCI_Error'.

2.3.10 تقابل نمط معقد

TriPortIdType 1.2.3.10

يتقابل **TriPortIdType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriPortIdType">
    <xsd:sequence>
        <xsd:element name="comp" type="Types:TriComponentIdType" minOccurs="1"  maxOccurs="1"/>
        <xsd:element name="port" type="Types:Port"  minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- comp معرف المكون TRI؛
- port تعرف المنفذ.

نعوت:

- none.

TriComponentIdType 2.2.3.10

يتقابل **TriComponentIdType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriComponentIdType">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="null"/>
            <xsd:element name="id" type="Types:Id" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- `id` معرف المكون TRI؛
- `null` معرف `null`. ليستخدم إذا لم يوجد معرف مكون TRI.

نعوت:

- none.

3.2.3.10 TriComponentIdListType

يتقابل `TriComponentIdListType` مع النمط المعقد التالي:

```
<xsd:complexType name="TriComponentIdListType">
    <xsd:sequence>
        <xsd:element name="comp" type="Types:TriComponentIdType" minOccurs="0"
         maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- `comp` معرفات مكونات TRI في تلك القائمة.

نعوت:

- none.

4.2.3.10 Port

يتقابل **`Port`** مع النمط المعقد التالي:

```
<xsd:complexType name="Port">
    <xsd:sequence>
        <xsd:element name="id" type="Types:Id" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="index" type="xsd:int" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- `id` معرف المنفذ؛
- `port` دليل المنفذ.

نعوت:

- none.

5.2.3.10 Id

يستخدم **`Id`** كمعرف لمكونات ومنافذ ومؤقتات ويتقابل مع النمط المعقد التالي:

```
<xsd:complexType name="Id">
    <xsd:sequence>
        <xsd:element name="name" type="SimpleTypes:TString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="id" type="SimpleTypes:TInteger" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="type" type="SimpleTypes:TString" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- `name` اسم المكون أو المنفذ أو المؤقت؛
- `id` التمثيل الداخلي للمكون أو المنفذ أو المؤقت؛
- `type` نمط المكون أو المنفذ أو المؤقت.

نعوت:

- none.

6.2.3.10 TriMessageType

يتقابل **TriMessageType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriMessageType">
    <xsd:attribute name="val" type="xsd:hexBinary"/>
</xsd:complexType>
```

مكونات:

- val الرسالة المشفرة.

نعوت:

- none.

7.2.3.10 TriParameterType

يتقابل **TriParameterType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriParameterType">
    <xsd:sequence>
        <xsd:element name="val" type="Values:Value" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="SimpleTypes:TString"/>
    <xsd:attribute name="mode" type="SimpleTypes:TciParameterPassingModeType"/>
</xsd:complexType>
```

مكونات:

- val المعلمة المشفرة.

نعوت:

- name اسم المعلمة؛
- mode أسلوب الاتصال بالمعلمة.

8.2.3.10 TriParameterListType

يتقابل **TriParameterListType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriParameterListType">
    <xsd:sequence>
        <xsd:element name="par" type="Types:TriParameterType" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- par المعلمات في تلك القائمة.

نعوت:

- none.

9.2.3.10 TriAddressType

يتقابل **TriAddressType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriAddressType">
    <xsd:attribute name="val" type="SimpleTypes:TString"/>
</xsd:complexType>
```

مكونات:

- val قيمة العنوان.

نعوت:

- none.

TriAddressListType 10.2.3.10

يتقابل **TriAddressListType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriAddressListType">
    <xsd:sequence>
        <xsd:element name="addr" type="Types:TriAddressType" minOccurs="0"
         maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- `addr` العناوين في تلك القائمة.

نعوت:

- none.

TriExceptionType 10.3.2.11

يتقابل **TriExceptionType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriExceptionType">
    <xsd:attribute name="val" type="SimpleTypes:TString"/>
</xsd:complexType>
```

مكونات:

- `val` الاستثناء.

نعوت:

- none.

TriSignatureIdType 12.2.3.10

يتقابل **TriSignatureIdType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriSignatureIdType">
    <xsd:attribute name="val" type="SimpleTypes:TString"/>
</xsd:complexType>
```

مكونات:

- `val` التوقيع.

نعوت:

- none.

TriAddressType 13.2.3.10

يتقابل **TriAddressType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriAddressType">
    <xsd:attribute name="val" type="SimpleTypes:TString"/>
</xsd:complexType>
```

مكونات:

- `val` العنوان في SUT.

نعوت:

- none.

TriTimerIdType 10.3.2.14

يتقابل **TriTimerIdType** مع النمط المعقد التالي:

```
<xsd:complexType name="TriTimerIdType">
    <xsd:sequence>
        <xsd:element name="id" type="Types:Id" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- `id` تعرف المؤقت.

نعوت:

- none.

TriTimerDurationType 15.2.3.10

يتقابل **`TriTimerDurationType`** مع النمط المعقد التالي:

```
<xsd:complexType name="TriTimerDurationType">
    <xsd:attribute name="val" type="SimpleTypes:TriTimerDurationType"/>
</xsd:complexType>
```

مكونات:

- `val` مدة المؤقت.

نعوت:

- none.

QualifiedName 10.3.2.16

يستخدم **`QualifiedName`** ليؤهل بالكامل معلمات ومتغيرات وحدة ويتقابل مع النمط المعقد التالي:

```
<xsd:complexType name="QualifiedName">
    <xsd:attribute name="moduleName" type="SimpleTypes:TString" use="required"/>
    <xsd:attribute name="baseName" type="SimpleTypes:TString" use="required"/>
</xsd:complexType>
```

مكونات:

- `moduleName` اسم الوحدة لوحدة TTCN-3؛
- `baseName` اسم الشيء المؤهل بالكامل.

نعوت:

- none.

TciBehaviourIdType 17.2.3.10

يتقابل **`TciBehaviourIdType`** مع النمط المعقد التالي:

```
<xsd:complexType name="TciBehaviourIdType">
    <xsd:sequence>
        <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- `name` الاسم المؤهل للسلوك.

نعوت:

- none.

TciTestCaseIdType 18.2.3.10

يتقابل **`TciTestCaseIdType`** مع النمط المعقد التالي:

```
<xsd:complexType name="TciTestCaseIdType">
    <xsd:sequence>
        <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
```

مكونات:

- `name` الاسم المؤهل لحالة الاختبار.

نعوت:

- none.

TciParameterType 19.2.3.10

يتقابل **TciParameterType** مع النمط المعقد التالي:

```
<xsd:complexType name="TciParameterType">
    <xsd:sequence>
        <xsd:element name="val" type="Values:Value" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="SimpleTypes:TString"/>
    <xsd:attribute name="mode" type="SimpleTypes:TciParameterPassingModeType"/>
</xsd:complexType>
```

مكونات:

- val المعلمة المشفرة.

نعوت:

- name اسم المعلمة؛
- mode أسلوب اتصالات المعلمة.

TciParameterListType 20.2.3.10

يتقابل **TciParameterListType** مع النمط المعقد التالي:

```
<xsd:complexType name="TciParameterListType">
    <xsd:sequence>
        <xsd:element name="par" type="Types:TriParameterType"
         minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

تتابع مكونات:

- par المعلمات في تلك القائمة.

نعوت:

- none.

3.3.10 تقابل قيمة مجردة

1.3.3.10 قيمة

يتقابل **Value** مع النمط المعقد التالي:

```
<xsd:complexType name="Value" mixed="true">
    <xsd:choice>
        <xsd:element name="integer" type="Values:IntegerValue"/>
        <xsd:element name="float" type="Values:FloatValue"/>
        <xsd:element name="boolean" type="Values:BooleanValue"/>
        <xsd:element name="objid" type="Values:ObjidValue"/>
        <xsd:element name="verdicttype" type="Values:VerdictValue"/>
        <xsd:element name="bitstring" type="Values:BitstringValue"/>
        <xsd:element name="hexstring" type="Values:HexstringValue"/>
        <xsd:element name="octetstring" type="Values:OctetstringValue"/>
        <xsd:element name="charstring" type="Values:CharstringValue"/>
        <xsd:element name="universal_charstring" type="Values:UniversalCharstringValue"/>
        <xsd:element name="record" type="Values:RecordValue"/>
        <xsd:element name="record_of" type="Values:RecordOfValue"/>
        <xsd:element name="set" type="Values:SetValue"/>
        <xsd:element name="set_of" type="Values:SetOfValue"/>
        <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
        <xsd:element name="union" type="Values:UnionValue"/>
        <xsd:element name="anytype" type="Values:AnytypeValue"/>
        <xsd:element name="address" type="Values:AddressValue"/>
    </xsd:choice>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
<xsd:attributeGroup name="ValueAtts">
```

```
    <xsd:attribute name="name" type="SimpleTypes:TString" use="optional"/>
    <xsd:attribute name="type" type="SimpleTypes:TString" use="optional"/>
    <xsd:attribute name="module" type="SimpleTypes:TString" use="optional"/>
</xsd:attributeGroup>
```

اختيار مكونات:

- `integer` قيمة الصحيح.
- `float` قيمة مطلق.
- `boolean` قيمة بولاني.
- `objid` قيمة معرف شيء.
- `verdicttype` قيمة نمط حكم.
- `bitstring` قيمة سلسلة ثنائية.
- `hexstring` قيمة سلسلة ستة عشري.
- `octetstring` قيمة سلسلة أثمونات.
- `charstring` قيمة سلسلة سمات.
- `universal_charstring` قيمة سلسلة سمات عالمية.
- `record` قيمة سجل.
- `record_of` قيمة سجل لـــ.
- `set` قيمة مجموعة.
- `set_of` قيمة مجموعة لـــ.
- `enumerated` قيمة معدودة.
- `union` قيمة اتحاد.
- `anytype` قيمة أي نمط.
- `address` قيمة عنوان.

نعوت:

- `name` اسم القيمة، إذا عرف.
- `type` نمط القيمة، إذا عرف.
- `module` وحدة القيمة، إذا عرفت.

IntegerValue 2.3.3.10

يتقابل **`IntegerValue`** مع النمط المعقد التالي:

```
<xsd:complexType name="IntegerValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- `base` قيمة الصحيح كسلسلة.
- `extension` نفس النعوت كالتي للقيمة.

FloatValue 3.3.3.10

يتقابل **`FloatValue`** مع النمط المعقد التالي:

```
<xsd:complexType name="FloatValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- base قيمة المطلق كسلسلة.
- extension نفس النعوت كالتي للقيمة.

BooleanValue 4.3.3.10

يتقابل **BooleanValue** مع النمط المعقد التالي:

```
<xsd:complexType name="BooleanValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- base قيمة البولاني كسلسلة.
- extension نفس النعوت كالتي للقيمة.

ObjidValue 5.3.3.10

يتقابل **ObjidValue** مع النمط المعقد التالي:

```
<xsd:complexType name="ObjidValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- base قيمة معرف شيء كسلسلة.
- extension نفس النعوت كالتي للقيمة.

VerdictValue 6.3.3.10

يتقابل **VerdictValue** مع النمط المعقد التالي:

```
<xsd:complexType name="VerdictValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- base قيمة الحكم كسلسلة.
- extension نفس النعوت كالتي للقيمة.

BitstringValue 7.3.3.10

يتقابل **BitstringValue** مع النمط المعقد التالي:

```
<xsd:complexType name="BitstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
            <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- base قيمة سلسلة ثنائية كسلسلة.
- extension نفس النعوت كالتي للقيمة.

HexstringValue 8.3.3.10

يتقابل **HexstringValue** مع النمط المعقد التالي:

```
<xsd:complexType name="HexstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
            <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- base قيمة ستة عشري كسلسلة.
- extension نفس النعوت كالتي للقيمة.

OctetstringValue 9.3.3.10

يتقابل **OctetstringValue** مع النمط المعقد التالي:

```
<xsd:complexType name="OctetstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
            <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- base قيمة سلسلة أثمونات كسلسلة.
- extension نفس النعوت كالتي للقيمة.

CharstringValue 10.3.3.10

يتقابل **CharstringValue** مع النمط المعقد التالي:

```
<xsd:complexType name="CharstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
            <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- base قيمة سلسلة سمات كسلسلة.
- extension نفس النعوت كالتي للقيمة.

UniversalCharstringValue 11.3.3.10

يتقابل **UniversalCharstringValue** مع النمط المعقد التالي:

```
<xsd:complexType name="UniversalCharstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
            <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

محتوي بسيط:

- base قيمة سلسلة سمات عالمية كسلسلة.
- extension نفس النعوت كالتي للقيمة.

RecordValue 12.3.3.10

يتقابل **RecordValue** مع النمط المعقد التالي:

```
<xsd:complexType name="RecordValue">
    <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="integer" type="Values:IntegerValue"/>
            <xsd:element name="float" type="Values:FloatValue"/>
            <xsd:element name="boolean" type="Values:BooleanValue"/>
            <xsd:element name="objid" type="Values:ObjidValue"/>
            <xsd:element name="verdicttype" type="Values:VerdictValue"/>
            <xsd:element name="bitstring" type="Values:BitstringValue"/>
            <xsd:element name="hexstring" type="Values:HexstringValue"/>
            <xsd:element name="octetstring" type="Values:OctetstringValue"/>
            <xsd:element name="charstring" type="Values:CharstringValue"/>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue"/>
            <xsd:element name="record" type="Values:RecordValue"/>
            <xsd:element name="record_of" type="Values:RecordOfValue"/>
            <xsd:element name="set" type="Values:SetValue"/>
            <xsd:element name="set_of" type="Values:SetOfValue"/>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
            <xsd:element name="union" type="Values:UnionValue"/>
            <xsd:element name="anytype" type="Values:AnytypeValue"/>
            <xsd:element name="address" type="Values:AddressValue"/>
        </xsd:choice>
    </xsd:sequence>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
```

تتابع مكونات:

- `integer` قيمة صحيح.
- `float` قيمة مطلق.
- `boolean` قيمة بولاني.
- `objid` قيمة معرف شيء.
- `verdicttype` قيمة نمط حكم.
- `bitstring` قيمة سلسلة ثنائية.
- `hexstring` قيمة سلسلة ستة عشري.
- `octetstring` قيمة سلسلة أثمونات.
- `charstring` قيمة سلسلة سمات.
- `universal_charstring` قيمة سلسلة سمات عالمية.
- `record` قيمة سجل.
- `record_of` قيمة سجل لـــ.
- `set` قيمة مجموعة.
- `set_of` قيمة مجموعة لـــ.
- `enumerated` قيمة معدودة.
- `union` قيمة اتحاد.
- `anytype` قيمة أي نمط.
- `address` قيمة عنوان.

نعوت:

- نفس النعوت كالتي للقيمة.

RecordOfValue 13.3.3.10

يتقابل **RecordOfValue** مع النمط المعقد التالي:

```
<xsd:complexType name="RecordOfValue">
    <xsd:choice>
        <xsd:sequence>
            <xsd:element name="integer" type="Values:IntegerValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="float" type="Values:FloatValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="boolean" type="Values:BooleanValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="objid" type="Values:ObjidValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="bitstring" type="Values:BitstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="hexstring" type="Values:HexstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="octetstring" type="Values:OctetstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="charstring" type="Values:CharstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="record" type="Values:RecordValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="record_of" type="Values:RecordOfValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="set" type="Values:SetValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="set_of" type="Values:SetOfValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="union" type="Values:UnionValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="anytype" type="Values:AnytypeValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="address" type="Values:AddressValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:choice>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
```

اختيار تتابع مكونات:

- `integer` قيمة صحيح.
- `float` قيمة مطلق.
- `boolean` قيمة بولاني.
- `objid` قيمة معرف شيء.
- `verdicttype` قيمة نمط حكم.
- `bitstring` قيمة سلسلة ثنائية.
- `hexstring` قيمة سلسلة ستة عشري.
- `octetstring` قيمة سلسلة أثمونات.
- `charstring` قيمة سلسلة سمات.
- `universal_charstring` قيمة سلسلة سمات عالمية.
- `record` قيمة سجل.
- `record_of` قيمة سجل لـــ.
- `set` قيمة مجموعة.
- `set_of` قيمة مجموعة لـــ.
- `enumerated` قيمة معدودة.
- `union` قيمة اتحاد.
- `anytype` قيمة أي نمط.
- `address` قيمة عنوان.

نعوت:

- نفس النعوت كالتي للقيمة.

SetValue 14.3.3.10

يتقابل **`SetValue`** مع النمط المعقد التالي:

```
<xsd:complexType name="SetValue">
    <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="integer" type="Values:IntegerValue"/>
            <xsd:element name="float" type="Values:FloatValue"/>
            <xsd:element name="boolean" type="Values:BooleanValue"/>
            <xsd:element name="objid" type="Values:ObjidValue"/>
            <xsd:element name="verdicttype" type="Values:VerdictValue"/>
            <xsd:element name="bitstring" type="Values:BitstringValue"/>
            <xsd:element name="hexstring" type="Values:HexstringValue"/>
            <xsd:element name="octetstring" type="Values:OctetstringValue"/>
            <xsd:element name="charstring" type="Values:CharstringValue"/>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue"/>
            <xsd:element name="record" type="Values:RecordValue"/>
            <xsd:element name="record_of" type="Values:RecordOfValue"/>
            <xsd:element name="set" type="Values:SetValue"/>
            <xsd:element name="set_of" type="Values:SetOfValue"/>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
            <xsd:element name="union" type="Values:UnionValue"/>
            <xsd:element name="anytype" type="Values:AnytypeValue"/>
            <xsd:element name="address" type="Values:AddressValue"/>
        </xsd:choice>
    </xsd:sequence>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
```

اختيار تتابع مكونات:

- `integer` قيمة صحيح.
- `float` قيمة مطلق.
- `boolean` قيمة بولاني.

- `objid` قيمة معرف شيء.
- `verdicttype` قيمة نمط حكم.
- `bitstring` قيمة سلسلة ثنائية.
- `hexstring` قيمة سلسلة ستة عشري.
- `octetstring` قيمة سلسلة أثمونات.
- `charstring` قيمة سلسلة سمات.
- `universal_charstring` قيمة سلسلة سمات عالمية.
- `record` قيمة سجل.
- `record_of` قيمة سجل لـــ.
- `set` قيمة مجموعة.
- `set_of` قيمة مجموعة لـــ.
- `enumerated` قيمة معدودة.
- `union` قيمة اتحاد.
- `anytype` قيمة أي نمط.
- `address` قيمة عنوان.

نعوت:

- نفس النعوت كالتي للقيمة.

SetOfValue 15.3.3.10

يتقابل **`SetOfValue`** مع النمط المعقد التالي:

```
<xsd:complexType name="SetOfValue">
    <xsd:choice>
        <xsd:sequence>
            <xsd:element name="integer" type="Values:IntegerValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="float" type="Values:FloatValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="boolean" type="Values:BooleanValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="objid" type="Values:ObjidValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="bitstring" type="Values:BitstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="hexstring" type="Values:HexstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="octetstring" type="Values:OctetstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="charstring" type="Values:CharstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="record" type="Values:RecordValue" minOccurs="0"
                maxOccurs="unbounded"/>
```

```
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="record_of" type="Values:RecordOfValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="set" type="Values:SetValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="set_of" type="Values:SetOfValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="union" type="Values:UnionValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="anytype" type="Values:AnytypeValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="address" type="Values:AddressValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:choice>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
```

اختيار تتابع مكونات:

- `integer` قيمة صحيح.
- `float` قيمة مطلق.
- `boolean` قيمة بولاني.
- `objid` قيمة معرف شيء.
- `verdicttype` قيمة نمط حكم.
- `bitstring` قيمة سلسلة ثنائية.
- `hexstring` قيمة سلسلة ستة عشري.
- `octetstring` قيمة سلسلة أثمونات.
- `charstring` قيمة سلسلة سمات.
- `universal_charstring` قيمة سلسلة سمات عالمية.
- `record` قيمة سجل.
- `record_of` قيمة سجل لـــ.
- `set` قيمة مجموعة.
- `set_of` قيمة مجموعة لـــ.
- `enumerated` قيمة معدودة.
- `union` قيمة اتحاد.
- `anytype` قيمة أي نمط.
- `address` قيمة عنوان.

نعوت:

- نفس النعوت كالتي للقيمة.

EnumeratedValue 16.3.3.10

يتقابل **EnumeratedValue** مع النمط المعقد التالي:

```
<xsd:complexType name="EnumeratedValue">
    <xsd:sequence>
        <xsd:element name="element" type="SimpleTypes:TString"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
```

تتابع مكونات:

- element قيمة العد.

نعوت:

- نفس النعوت كالتي للقيمة.

UnionValue 17.3.3.10

يتقابل **UnionValue** مع النمط المعقد التالي:

```
<xsd:complexType name="UnionValue">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="integer" type="Values:IntegerValue"/>
        <xsd:element name="float" type="Values:FloatValue"/>
        <xsd:element name="boolean" type="Values:BooleanValue"/>
        <xsd:element name="objid" type="Values:ObjidValue"/>
        <xsd:element name="verdicttype" type="Values:VerdictValue"/>
        <xsd:element name="bitstring" type="Values:BitstringValue"/>
        <xsd:element name="hexstring" type="Values:HexstringValue"/>
        <xsd:element name="octetstring" type="Values:OctetstringValue"/>
        <xsd:element name="charstring" type="Values:CharstringValue"/>
        <xsd:element name="universal_charstring"
            type="Values:UniversalCharstringValue"/>
        <xsd:element name="record" type="Values:RecordValue"/>
        <xsd:element name="record_of" type="Values:RecordOfValue"/>
        <xsd:element name="set" type="Values:SetValue"/>
        <xsd:element name="set_of" type="Values:SetOfValue"/>
        <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
        <xsd:element name="union" type="Values:UnionValue"/>
        <xsd:element name="anytype" type="Values:AnytypeValue"/>
        <xsd:element name="address" type="Values:AddressValue"/>
    </xsd:choice>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
```

اختيار تتابع مكونات:

- integer قيمة صحيح.
- float قيمة مطلق.
- boolean قيمة بولاني.
- objid قيمة معرف شيء.
- verdicttype قيمة نمط حكم.
- bitstring قيمة سلسلة ثنائية.
- hexstring قيمة سلسلة ستة عشري.
- octetstring قيمة سلسلة أثمونات.
- charstring قيمة سلسلة سمات.
- universal_charstring قيمة سلسلة سمات عالمية.
- record قيمة سجل.
- record_of قيمة سجل لـــ.
- set قيمة مجموعة.
- set_of قيمة مجموعة لـــ.
- enumerated قيمة معدودة.

- `union` قيمة اتحاد.
- `anytype` قيمة أي نمط.
- `address` قيمة عنوان.

نعوت:

- نفس النعوت كالتي للقيمة.

AnytypeValue 10.3.3.18

يتقابل **`AnytypeValue`** مع النمط المعقد التالي:

```
<xsd:complexType name="AnytypeValue">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="integer" type="Values:IntegerValue"/>
        <xsd:element name="float" type="Values:FloatValue"/>
        <xsd:element name="boolean" type="Values:BooleanValue"/>
        <xsd:element name="objid" type="Values:ObjidValue"/>
        <xsd:element name="verdicttype" type="Values:VerdictValue"/>
        <xsd:element name="bitstring" type="Values:BitstringValue"/>
        <xsd:element name="hexstring" type="Values:HexstringValue"/>
        <xsd:element name="octetstring" type="Values:OctetstringValue"/>
        <xsd:element name="charstring" type="Values:OctetstringValue"/>
        <xsd:element name="universal_charstring"
            type="Values:UniversalCharstringValue"/>
        <xsd:element name="record" type="Values:RecordValue"/>
        <xsd:element name="record_of" type="Values:RecordOfValue"/>
        <xsd:element name="set" type="Values:SetValue"/>
        <xsd:element name="set_of" type="Values:SetOfValue"/>
        <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
        <xsd:element name="union" type="Values:UnionValue"/>
        <xsd:element name="address" type="Values:AddressValue"/>
    </xsd:choice>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
```

اختيار تتابع مكونات:

- `integer` قيمة صحيح.
- `float` قيمة مطلق.
- `boolean` قيمة بولاني.
- `objid` قيمة معرف شيء.
- `verdicttype` قيمة نمط حكم.
- `bitstring` قيمة سلسلة ثنائية.
- `hexstring` قيمة سلسلة ستة عشري.
- `octetstring` قيمة سلسلة أثمونات.
- `charstring` قيمة سلسلة سمات.
- `universal_charstring` قيمة سلسلة سمات عالمية.
- `record` قيمة سجل.
- `record_of` قيمة سجل لـــ.
- `set` قيمة مجموعة.
- `set_of` قيمة مجموعة لـــ.
- `enumerated` قيمة معدودة.
- `union` قيمة اتحاد.
- `address` قيمة عنوان.

نعوت:

- نفس النعوت كالتي للقيمة.

AddressValue 19.3.3.10

يتقابل **AddressValue** مع النمط المعقد التالي:

```
<xsd:complexType name="AddressValue">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="integer" type="Values:IntegerValue"/>
        <xsd:element name="float" type="Values:FloatValue"/>
        <xsd:element name="boolean" type="Values:BooleanValue"/>
        <xsd:element name="objid" type="Values:ObjidValue"/>
        <xsd:element name="verdicttype" type="Values:VerdictValue"/>
        <xsd:element name="bitstring" type="Values:BitstringValue"/>
        <xsd:element name="hexstring" type="Values:HexstringValue"/>
        <xsd:element name="octetstring" type="Values:OctetstringValue"/>
        <xsd:element name="charstring" type="Values:OctetstringValue"/>
        <xsd:element name="universal_charstring"
            type="Values:UniversalCharstringValue"/>
        <xsd:element name="record" type="Values:RecordValue"/>
        <xsd:element name="record_of" type="Values:RecordOfValue"/>
        <xsd:element name="set" type="Values:SetValue"/>
        <xsd:element name="set_of" type="Values:SetOfValue"/>
        <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
        <xsd:element name="union" type="Values:UnionValue"/>
        <xsd:element name="anytype" type="Values:AnytypeValue"/>
    </xsd:choice>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
```

اختيار تتابع مكونات:

- `integer` قيمة صحيح.
- `float` قيمة مطلق.
- `boolean` قيمة بولاني.
- `objid` قيمة معرف شيء.
- `verdicttype` قيمة نمط حكم.
- `bitstring` قيمة سلسلة ثنائية.
- `hexstring` قيمة سلسلة ستة عشري.
- `octetstring` قيمة سلسلة أثمونات.
- `charstring` قيمة سلسلة سمات.
- `universal_charstring` قيمة سلسلة سمات عالمية.
- `record` قيمة سجل.
- `record_of` قيمة سجل لـــ.
- `set` قيمة مجموعة.
- `set_of` قيمة مجموعة لـــ.
- `enumerated` قيمة معدودة.
- `union` قيمة اتحاد.
- `anytype` قيمة أي نمط.

نعوت:

- نفس النعوت كالتي للقيمة.

4.3.10 تقابل أنماط تسجيل مجرد

تعرف أنماط اضافية لتسهيل التسجيل للمواءمات بين قيم ومقاسات.

TciValueTemplate 1.4.3.10

يتقابل **TciValueTemplate** مع النمط المعقد التالي:

```
<xsd:complexType name="TciValueTemplate">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Values:Value">
        <xsd:choice>
```

```
            <xsd:element name="integer" type="Templates:IntegerTemplate"/>
            <xsd:element name="float" type="Templates:FloatTemplate"/>
            <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
            <xsd:element name="objid" type="Templates:ObjidTemplate"/>
            <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
            <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
            <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
            <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
            <xsd:element name="universal_charstring"
             type="Templates:UniversalCharstringTemplate"/>
            <xsd:element name="record" type="Templates:RecordTemplate"/>
            <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
            <xsd:element name="set" type="Templates:SetTemplate"/>
            <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
            <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
            <xsd:element name="union" type="Templates:UnionTemplate"/>
            <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
            <xsd:element name="address" type="Templates:AddressTemplate"/>
            <xsd:element name="omit" type="Templates:omit"/>
            <xsd:element name="any" type="Templates:any"/>
            <xsd:element name="anyoromit" type="Templates:anyoromit"/>
            <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

اختيار مكونات:

- `integer` مقاس صحيح.
- `float` مقاس مطلق.
- `boolean` مقاس بولاني.
- `objid` مقاس معرف شيء.
- `verdicttype` مقاس نمط حكم.
- `bitstring` مقاس سلسلة ثنائية.
- `hexstring` مقاس سلسلة ستة عشري.
- `octetstring` مقاس سلسلة أثمونات.
- `charstring` مقاس سلسلة سمات.
- `universal_charstring` مقاس سلسلة سمات عالمية.
- `record` مقاس سجل.
- `record_of` مقاس سجل لـــ.
- `set` مقاس مجموعة.
- `set_of` مقاس مجموعة لـــ.
- `enumerated` مقاس معدودة.
- `union` مقاس اتحاد.
- `anytype` مقاس أي نمط.
- `address` مقاس عنوان.
- `omit` مقاس شطب.
- `any` أي مقاس.
- `anyoromit` مقاس أي شطب.
- `templateDef` تعريف مقاس معقد.

نعوت:

- none.

TciNonValueTemplate 2.4.3.10

يتقابل **TciNonValueTemplate** مع النمط المعقد التالي:

```
<xsd:complexType name="TciNonValueTemplate">
    <xsd:sequence>
        <xsd:choice>
            <xsd:element name="any" type="Templates:any"/>
            <xsd:element name="all" type="Templates:all"/>
            <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

اختيار مكونات:

- `any` أي مقاس.
- `all` كل مقاسات.
- `templateDef` تعريف مقاس معقد.

نعوت:

- none.

TciValueList 3.4.3.10

يتقابل **TciValueList** مع النمط المعقد التالي:

```
<xsd:complexType name="TciValueList">
    <xsd:sequence>
        <xsd:element name="val" type="Values:Value" minOccurs="1"
         maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

تتابع مكونات:

- `val` القيم في قائمة القيمة.

نعوت:

- none.

TciValueDifference 4.4.3.10

يتقابل **TciValueDifference** مع النمط المعقد التالي:

```
<xsd:complexType name="TciValueDifference">
    <xsd:attribute name="desc" type="SimpleTypes:TString" use="optional"/>
    <xsd:attribute name="val" type="SimpleTypes:xpath" use="required"/>
    <xsd:attribute name="tmpl" type="SimpleTypes:xpath" use="required"/>
</xsd:complexType>
```

تتابع مكونات:

- `desc` السبب لعدم المواءمة.
- `val` مرجع لقيمة عدم المواءمة.
- `tmpl` مرجع للمقاس.

نعوت:

- none.

TciValueDifferenceList 5.4.3.10

يتقابل **TciValueDifferenceList** مع النمط المعقد التالي:

```
<xsd:complexType name="TciValueDifferenceList">
    <xsd:sequence>
        <xsd:element name="diff" type="Templates:TciValueDifference" minOccurs="1"
         maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

تتابع مكونات:

- `diff` فروق القيمة/المقاس في قائمة فرق القيمة.

نعوت:

- none.

4.10 تقابل عمليات على السطح البيني للتسجيل

يكون لكل عملية موفرة عند السطح البيني للتسجيل تعريف نمط معقد متوافق في XML. وتعاريف النمط المعقد هذه هي تمديدات لحدث.

1.4.10 حدث

يتقابل **Event** مع النمط المعقد التالي:

```
<!-- common definition for all events -->
<xsd:complexType name="Event" mixed="true">
    <xsd:sequence>
        <xsd:element name="am" type="SimpleTypes:TString"/>
    </xsd:sequence>
    <xsd:attribute name="ts" type="xsd:time" use="required"/>
    <xsd:attribute name="src" type="SimpleTypes:TString" use="optional"/>
    <xsd:attribute name="line" type="SimpleTypes:TInteger" use="optional"/>
<!-- general identifier structure for test components, ports and timer -->
    <xsd:attribute name="name" type="SimpleTypes:TString" use="required"/>
    <xsd:attribute name="id" type="SimpleTypes:TInteger" use="required"/>
    <xsd:attribute name="type" type="SimpleTypes:TString" use="required"/>
</xsd:complexType>
```

مكونات:

- `am` رسالة، تستخدم لمعلومات إضافية في السجل.

نعوت:

- `ts` الوقت الذي نتج فيه الحدث.
- `src` ملف المصدر لمواصفة الاختبار.
- `line` رقم الخط حيث يؤدي الطلب.
- `name` اسم المكون الذي انتج الحدث.
- `id` معرف المكون الذي انتج الحدث.
- `type` نمط المكون الذي انتج الحدث.

2.4.10 تقابل عمليات

يرد فيما يلي تقابل العمليات:

TCI-TL provided	

void tliTcExecute
(in TString am, in TInteger ts, in TString src,
in TInteger line, in TriComponentIdType c,
in TciTestCaseIdType tcId,
in TriParameterListType pars,
in TriTimerDurationType dur)
	`<xsd:complexType name="tliTcExecute">`
	`<xsd:complexContent mixed="true">`
	`<xsd:extension base="Events:Event">`
	`<xsd:sequence>`
	`<xsd:element name="tcId"`
	`type="Types:TciTestCaseIdType"/>`
	`<xsd:element name="pars"`
	`type="Types:TriParameterListType" minOccurs="0"/>`
	`<xsd:element name="dur"`
	`type="Types:TriTimerDurationType" minOccurs="0"/>`
	`</xsd:sequence>`
	`</xsd:extension>`
	`</xsd:complexContent>`
	`</xsd:complexType>`

TCI-TL provided	
void tliTcStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in TriTimerDurationType dur)	<xsd:complexType name="tliTcStart"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="tcId" type="Types:TciTestCaseIdType"/> <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="dur" type="Types:TriTimerDurationType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliTcStop(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliTcStop"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>
void tliTcStarted (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in TriTimerDurationType dur)	<xsd:complexType name="tliTcStarted"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="tcId" type="Types:TciTestCaseIdType"/> <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="dur" type="Types:TriTimerDurationType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliTcTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in VerdictValue outcome)	<xsd:complexType name="tliTcTerminated"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="tcId" type="Types:TciTestCaseIdType"/> <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="outcome" type="Values:VerdictValue"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCtrlStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliCtrlStart"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>
void tliCtrlStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliCtrlStop"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>
void tliCtrlTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliCtrlTerminated"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliMSend_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriAddressType address, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure)	<xsd:complexType name="tliMSend_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliMSend_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure)	<xsd:complexType name="tliMSend_m_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliMSend_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure)	<xsd:complexType name="tliMSend_m_MC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliMSend_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdType to, in TriStatusType transmissionFailure)	<xsd:complexType name="tliMSend_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliMSend_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriStatusType transmissionFailure)	<xsd:complexType name="tliMSend_c_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliMSend_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)	<xsd:complexType name="tliMSend_c_MC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliMDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriMessageType msg, in TriAddressType address)	<xsd:complexType name="tliMDetected_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Types:TriMessageType"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliMDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdType from)	<xsd:complexType name="tliMDetected_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Types:TriMessageType"/> <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliMMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)	<xsd:complexType name="tliMMismatch_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="msgTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliMMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)	<xsd:complexType name="tliMMismatch_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="msgTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/> <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/> <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliMReceive_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)	<xsd:complexType name="tliMReceive_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value" minOccurs="0"/> <xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliMReceive_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)	<xsd:complexType name="tliMReceive_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value" minOccurs="0"/> <xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/> <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/> <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrCall_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriAddressType address, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrCall_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCall_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrCall_m_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCall_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrCall_m_MC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrCall_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdType to, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrCall_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType" minOccurs="0"/> <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCall_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrCall_c_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCall_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrCall_c_MC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType" minOccurs="0"/> <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrGetCallDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriParameterListType pars, in TriAddressType address)	<xsd:complexType name="tliPrGetcallDetected_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="pars" type="Types:TriParameterListType"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrGetCallDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdType from)	<xsd:complexType name="tliPrGetcallDetected_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="pars" type="Types:TciParameterListType"/> <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrGetCallMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)	<xsd:complexType name="tliPrGetcallMismatch_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="pars" type="Types:TriParameterListType"/> <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrGetCallMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)	<xsd:complexType name="tliPrGetcallMismatch_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="pars" type="Types:TciParameterListType"/> <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrGetCall_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)	<xsd:complexType name="tliPrGetcall_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="from" type="Types:TriAddressType" minOccurs="0"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="pars" type="Types:TriParameterListType"/> <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrGetCall_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)	<xsd:complexType name="tliPrGetcall_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="from" type="Types:TriAddressType" minOccurs="0"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="pars" type="Types:TciParameterListType"/> <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrReply_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriAddressType address, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrReply_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrReply_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrReply_m_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrReply_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrReply_m_MC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrReply_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriComponentIdType to, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrReply_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrReply_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrReply_c_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrReply_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrReply_c_MC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrGetReplyDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriParameterType repl, in TriAddressType address)	<xsd:complexType name="tliPrGetReplyDetected_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="address" type="Types:TriAddressType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrGetReplyDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value replValue, in TriComponentIdType from)	<xsd:complexType name="tliPrGetReplyDetected_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="from" type="Types:TriComponentIdType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrGetReplyMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)	<xsd:complexType name="tliPrGetReplyMismatch_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="replTmpl" type="Values:Value"/> <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/> <xsd:element name="address" type="Types:TriAddressType"/> <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrGetReplyMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)	<xsd:complexType name="tliPrGetReplyMismatch_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="replTmpl" type="Values:Value"/> <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/> <xsd:element name="from" type="Types:TriComponentIdType"/> <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrGetReply_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)	<xsd:complexType name="tliPrGetReply_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="replTmpl" type="Values:Value"/> <xsd:element name="address" type="Types:TriAddressType"/> <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrGetReply_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)	<xsd:complexType name="tliPrGetReply_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="replTmpl" type="Values:Value"/> <xsd:element name="from" type="Types:TriComponentIdType"/> <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrRaise_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriAddressType address, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrRaise_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> <xsd:element name="excValue" type="Values:Value"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrRaise_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrRaise_m_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> <xsd:element name="excValue" type="Values:Value"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrRaise_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrRaise_m_MC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> <xsd:element name="excValue" type="Values:Value"/> <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrRaise_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriComponentIdType to, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrRaise_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="excValue" type="Values:Value"/> <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrRaise_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrRaise_c_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="excValue" type="Values:Value"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrRaise_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)	<xsd:complexType name="tliPrRaise_c_MC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="excValue" type="Values:Value"/> <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCatchDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriExceptionType exc, in TriAddressType address)	<xsd:complexType name="tliPrCatchDetected_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="exc" type="Types:TriExceptionType"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrCatchDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value excValue, in TriComponentIdType from)	<xsd:complexType name="tliPrCatchDetected_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="exc" type="Types:TriExceptionType"/> <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCatchMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)	<xsd:complexType name="tliPrCatchMismatch_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="exc" type="Values:Value"/> <xsd:element name="excTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/> <xsd:element name="address" type="Types:TriAddressType"/> <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCatchMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)	<xsd:complexType name="tliPrCatchMismatch_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="exc" type="Values:Value"/> <xsd:element name="excTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="address" type="Types:TriAddressType"/> <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)	<xsd:complexType name="tliPrCatch_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="exception" type="Values:Value"/> <xsd:element name="exceptionTmpl" type="Templates:TciValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPrCatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)	<xsd:complexType name="tliPrCatch_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="exception" type="Values:Value"/> <xsd:element name="exceptionTmpl" type="Templates:TciValueTemplate"/> <xsd:element name="from" type="Types:TriComponentIdType"/> <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCatchTimeoutDetected(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature)	<xsd:complexType name="tliPrCatchTimeoutDetected"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPrCatchTimeout (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue)	<xsd:complexType name="tliPrCatchTimeout"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCCreate (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TString name)	<xsd:complexType name="tliCCreate"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="name" type="SimpleTypes:TString"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TciBehaviourIdType name, in TciParameterListType parsValue)	<xsd:complexType name="tliCStart"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="name" type="Types:TciBehaviourIdType"/> <xsd:element name="pars" type="Types:TciParameterListType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliCRunning (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status)	<xsd:complexType name="tliCRunning"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="status" type="SimpleTypes:TBoolean"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCAlive, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status)	<xsd:complexType name="tliCAlive"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="status" type="SimpleTypes:TBoolean"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp)	<xsd:complexType name="tliCStop"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCKill (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp)	<xsd:complexType name="tliCKill"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCDoneMismatch (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTmpl)	<xsd:complexType name="tliCDoneMismatch"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/> </xsd:sequence> <xsd:attribute name="done" type="SimpleTypes:TBoolean"/> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCDone (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTmpl)	<xsd:complexType name="tliCDone"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/> </xsd:sequence> <xsd:attribute name="done" type="SimpleTypes:TBoolean"/> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliCKilledMismatch (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTmpl)	<xsd:complexType name="tliCKilledMismatch"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/> </xsd:sequence> <xsd:attribute name="done" type="SimpleTypes:TBoolean"/> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCKill (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTmpl)	<xsd:complexType name="tliCKill"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/> </xsd:sequence> <xsd:attribute name="done" type="SimpleTypes:TBoolean"/> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliCTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict)	<xsd:complexType name="tliCTerminated"> without verdict) --> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="verdict" type="Values:VerdictValue" maxOccurs="1"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliPConnect (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)	<xsd:complexType name="tliPConnect"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType>
void tliPDisconnect (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)	<xsd:complexType name="tliPDisconnect"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType>
void tliPMap (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)	<xsd:complexType name="tliPMap"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType>
void tliPUnmap (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2)	<xsd:complexType name="tliPUnmap"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliPClear (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)	<xsd:complexType name="tliPClear"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType>
void tliPStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)	<xsd:complexType name="tliPStart"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType>
void tliPStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)	<xsd:complexType name="tliPStop"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType>
void tliPHalt (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port)	<xsd:complexType name="tliPHalt"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType>
void tliEncode (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value val, in TriStatusType encoderFailure, in TriMessageType msg, in TString codec)	<xsd:complexType name="tliEncode"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="val" type="Values:Value"/> <xsd:choice> <xsd:element name="msg" type="Types:TriMessageType"/> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> </xsd:choice> </xsd:sequence> <xsd:attribute name="codec" type="SimpleTypes:TString" use="optional"/> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliDecode (in TString am, in TInteger ts, in TString src, in TInteger l ine, in TriComponentIdType c, in TriMessageType msg, in TriStatusType decoderFailure, in Value val, in TString codec)	<xsd:complexType name="tliDecode" mixed="true"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:choice> <xsd:element name="val" type="Values:Value"/> <xsd:element name="decoder-failure" type="SimpleTypes:TciStatusType"/> </xsd:choice> <xsd:element name="msg" type="Types:TriMessageType"/> </xsd:sequence> <xsd:attribute name="codec" type="SimpleTypes:TString" use="optional"/> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliTTimeoutDetected (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer)	<xsd:complexType name="tliTTimeoutDetected"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1" minOccurs="1"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliTTimeoutMismatch (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTmpl)	<xsd:complexType name="tliTTimeoutMismatch"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1" minOccurs="1"/> <xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" maxOccurs="1" minOccurs="1"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliTTimeout (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTmpl)	<xsd:complexType name="tliTTimeout"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1" minOccurs="1"/> <xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" maxOccurs="1" minOccurs="1"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliTStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType dur)	<xsd:complexType name="tliTStart"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType"/> <xsd:element name="dur" type="Types:TriTimerDurationType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliTStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer)	<xsd:complexType name="tliTStop"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliTRead (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType elapsed)	<xsd:complexType name="tliTRead"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType"/> <xsd:element name="elapsed" type="Types:TriTimerDurationType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliTRunning (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TBoolean status)	<xsd:complexType name="tliTRunning"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType"/> </xsd:sequence> <xsd:attribute name="status" type="SimpleTypes:TBoolean"/> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliSEnter (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in TciParameterListType parsValue, in TString kind)	<xsd:complexType name="tliSEnter"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/> <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="kind" type="SimpleTypes:TString"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliSLeave (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value returnValue, in TString kind)	<xsd:complexType name="tliSLeave"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/> <xsd:element name="return" type="Values:Value" minOccurs="0"/> <xsd:element name="kind" type="SimpleTypes:TString"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliVar (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value varValue)	<xsd:complexType name="tliVar"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/> <xsd:element name="val" type="Values:Value" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliModulePar (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value parValue)	<xsd:complexType name="tliModulePar"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/> <xsd:element name="val" type="Values:Value" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliGetVerdict (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict)	<xsd:complexType name="tliGetVerdict"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="verdict" type="Values:VerdictValue"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliSetVerdict (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict)	<xsd:complexType name="tliSetVerdict"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="verdict" type="Values:VerdictValue"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

TCI-TL provided	
void tliLog (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciValueList log)	<xsd:complexType name="tliLog"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="log" type=" Values:Value"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliAEnter (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliAEnter"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>
void tliALeave (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliALeave"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>
void tliADefaults (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliADefaults"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>
void tliAActivate (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in TciParameterListType pars, in Value ref)	<xsd:complexType name="tliAActivate"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/> <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="ref" type="Values:Value"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliADeactivate (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value ref)	<xsd:complexType name="tliADeactivate"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="ref" type="Values:Value"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>
void tliANomatch (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliANomatch"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>
void tliARepeat (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliARepeat"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>
void tliAWait (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c)	<xsd:complexType name="tliAWait"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType>

11 سيناريوهات الاستخدام

يحتوي هذا القسم على سيناريوهات الاستخدام التي ينبغي أن تساعد المستعملين لـــ TCI وبائعي الأدوات الموفرين لعلم دلالات فهم TCI للعمليات المعرفة في هذه التوصية.

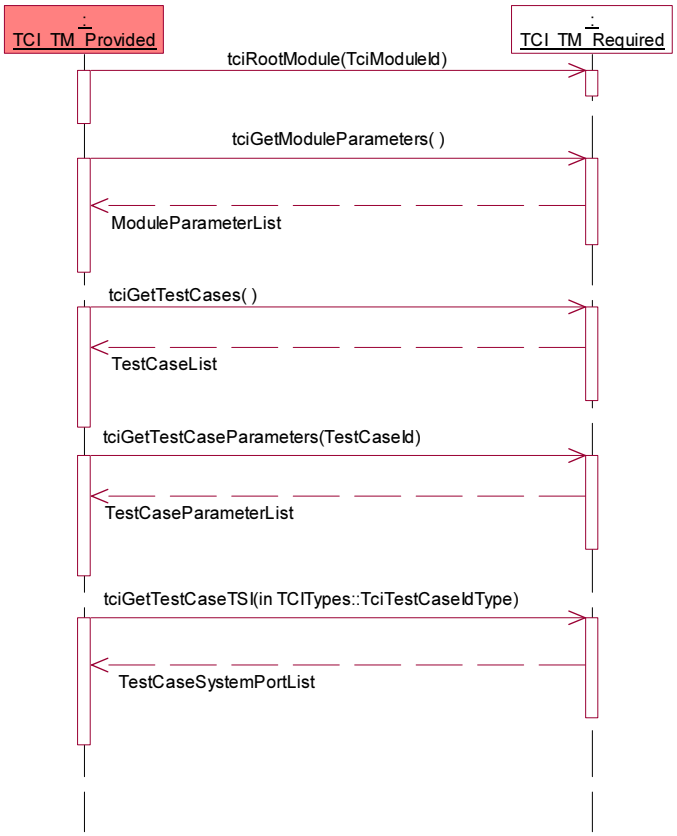
وتعرف السيناريوهات حسب الرسومات التخطيطية لتتابع UML. ويبين الرسم التخطيطي لتتابع التفاعلات بين كيانات TCI. ويتم شرح السيناريوهات وكلما كان منطبقاً تدعم تقطيع TTCN-3 المتوافق مع السيناريو.

1.11 التدميث وجمع المعلومات والتسجيل

1.1.11 سيناريو الاستخدام: التدميث

يبين السيناريو في الشكل 9 مرحلة تدميث نظام اختبار عندما تختار وحدة TTCN-3 للتنفيذ. وفي البداية، تضبط وحدة الجذر مع `tciRootModule`. ويمكن الحصول على معلمات الوحدة لوحدة الجذر مع `tciGetModuleParameters`. ويمكن أن تستخدم معلومات معلمة وحدة لتطلب من مستعمل نظام اختبار قيم ملموسة لكل معلمة وحدة. ويمكن استرداد قائمة اختبارات مجردة متاحة في وحدة الجذر مع `tciGetTestCases`. ويمكن تنفيذ الاختبارات المجردة هذه مباشرة من إدارة الاختبار. ويمكن الحصول على معلماتها ونظم اختبارها مع `tciGetTestCaseParameters` و`tciGetTestCaseTSI` على التوالي.

1.1.1.11 رسم تخطيطي تتابعي



الشكل Z.145/9 – سيناريو الاستخدام – التدميث

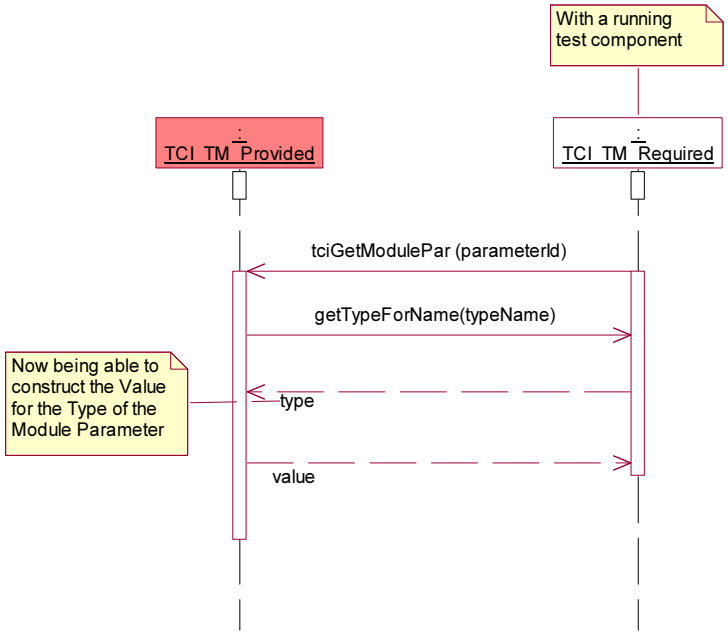
2.1.1.11 تقطيع TTCN-3

إن التدميث هو خارج مدى TTCN-3.

2.1.11 سيناريو الاستخدام: طلب معلمات وحدة

يبين السيناريو في الشكل 10 كيف يطلب مكون اختبار القيمة الفعلية لمعلمة وحدة مطلوبة لتنفيذ سلوك اختبارها. أولاً، يطلب نمط معلمة وحدة، ثم يمكن لـــ TM أن يضع القيمة ويعطيها لـــ TE.

1.2.1.11 رسم تخطيطي تتابعي



الشكل Z.145/10 – سيناريو الاستخدام – طلب معلمات وحدة

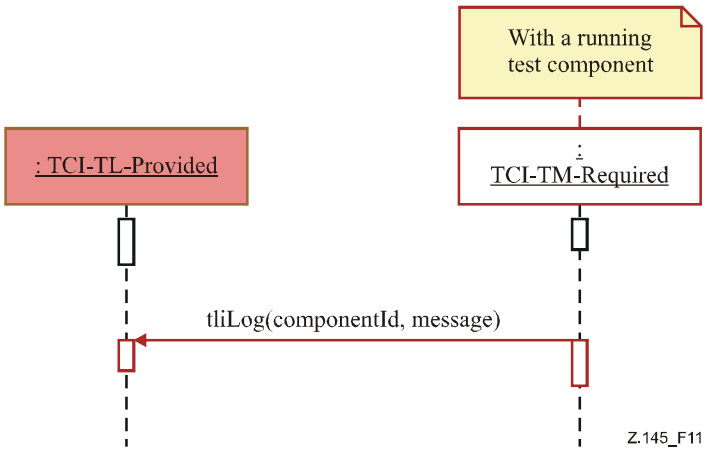
2.2.1.11 تقطيع TTCN-3

```
module AModule {
    …
    modulepar {
        integer AModulePar
    }
    …
    function AFunction (…) … {
        integer x;
        …
        x:= 2+AModulePar; // an expression with a module parameter
        …
    }
    …
}
```

3.1.11 سيناريو الاستخدام: التسجيل

يبين السيناريو في الشكل 10 تسجيل معلومات خلال تنفيذ سلوك اختبار بواسطة مكون اختبار. وتنشر الرسالة التي تسجل على تسجيل الاختبار.

1.3.1.11 رسم تخطيطي تتابعي



الشكل Z.145/11 – سيناريو الاستخدام – التسجيل

2.3.1.11 تقطيع TTCN-3

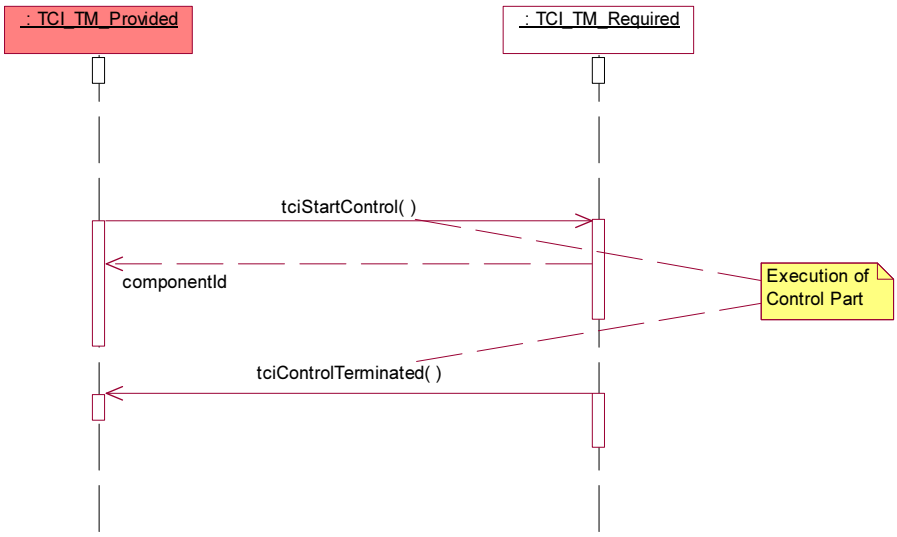
```
module AModule {
    …
    function AFunction (…) … {
        …
        log('AMessage');
        …
    }
    …
}
```

2.11 تنفيذ اختبارات مجردة وتحكم

1.2.11 سيناريو الاستخدام: تنفيذ تحكم

يبين السيناريو في الشكل 12 تتابع عمليات لتنفيذ جزء التحكم لوحدة TTCN-3. وتختار الوحدة المحتوية على جزء الحكم أولاً، ثم يبدأ التحكم، ثم ينفذ حتى ينهي TE التنفيذ.

1.1.2.11 رسم تخطيطي تتابعي



الشكل Z.145/12 – سيناريو الاستخدام – تنفيذ تحكم

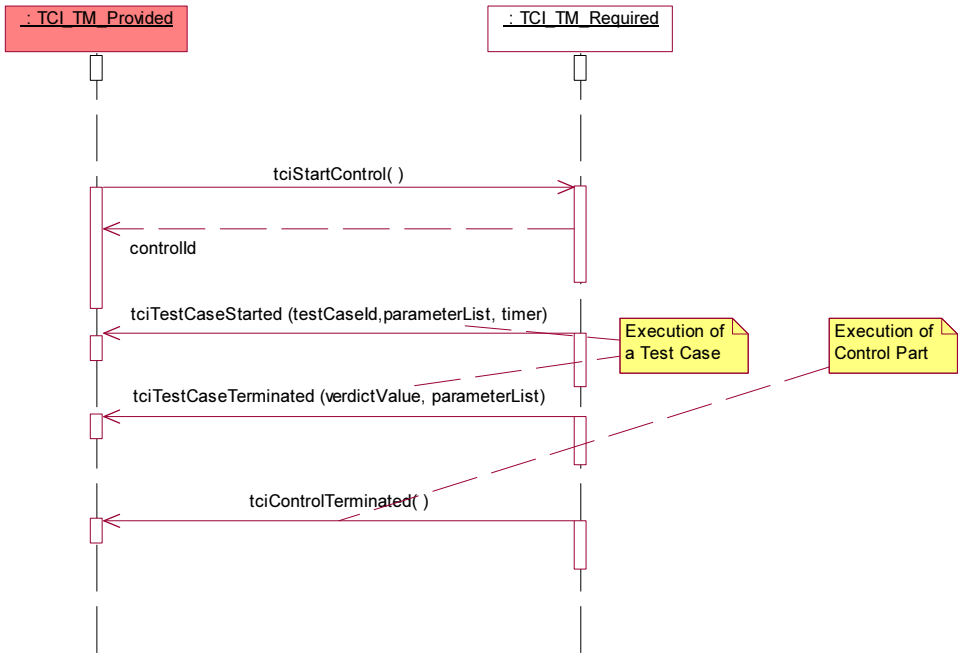
2.1.2.11 تقطيع TTCN-3

```
module AModule {
    …
    control {
        …
    }
    …
}
```

2.2.11 سيناريو الاستخدام: تنفيذ اختبار مجرد في تحكم

يبين السيناريو في الشكل 13 كيف ينفذ اختبار مجرد في جزء التحكم.

1.2.2.11 رسم تخطيطي تتابعي



الشكل 13/Z.145 – سيناريو الاستخدام – تنفيذ اختبار مجرد في تحكم

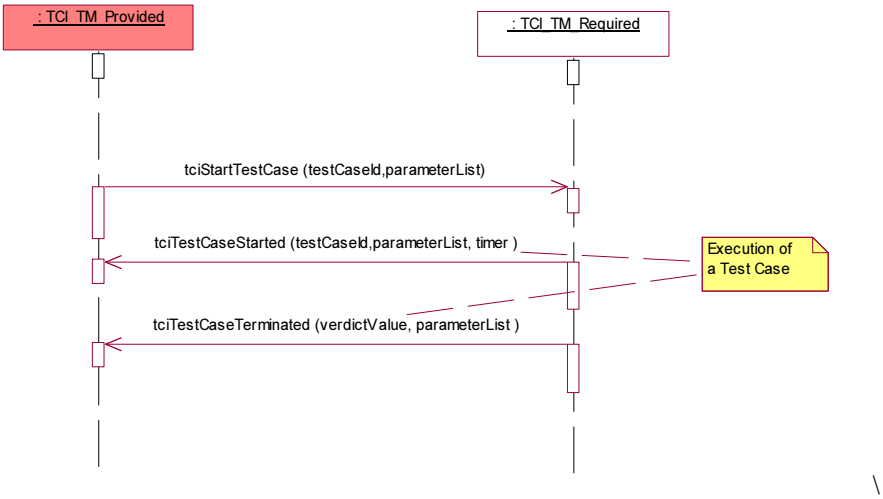
2.2.2.11 تقطيع TTCN-3

```
module AModule {
    …
    testcase ATestCase(…)… {
        … //the test case behaviour
    }
    …
    control {
        …
        execute(ATestCase(…));
        …
    }
    …
}
```

3.2.11 سيناريو الاستخدام – تنفيذ مباشر لاختبار مجرد

يبين السيناريو في الشكل 14 كيف يمكن تنفيذ اختبار مجرد مباشرة من إدارة الاختبار خارج جزء التحكم. وبعد اختيار وحدة TTCN-3 المحتوية على الاختبار المجرد الذي ينفذ، يطلب بدء الاختبار المجرد. وعندما يكمل الاختبار المجرد تنفيذه، يخطر TE إدارة الاختبار بإنهاء الاختبار المجرد.

1.3.2.11 رسم تخطيطي تتابعي



الشكل Z.145/14 – سيناريو الاستخدام – تنفيذ مباشر لاختبار مجرد

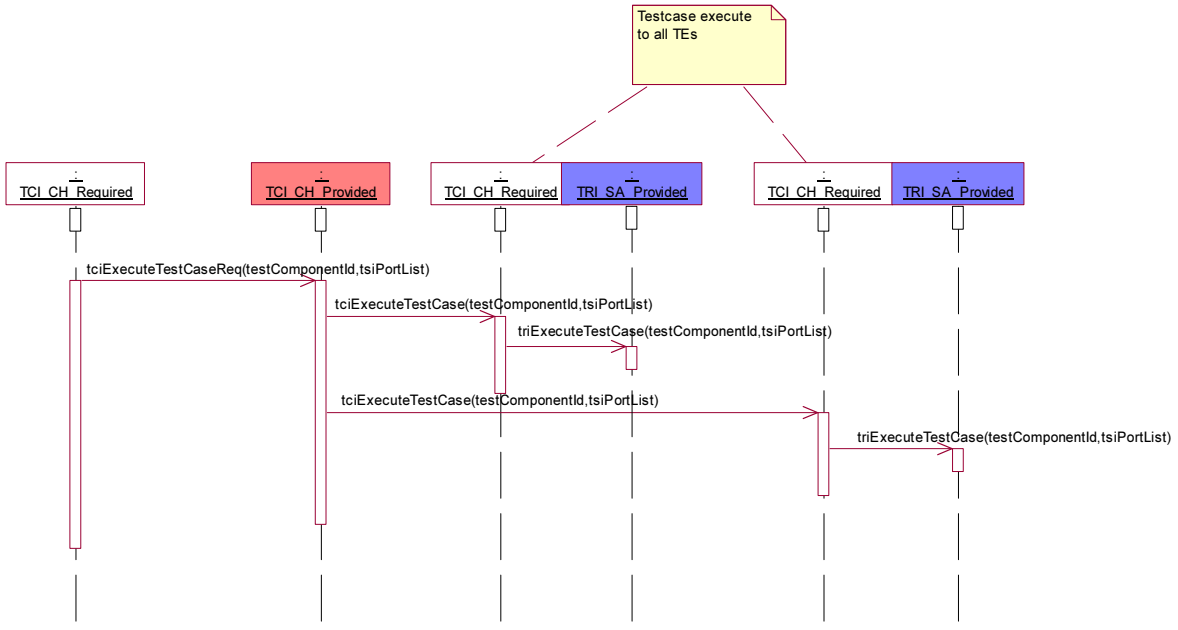
2.3.2.11 تقطيع TTCN-3

إن التنفيذ المباشر لاختبار مجرد هو خارج مدى TTCN-3.

4.2.11 سيناريو الاستخدام: تنفيذ اختبار مجرد بواسطة TRI

يبين السيناريو في الشكل 15 كيف يجرى اخطار TRI عن تنفيذ اختبار مجرد يمكنه إنشاء وتدميث منافذ نظام عند الحاجة. ويتعين إصدار طلب تنفيذ اختبار مجرد قبل أن يبدأ سلوك الاختبار على MTC للاختبار المجرد الحالي.

1.4.2.11 رسم تخطيطي تتابعي



الشكل Z.145/15 – سيناريو الاستخدام – تنفيذ اختبار مجرد بواسطة TRI

2.4.2.11 رسم تخطيطي تتابعي

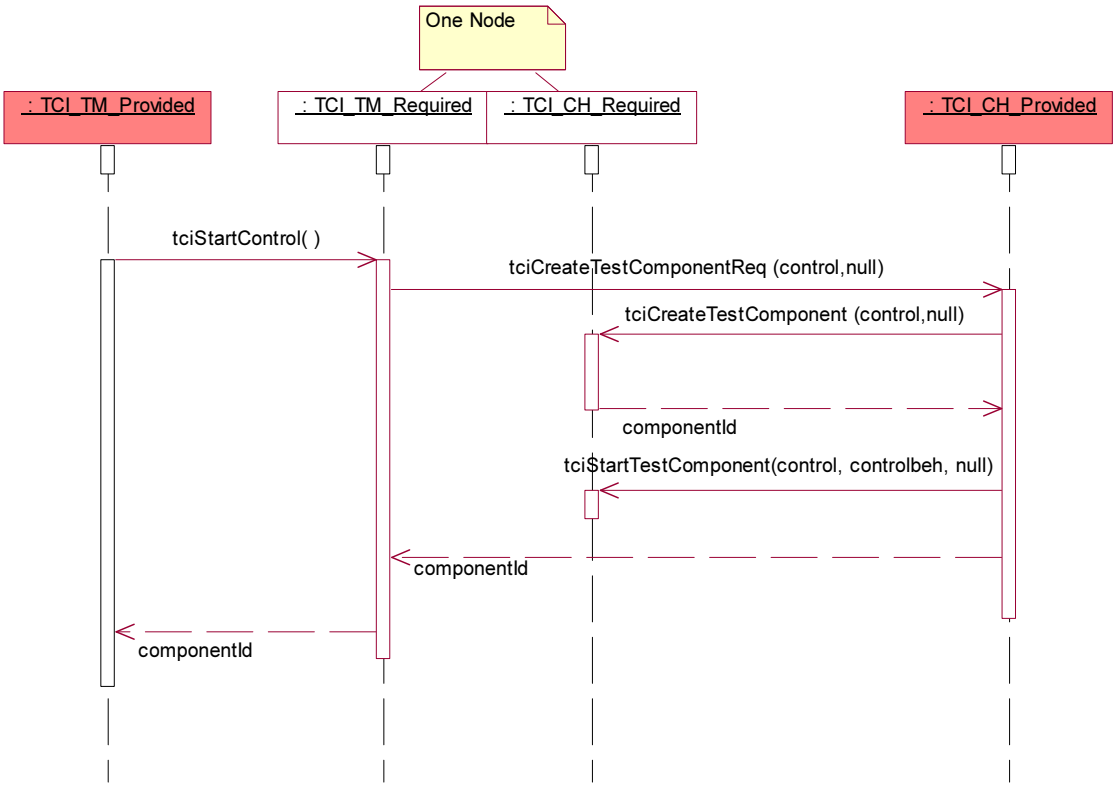
```
module AModule {
    …
    testcase ATestCase(…)… {
        … //the test case behaviour
    }
    …
    control {
        …
        execute(ATestCase(…));
        …
    }
    …
}
```

3.11 مناولة مكون

1.3.11 سيناريو الاستخدام: إنشاء مكون تحكم محلي

يبين السيناريو في الشكل 16 إنشاء مكون التحكم على نفس العقدة حيث يوجد السطح البيني للمستعمل مع إدارة الاختبار TCI-TM. ويجرى إنشاء مكون التحكم عندما ينفذ جزء التحكم لوحدة TTCN-3. وعندما تصدر إدارة الاختبار TCI-TM بدء جزء التحكم، يرسل طلب مكون إنشاء الاختبار إلى TCI-CH، التي تنشره على TE حيث ينبغي إنشاء مكون التحكم. وفي هذه الحالة، يكون TE على نفس العقدة. ويعاد معرف مكون التحكم ويعطي إلى TCI-TM. ثم يستخدم المعرف لبدء سلوك جزء التحكم على مكون التحكم.

1.1.3.11 رسم تخطيطي تتابعي



الشكل Z.145/16 – سيناريو الاستخدام – إنشاء مكون تحكم محلي

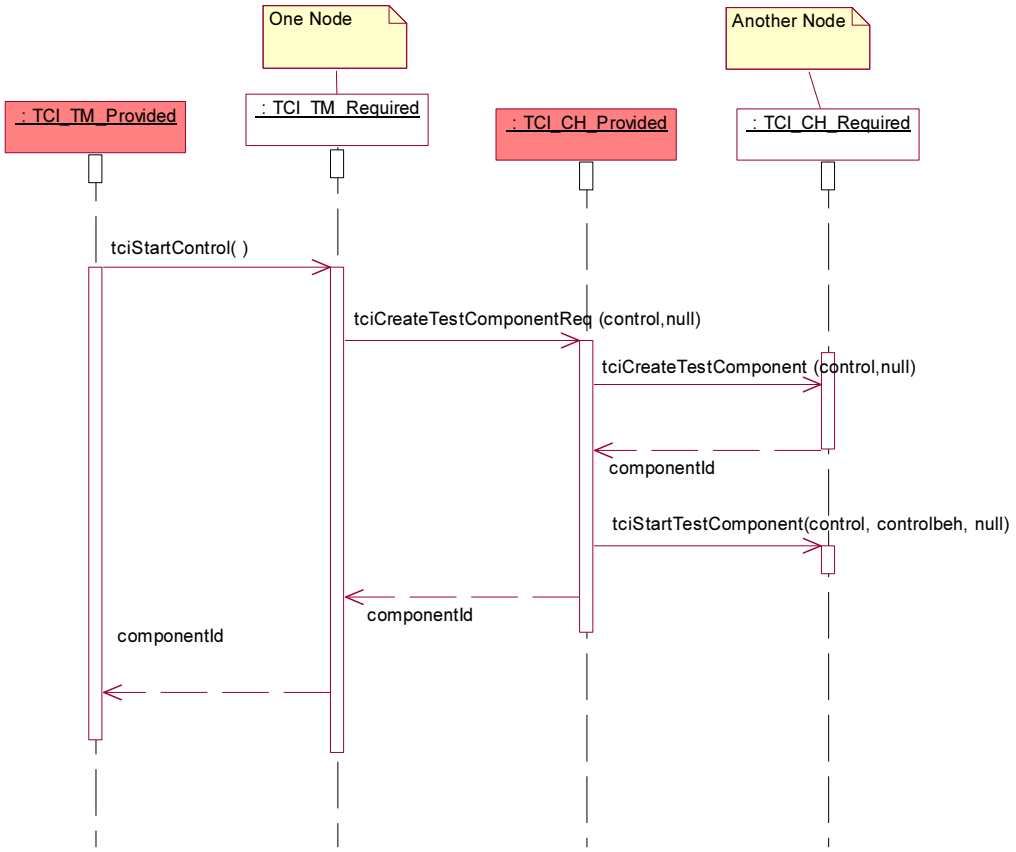
2.1.3.11 رسم تخطيطي تتابعي

```
module AModule {
    …
    control {
        …
    }
    …
}
```

2.3.11 سيناريو الاستخدام: إنشاء مكون تحكم عن ُبعد

يبين السيناريو في الشكل 17 إنشاء مكون التحكم على عقدة أخرى حيث يوجد السطح البيني للمستعمل مع إدارة الاختبار TCI-TM. ويجرى إنشاء مكون التحكم عندما ينفذ جزء التحكم لوحدة TTCN-3. وعندما تصدر إدارة الاختبار TCI-TM بدء جزء التحكم، يرسل طلب مكون إنشاء الاختبار إلى TCI-CH، التي تنشره على TE حيث ينبغي إنشاء مكون التحكم. وفي هذه الحالة، يكون TE على عقدة بعيدة أخرى. ويعاد معرف مكون التحكم ويعطي إلى TCI-TM. ثم يستخدم المعرف لبدء سلوك جزء التحكم على مكون التحكم.

1.2.3.11 رسم تخطيطي تتابعي



الشكل Z.145/17 – سيناريو الاستخدام – إنشاء مكون تحكم عن ٍبعد

2.2.3.11 تقطيع TTCN-3

```
module AModule {
    …
    control {
        …
    }
    …
}
```

3.3.11 سيناريو الاستخدام: إنشاء MTC محلي

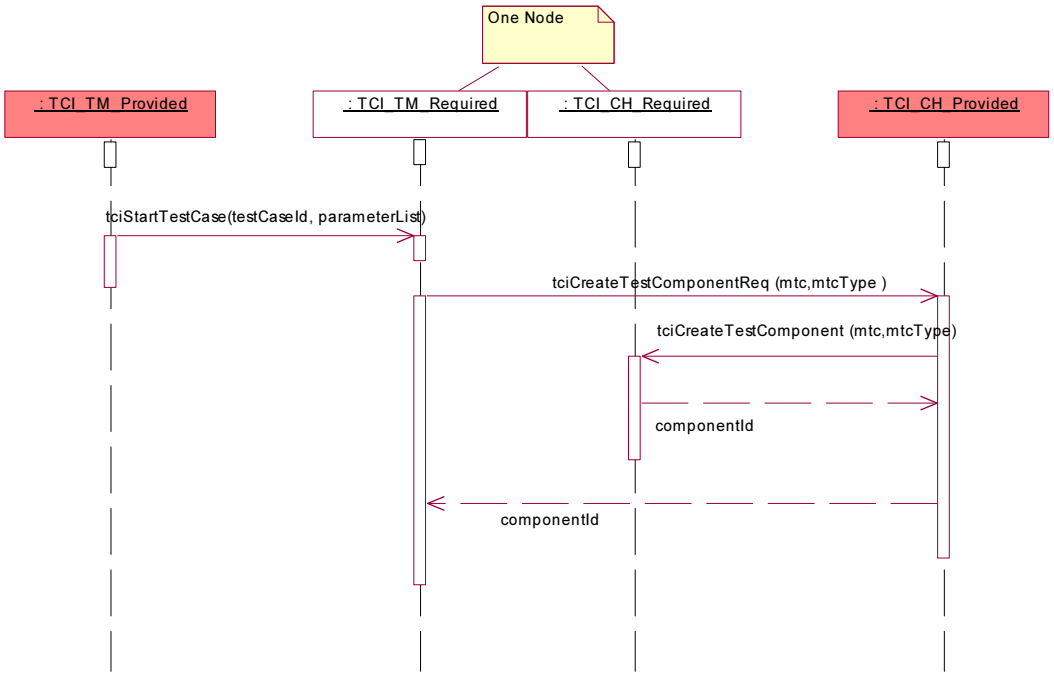
يبين السيناريو في الشكل 18 الإنشاء المحلي لمكون الاختبار الرئيسي. ويعني محلي اختبارين:

(1) على نفس العقدة حيث يوجد السطح البيني لإدارة الاختبار TCI-TM (عندما يبدأ اختبار مجرد مباشرة)؛ أو

(2) على نفس العقدة حيث يوجد مكون التحكم (عندما ينفذ اختبار مجرد من جزء التحكم).

يجرى إنشاء مكون اختبار رئيسي عندما ينفذ اختبار مجرد: يرسل طلب مكون إنشاء الاختبار إلى TCI-CH، التي تنشره على TE حيث ينبغي إنشاء مكون الاختبار الرئيسي. وفي هذه الحالة، يكون TE على نفس العقدة. ويعاد معرف مكون الاختبار الرئيسي ويعطي إلى TCI-TM. ثم يستخدم المعرف لبدء سلوك اختبار مجرد على مكون الاختبار الرئيسي (ولا يبين هذا هنا، ولكن يتم مناولته بنفس الطريقة كما في السيناريوهات الواردة في 5.3.11 و6.3.11).

1.3.3.11 رسم تخطيطي تتابعي



الشكل Z.145/18 – سيناريو الاستخدام – إنشاء MTC محلي

2.3.3.11 تقطيع TTCN-3

```
module AModule {
    …
    testcase ATestCase (…)runs on MTCType… {
        … //the test case behaviour
    }
    …
}
```

4.3.11 سيناريو الاستخدام – إنشاء MTC عن بُعد

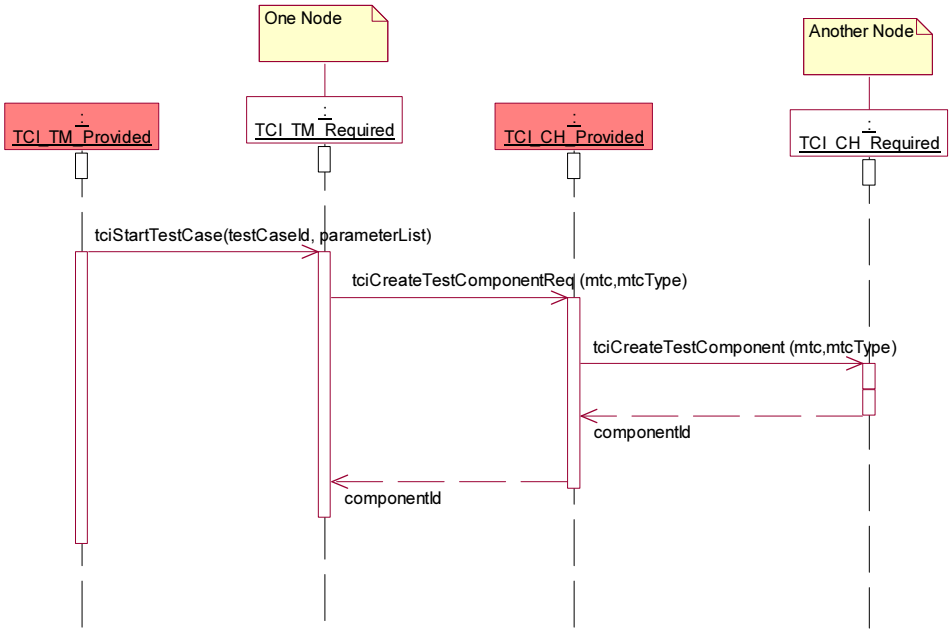
يبين السيناريو في الشكل 19 إنشاء عن بعد لمكون الاختبار الرئيسي. ويعني عن بعد اختبارين:

(1) على عقدة أخرى حيث يوجد السطح البيني لمستعمل مع إدارة الاختبار TCI-TM (عندما يبدأ اختبار مجرد مباشرة)؛ أو

(2) على عقدة أخرى حيث يوجد مكون التحكم (عندما ينفذ اختبار مجرد من جزء التحكم).

ويجرى إنشاء مكون اختبار رئيسي عندما ينفذ اختبار مجرد: يرسل طلب مكون إنشاء الاختبار إلى TCI-CH، التي تنشره على TE حيث ينبغي إنشاء مكون الاختبار الرئيسي. وفي هذه الحالة، يكون TE على عقدة أخرى. ويعاد معرف مكون الاختبار الرئيسي ويعطي إلى TCI-TM. ثم يستخدم المعرف لبدء سلوك اختبار مجرد على مكون الاختبار الرئيسي (ولا يبين هذا هنا، ولكن يتم مناولته بنفس الطريقة كما في السيناريوهات الواردة في 5.3.11 و6.3.11).

1.3.3.11 رسم تخطيطي تتابعي



الشكل Z.145/19 – سيناريو الاستخدام – إنشاء MTC بعيد

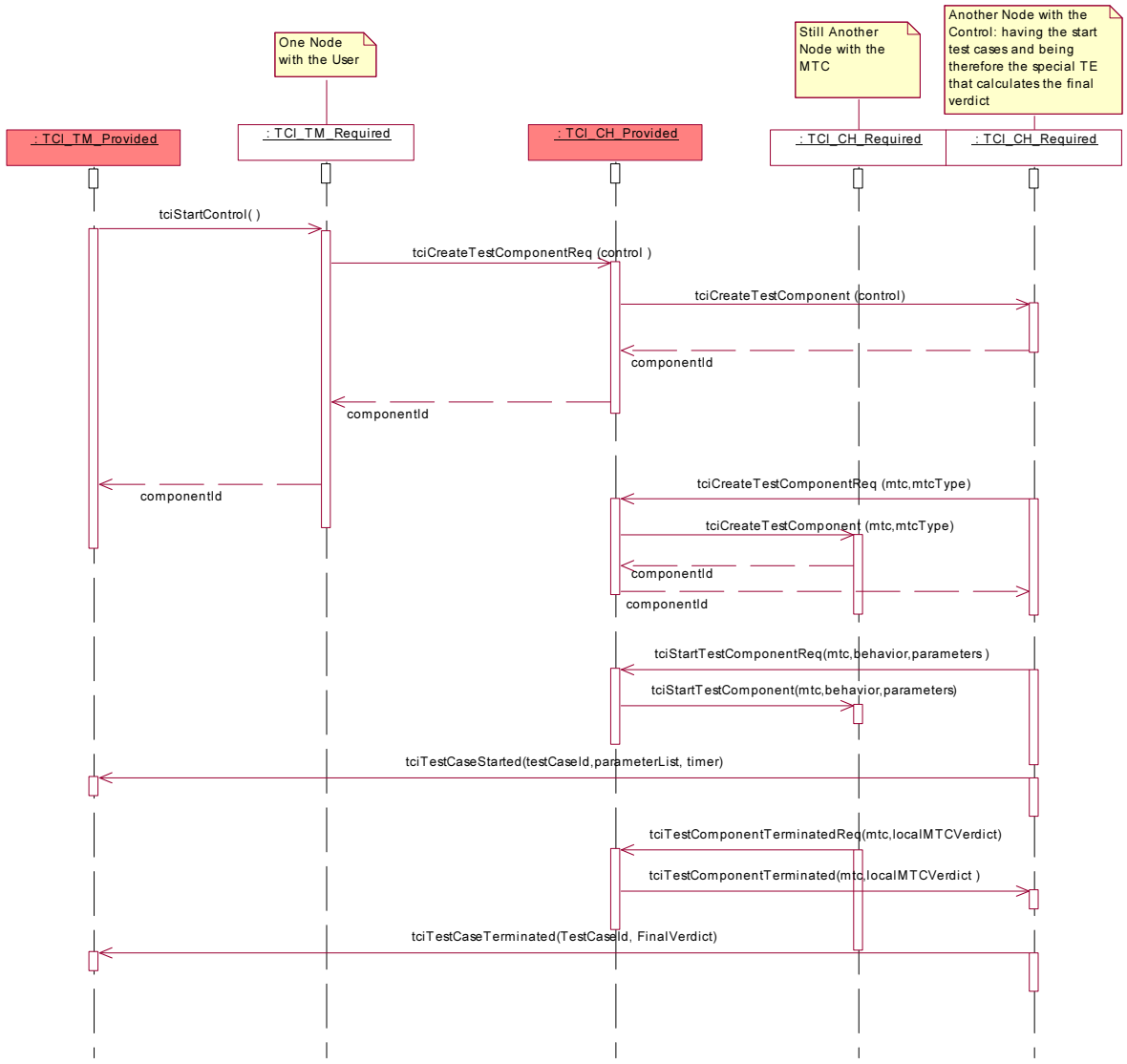
2.4.3.11 تقطيع TTCN-3

```
module AModule {
    …
    testcase ATestCase(…)runs on MTCType … {
        … //the test case behaviour
    }
    …
}
```

5.3.11 سيناريو الاستخدام: مناولة مكون لتنفيذ اختبار مجرد في تحكم

يبين السيناريو في الشكل 20 مناولة مكونات تنفيذ اختبار مجرد في جزء التحكم. وعندما يبدأ جزء التحكم، ينشأ مكون تحكم ويعاد معرف مكونه إلى إدارة الاختبار. ولكل اختبار مجرد ينفذ في جزء التحكم، ينشأ مكون اختبار رئيسي ويعاد معرف المكون إلى مكون التحكم. وبعد ذلك، يبدأ سلوك الاختبار المجرد على مكون الاختبار الرئيسي ويتم إبلاغ إدارة الاختبار عن بدء الاختبار المجرد. وعندما ينتهي مكون الاختبار الرئيسي، ينشر طلب لإنهاء مكون الاختبار الرئيسي مع الحكم المحلي لمكون الاختبار الرئيسي للتمكن من اشتقاق حكم اختبار عالمي والتمكن من إنهاء المعلومات عن الاختبار المجرد.

1.5.3.11 رسم تخطيطي تتابعي



الشكل Z.145/20 – سيناريو الاستخدام – مناولة مكون لتنفيذ اختبار مجرد في تحكم

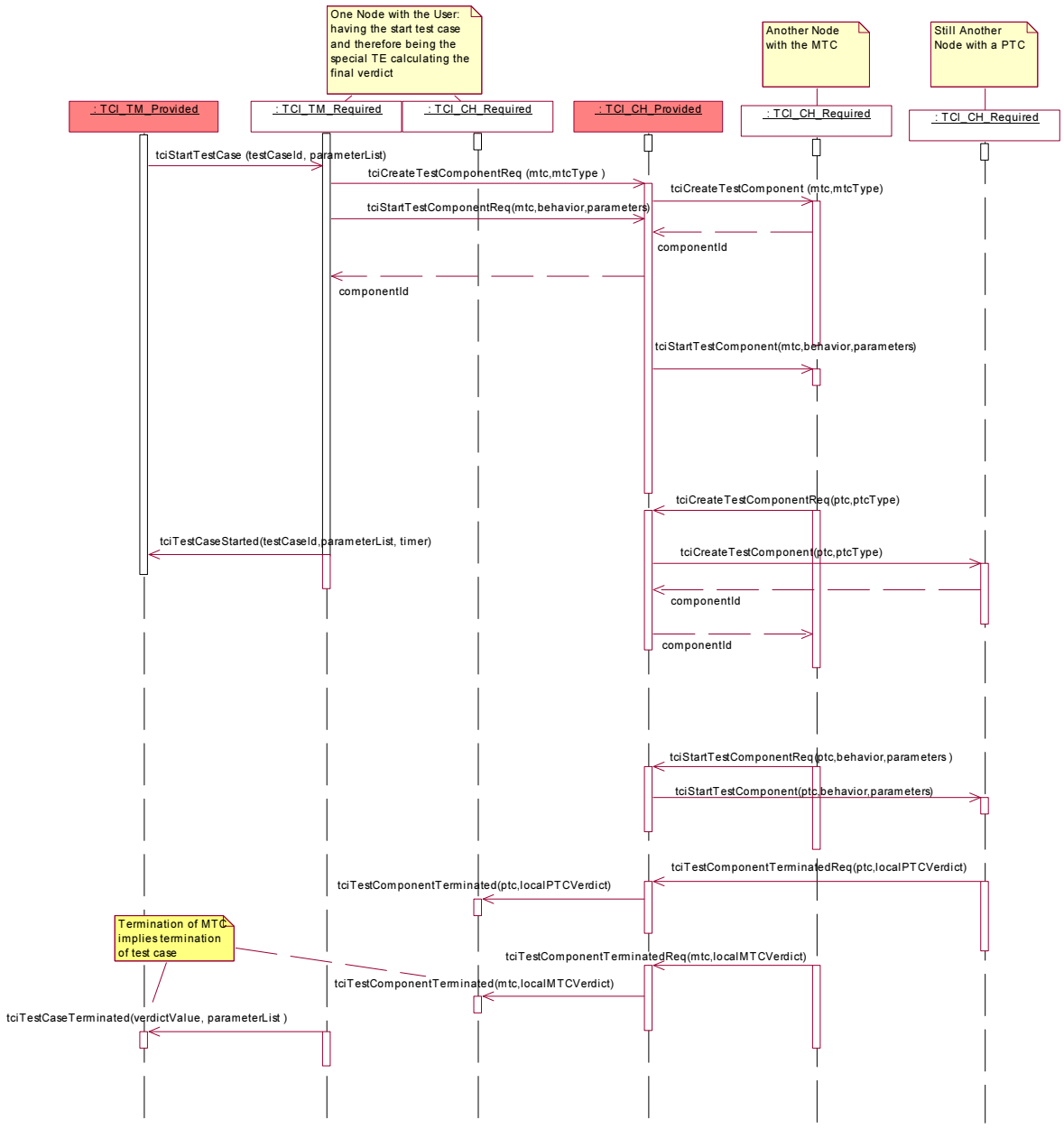
2.5.3.11 تقطيع TTCN-3

```
module AModule {
    …
    testcase ATestCase(…)… {
        … //the test case behaviour
    }
    …
    control {
        …
        execute(ATestCase(…));
        …
    }
    …
}
```

6.3.11 سيناريو الاستخدام: مناولة مكون للتنفيذ المباشر لاختبار مجرد

يبين السيناريو في الشكل 21 كيفية مناولة مكونات اختبار عند تنفيذ اختبار مجرد مباشرة، أي، خارج جزء التحكم. وعندما يبدأ الاختبار المجرد، ينشأ مكون الاختبار الرئيسي ويكون سلوك الاختبار المجرد قد بدأ على مكون الاختبار الرئيسي أولاً. وعندما يستخدم مكون اختبار متوازي في اختبار مجرد، يتم مناولته بنفس الطريقة: يبدأ مكون الاختبار المتوازي أولاً: معطياً طلب إنشاء مكون اختبار إلى كيان TCI-CH، الذي ينشر إنشاء مكون الاختبار على TE الذي ينشأ فيه مكون اختبار ويعاد معرف مكون الاختبار المتوازي الذي أنشئ. ويستخدم المعرف لبدء سلوك PTC لبدء العملية. وعندما ينهي PTC تنفيذه، يصدر طلب إنهاء مكون اختبار مع حكم اختبار محلي ليخطر TCI-CH عن هذا الإنهاء. ويتم نفس الشيء عند إنتهاء مكون الاختبار الرئيسي. وبالإضافة إلى ذلك، يؤدى أنهاء مكون الاختبار الرئيسي إلى الإنهاء الشامل للاختبار المجرد.

1.6.3.11 رسم تخطيطي تتابعي



الشكل Z.145/21 – سيناريو الاستخدام – مناولة مكون للتنفيذ المباشر لاختبار مجرد

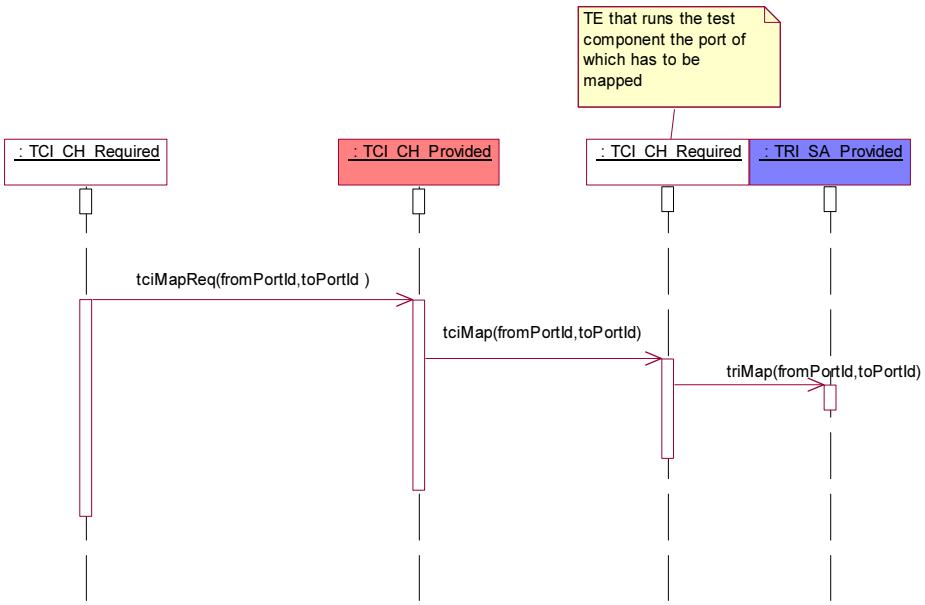
2.6.3.11 تقطيع TTCN-3

```
module AModule {
    …
    function APTCBehaviour(…) runs on APTCType {
        … //the PTC behaviour
    }
    …
    testcase ATestCase(…)… {
        … //the test case behaviour
        var APTCType PTC:= APTCType.create;
        …
        PTC.start(APTCBehaviour(…));
        …
    }
    …
}
```

7.3.11 سيناريو الاستخدام: انتشار عمليات تقابل/توصيل

يبين السيناريو في الشكل 22 كيفية تقابل منافذ. وينشر طلب تقابل منفذ على TE حيث يؤدى التقابل في النهاية. ويعمل انتشار طلبات توصيل تماثلياً.

1.7.3.11 رسم تخطيطي تتابعي



الشكل Z.145/22 – سيناريو الاستخدام – انتشار تقابل

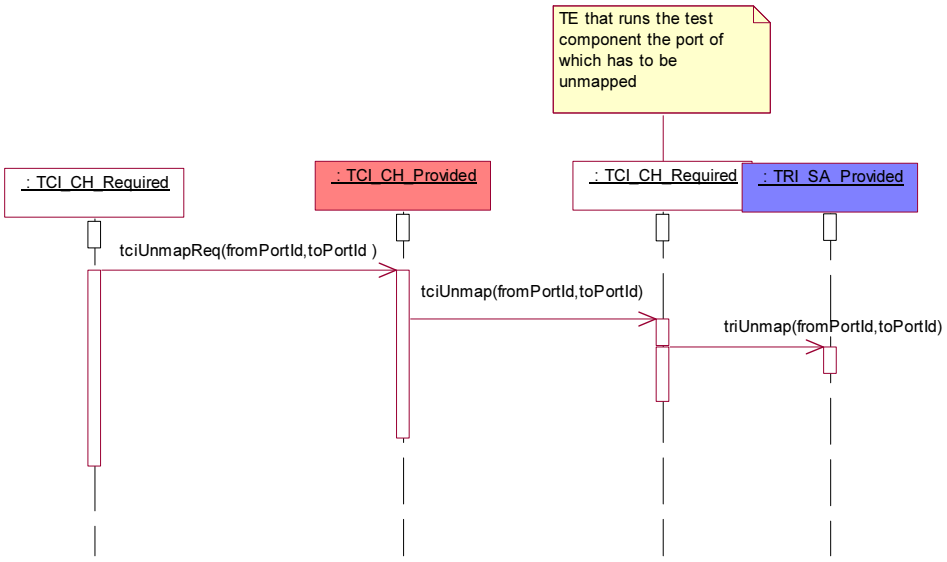
2.7.3.11 تقطيع TTCN-3

```
module AModule {
    …
    type port A { … }
    type component CA { port A a }
        type component CB { port A a }
    …
    testcase ATestCase(…)runs on CA system CB {
        var CA ptc := CA.create;
        … //the test case behaviour
        map(ptc:a,system:a);
        …
    }
    …
}
```

8.3.11 سيناريو الاستخدام: انتشار عمليات فك تقابل/فك توصيل

يبين السيناريو في الشكل 23 كيفية فك تقابل منافذ. وينشر طلب فك تقابل منفذ على TE حيث يؤدى فك التقابل في النهاية. ويعمل انتشار طلبات فك توصيل تماثلياً.

1.8.3.11 رسم تخطيطي تتابعي



الشكل Z.145/23 – سيناريو الاستخدام – انتشار فك تقابل

2.8.3.11 تقطيع TTCN-3

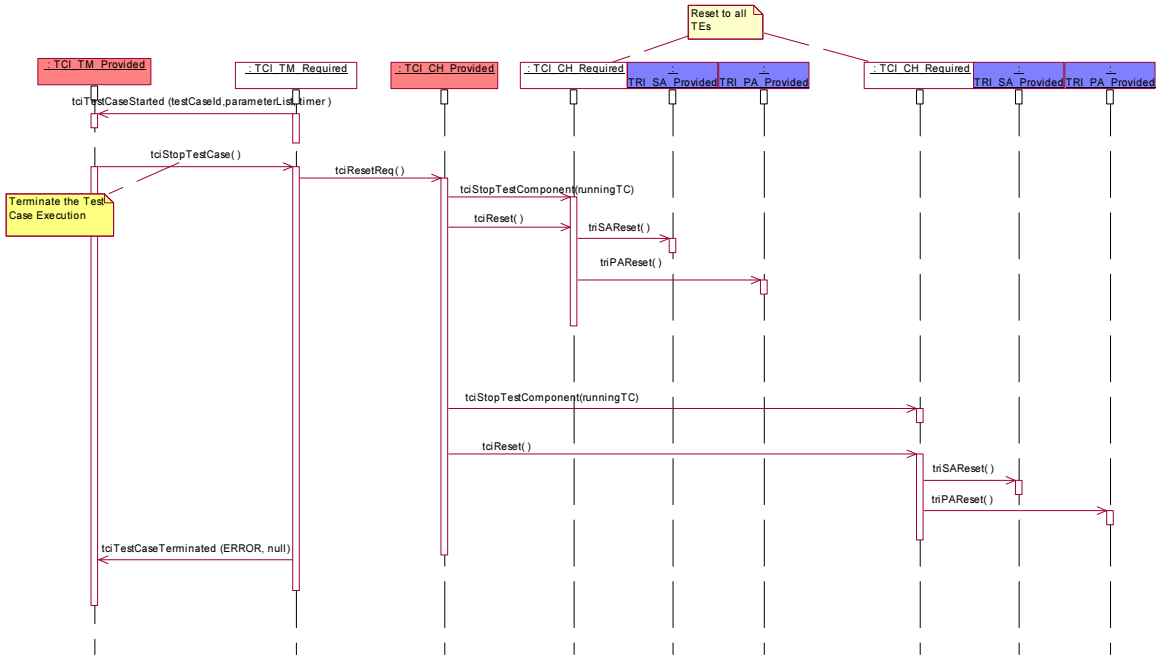
```
module AModule {
    …
    type port A { … }
    type component CA { port A a }
    type component CB { port A a }
    …
    testcase ATestCase(…)runs on CA system CB {
        var CA ptc := CA.create;
        … //the test case behaviour
        unmap(ptc:a,system:a);
        …
    }
    …
}
```

4.11 إنهاء اختبارات مجردة وتحكم

1.4.11 سيناريو الاستخدام: وقف اختبار مجرد

يبين السيناريو في الشكل 24 كيف يتوقف اختبار مجرد من إدارة الاختبار خلال تنفيذ اختبار مجرد. وبمجرد تسلم TM معلومات عن بدء اختبار مجرد، يمكن طلب وقف اختبار مجرد حتى استقبال معلومات عن أن الاختبار المجرد قد إنتهي. وعند توقف اختبار مجرد، يجرى وقف جميع المكونات المتوازية للاختبار ويعاد تدميث نظام الاختبار.

1.1.4.11 رسم تخطيطي تتابعي



الشكل Z.145/24 – سيناريو الاستخدام – وقف اختبار مجرد

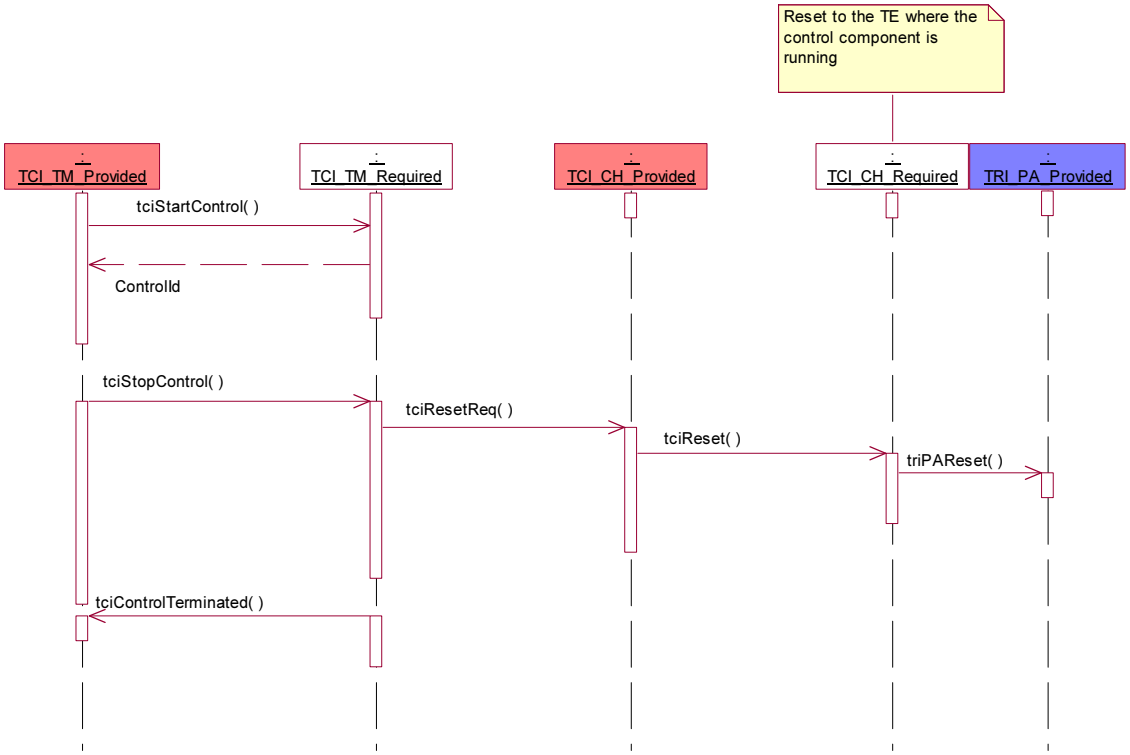
2.1.4.11 تقطيع TTCN-3

لا توجد شفرة TTCN-3 تتعلق بكيفية اختيار TM لتنفيذ إنهاء اختبار مجرد. وهذا خارج مدى TTCN-3.

2.4.11 سيناريو الاستخدام: وقف تحكم

يبين السيناريو في الشكل 25 كيف يتوقف جزء التحكم من إدارة الاختبار خلال تنفيذ جزء التحكم. ويمكن وقف جزء التحكم بين بداية التحكم وانتهائه. وإذا استقبل جزء التحكم طلب وقف اختبار مجرد بينما يجرى تنفيذ اختبار مجرد، يتوقف تنفيذ الاختبار المجرد. وفضلاً عن ذلك، يعاد تدميث نظام الاختبار كما ورد في الشكل 24.

1.2.4.11 رسم تخطيطي تتابعي



الشكل Z.145/25 – سيناريو الاستخدام – وقف تحكم

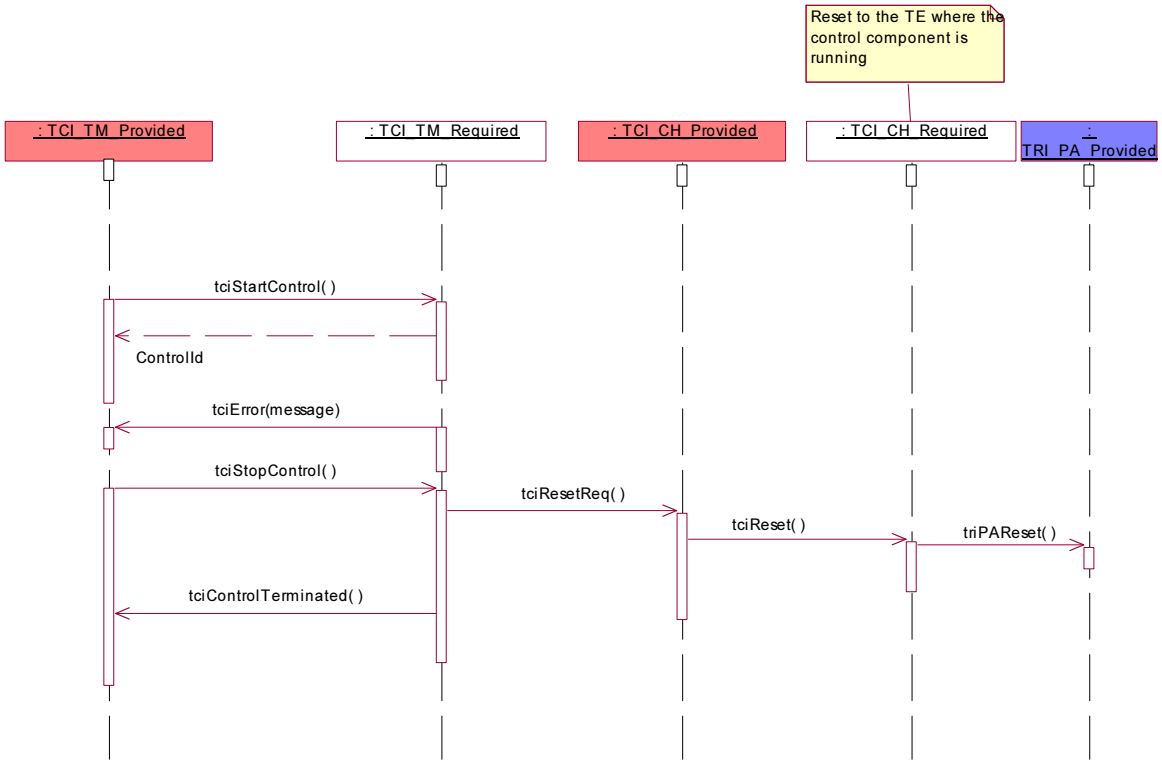
2.2.4.11 تقطيع TTCN-3

إن وقف جزء التحكم من إدارة الاختبار هو خارج مدى TTCN-3 بحيث لا يوجد تقطيع TTCN-3.

3.4.11 سيناريو الاستخدام: إنهاء تحكم بعد خطأ

يبين السيناريو في الشكل 26 مناولة حالات خطأ خلال تنفيذ جزء التحكم عندما لا ينفذ اختبار مجرد. وتبلغ إدارة الاختبار بحالة الخطأ ويتعين إنهاء تنفيذ جزء التحكم بوضوح. وعند إنهاء جزء التحكم، يعاد تدميث نظام الاختبار.

1.3.4.11 رسم تخطيطي تتابعي



الشكل Z.145/26 – سيناريو الاستخدام – إنهاء تحكم بعد خطأ

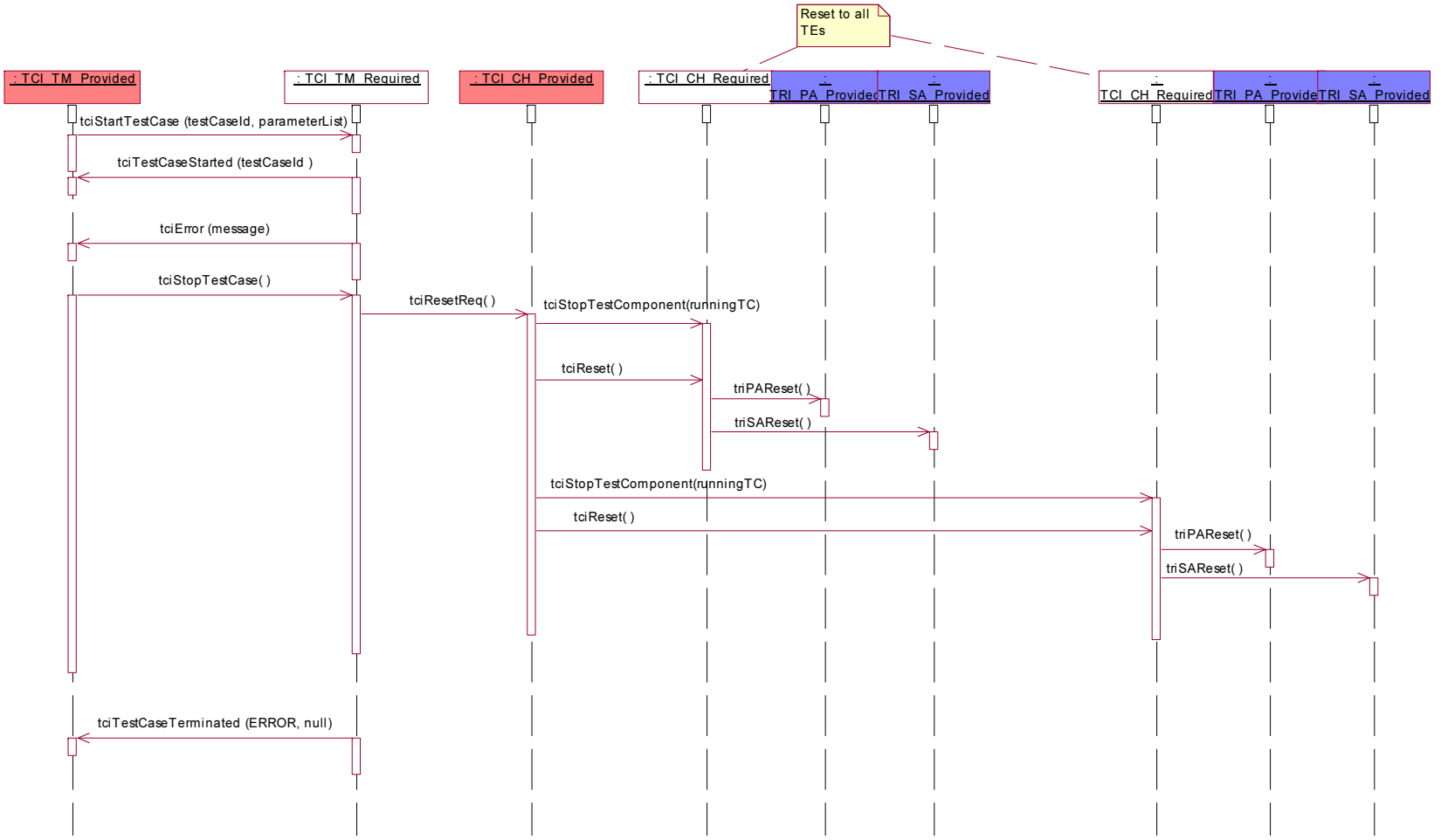
2.3.4.11 تقطيع TTCN-3

لا يوجد تقطيع TTCN-3 لهذا السيناريو نظراً لأن حالات خطأ هي حالات استثنائية في نظام اختبار وليس مفهوم TTCN-3 باعتباره كذلك. وبالأحري، يصف علم دلالات TTCN-3 حالات مختلفة لخطأ محتمل في نظام اختبار.

4.4.11 سيناريو الاستخدام: إنهاء اختبار مجرد بعد خطأ

يبين السيناريو في الشكل 27 مناولة حالات خطأ خلال تنفيذ مباشر لاختبار مجرد. وتبلغ إدارة الاختبار بحالة الخطأ. ويتعين على TM إنهاء تنفيذ اختبار مجرد بوضوح. وعند وقف اختبار مجرد، توقف مكونات اختبار متوازية ويعاد تدميث نظام الاختبار.

1.4.4.11 رسم تخطيطي تتابعي



الشكل Z.145/27 – سيناريو الاستخدام – إنهاء اختبار مجرد بعد خطأ

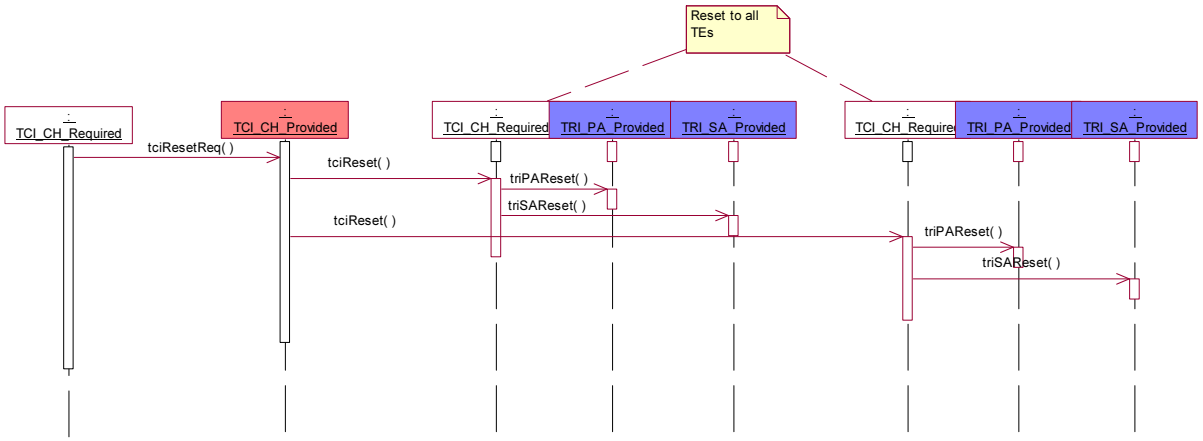
2.4.4.11 رسم تخطيطي تتابعي

لا يوجد تقطيع TTCN-3 لهذا السيناريو نظرا لأن حالات خطأ هي حالات استثنائية في نظام اختبار وليس مفهوم TTCN-3 باعتباره كذلك. وبالأحري، يصف علم دلالات TTCN-3 حالات مختلفة لخطأ محتمل في نظام اختبار.

5.4.11 سيناريو الاستخدام: إعادة تدميث

يبين السيناريو في الشكل 28 إعادة تدميث نظام الاختبار. وفي تلك الحالة، يعاد تدميث جميع TEs المتضمنة مع مكيفات نظام TRI (SA) ومكيفات منصة (PA).

1.5.4.11 رسم تخطيطي تتابعي



الشكل Z.145/28 – سيناريو الاستخدام – إعادة تدميث

1.5.4.11 تقطيع TTCN-3

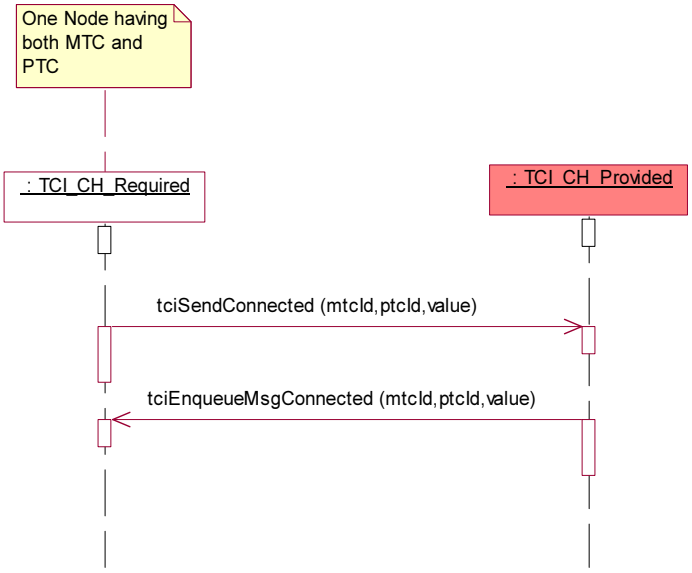
لا يوجد تقطيع TTCN-3 لهذا السيناريو نظراً لأن إعادة التدميث المطلوبة بعد حالات خطأ هي حالات استثنائية في نظام اختبار وليس مفهوم TTCN-3 باعتباره كذلك.

5.11 الاتصالات

1.5.11 سيناريو الاستخدام: الاتصالات بين المكونات المحلية

يبين السيناريو في الشكل 29 الاتصالات بين مكونات اختبار (مكونات الاختبار الرئيسية أو مكونات الاختبار المتوازية)، التي توجد على نفس العقدة. ويعطي طلب اتصالات إلى TCI-CH، التي تقرر أين يكون صف انتظار مقاس هذه الاتصالات. وفي هذه الحالة، تتم الاتصالات محلياً عبر TE على نفس العقدة. ويبين السيناريو اتصالات قائمة على رسالة باستخدام عملية إرسال – والسيناريو هو نفسه لنداء وإجابة وتنفيذ عمليات.

1.1.5.11 رسم تخطيطي تتابعي



الشكل 29/Z.145 – سيناريو الاستخدام – الاتصالات بين المكونات المحلية

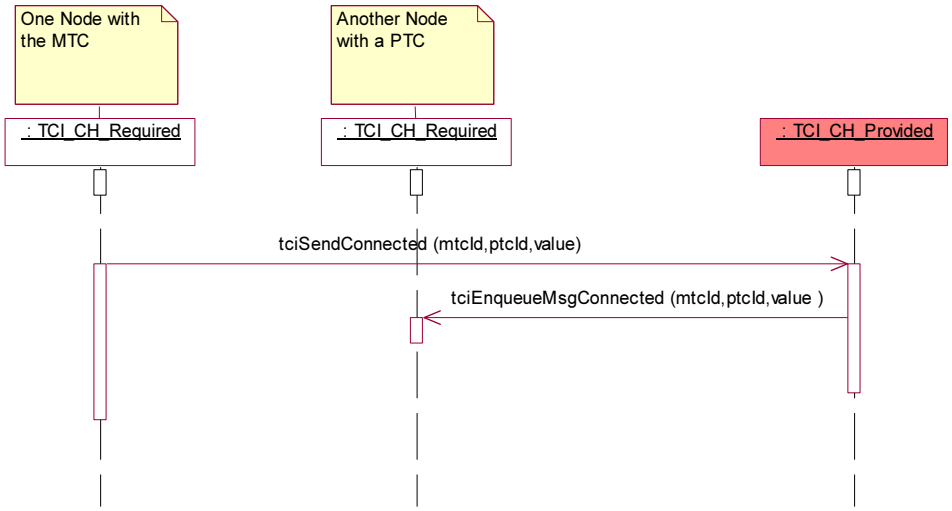
2.1.5.11 تقطيع TTCN-3

```
module AModule {
    …
    type port APortType message { … }
    …
    type component ATCType {
        …
        APortType APort;
        …
    }
    …
    template AType AMessageTemplate { … }
    …
    function APTCBehaviour(…) runs on APTCType {
        … //the PTC behaviour
    …
    }
    …
    testcase ATestCase(…) runs on ATCType… {
        … //the test case behaviour
        var ATCType PTC1:= ATCType.create;
        connect(PTC1:APort,mtc:APort);
        …
        PTC1.start(APTCBehaviour(…));
        APort.send(AMessageTemplate); //sending data to a test component
        …
    }
    …
}
```

2.5.11 سيناريو الاستخدام: الاتصالات بين العقد وبين مكونات اختبار

يبين السيناريو في الشكل 30 الاتصالات بين مكونات اختبار (مكونات الاختبار الرئيسية أو مكونات الاختبار المتوازية)، التي توجد على عقد مختلفة. ويعطي طلب اتصالات إلى TCI-CH، التي تقرر أين يكون صف انتظار مقاس هذه الاتصالات. وفي هذه الحالة، تتم الاتصالات عن بعد عبر TE على عقدة أخرى. ويبين السيناريو اتصالات قائمة على رسالة باستخدام عملية إرسال – السيناريو هو نفسه لنداء وإجابة وتنفيذ عمليات.

1.2.5.11 رسم تخطيطي تتابعي



الشكل Z.145/30 – سيناريو الاستخدام – الاتصالات بين العقد وبين مكونات اختبار

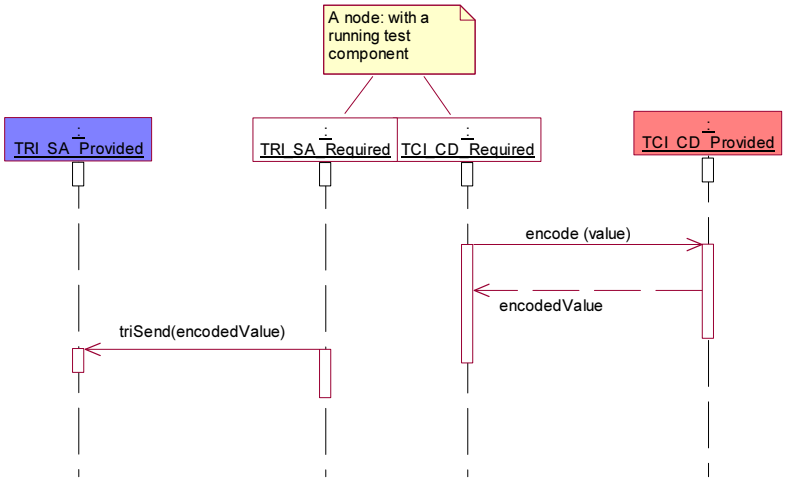
2.2.5.11 تقطيع TTCN-3

```
module AModule {
    …
    type port APortType message { … }
    …
    type component ATCType {
        …
        APortType APort;
        …
    }
    …
    template AType AMessageTemplate { … }
    …
    function APTCBehaviour(…) runs on APTCType {
        … //the PTC behaviour
    …
    }
    …
    testcase ATestCase(…) runs on ATCType… {
        … //the test case behaviour
        var ATCType PTC1:= ATCType.create;
        connect(PTC1:APort,mtc:APort);
        …
        PTC1.start(APTCBehaviour(…));
        APort.send(AMessageTemplate); //sending data to a test component
        …
    }
    …
}
```

3.5.11 سيناريو الاستخدام: التشفير

يبين السيناريو في الشكل 31 تشفير معطيات، الذي يرسل إلى SUT. وتستقبل المعطيات المشفرة من كيان تشفير/فك تشفير عبر TCI-CH. وترسل القيمة المشفرة إلى SUT عبر TRI-SA. والسيناريو هو نفسه للنداء والإجابة وتنفيذ العمليات.

1.3.5.11 رسم تخطيطي تتابعي



الشكل Z.145/31 – سيناريو الاستخدام – التشفير

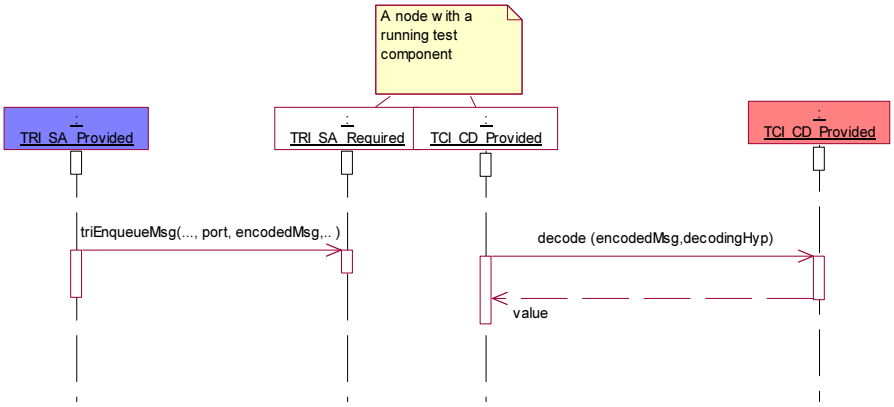
2.3.5.11 تقطيع TTCN-3

```
module AModule {
    …
    type port APortType message { … }
    …
    type component APTCType {
        …
        APortType APort;
        …
    }
    …
    template AType AMessageTemplate { … }
    …
    testcase ATestCase(…) runs on APTCType system APTCType {
        … //the test case behaviour
        map(mtc:APort,system:APort);
        …
        APort.send(AMessageTemplate); //sending data to the SUT
        …
    }
    …
} with { encoding = '…' }
```

4.5.11 سيناريو الاستخدام: فك تشفير

يبين السيناريو في الشكل 32 فك تشفير معطيات، حيث يستقبل من SUT عبر TRI-SA. وتستقبل المعطيات المفكك تشفيرها من كيان تشفير/فك تشفير عبر TCI-CD. والسيناريو هو نفسه للاستقبال ولمعالجة نداء ولمعالجة إجابة وللحصول وللتحقق من عمليات.

1.4.5.11 رسم تخطيطي تتابعي



الشكل Z.145/32 – سيناريو الاستخدام – فك تشفير

2.4.5.11 تقطيع TTCN-3

```
module AModule {
    …
    type port APortType message { … }
    …
    type component APTCType {
        …
        APortType APort;
        …
    }
    …
    template AType AMessageTemplate { … }
    …
    testcase ATestCase(…) runs on APTCType system APTCType {
        … //the test case behaviour
        map(mtc:APort,system:APort);
        …
        APort.receive(AMessageTemplate); //receiving data from the SUT
        …
    }
    …
} with { encoding = '…' }
```

الملحق A

مواصفة IDL لـــ TCI

يعرف هذا الملحق السطوح البينية لتحكم TTCN-3 باستخدام IDL.

```
// ********************************************************************************
// * Interface definitions for the TTCN-3 Control Interfaces
// ********************************************************************************

module tciInterface {

  /* Forward declaration */
  interface Value;
  interface Type;

  // ********************************************************************************
  // * Data types taken from the TRI definitions
  // ********************************************************************************

  // Connection
  native TriPortIdType ;
  native TriPortIdListType;
  native TriComponentIdType ;
native TriComponentIdListType;

  // Communications
  native TriMessageType;
  native TriParameterType;
  native TriParameterListType;
  native TriAddressType;
  native TriAddressListType;
  native TriExceptionType;
  native TriSignatureIdType;

  // Miscellaneous
  native TriStatusType;
  native TriTimerIdType;
native TriTimerDurationType;

  // ********************************************************************************
  // * General Abstract Data Types
  // ********************************************************************************

  // Basic definitions
  native TBoolean;
  native TFloat;
  native TChar;
  native TInteger;
  native TString;
  native TUniversalChar;
  typedef sequence <TString> TStringSeq;
  native TObjid;

  struct QualifiedName {
    TString moduleName;
    TString baseName;
  };

  // General TCI abstract data types
  typedef QualifiedName TciBehaviourIdType;
  typedef QualifiedName TciModuleIdType;
  typedef QualifiedName TciModuleParameterIdType;
  typedef QualifiedName TciTestCaseIdType;

  enum TciParameterPassingModeType {
    IN_MODE,
    OUT_MODE,
    INOUT_MODE
  };

  struct TciParameterType {
    TciModuleParameterIdType parameterName;
    Value parameterValue;
```

```
  TciParameterPassingModeType mode;
};

typedef sequence <TciParameterType> TciParameterListType;

struct TciParameterTypeType {
  Type parameterType;
  TciParameterPassingModeType mode;
};

typedef sequence <TciParameterTypeType> TciParameterTypeListType;

struct TciModuleParameterType {
  TciModuleParameterIdType parameterName;
  Value defaultValue;
};

typedef sequence <TciModuleIdType> TciModuleIdListType ;

typedef sequence <TciModuleParameterType> TciModuleParameterListType;

typedef sequence <TciTestCaseIdType> TciTestCaseIdListType;


enum TciTestComponentKindType {
  MTC,
  PTC,
  CONTROL
};

enum TciTypeClassType {
  ADDRESS_CLASS,
  ANYTYPE_CLASS,
  BITSTRING_CLASS,
  BOOLEAN_CLASS,
  CHAR_CLASS,
  CHARSTRING_CLASS,
  COMPONENT_CLASS,
  ENUMERATED_CLASS,
  FLOAT_CLASS,
  HEXSTRING_CLASS,
  INTEGER_CLASS,
  OBJID_CLASS,
  OCTETSTRING_CLASS,
  RECORD_CLASS,
  RECORDOF_CLASS,
  SET_CLASS,
  SETOF_CLASS,
  UNION_CLASS,
  UNIVERSALCHAR_CLASS,
  UNIVERSALCHARSTRING_CLASS,
  VERDICT_CLASS
};


// ***************************************************************************
// * Abstract TTCN-3 Data Types And Values
// ***************************************************************************

// Abstract data type "Type"
interface Type {
  TciModuleIdType  getDefiningModule ();
  TString          getName ();
  TciTypeClassType getTypeClass ();
  Value            newInstance ();
  TString          getTypeEncoding ();
  TString          getTypeEncodingVariant ();
  TStringSeq       getTypextension ();
};

// Abstract TTCN-3 Values
interface Value {
  TString  getValueEncoding ();
  TString  getValueEncodingVariant ();
  Type     getType ();
  TBoolean  notPresent ();
};

interface RecordOfValue : Value {
  Value    getField (in TInteger position);
```

```
  void     setField (
           in TInteger position,
           in Value value
             );
  void     appendField (in Value value);
  Type     getElementType ();
  TInteger getLength ();
  void     setLength (in TInteger len);
};

interface RecordValue : Value {
  Value      getField (in TString fieldName);
  void       setField (
           in TString fieldName,
           in Value value
           );
  TStringSeq getFieldNames ();
};

interface VerdictValue : Value {
  TInteger getVerdict ();
  void     setVerdict (in TInteger verdict);
};

interface BitstringValue : Value {
  TString   getString ();
  void      setString (in TString value);
  TInteger  getBit (in TInteger position);
  void      setBit (
           in TInteger position,
           in TInteger value
           );
  TInteger  getLength ();
  void      setLength (in TInteger len);
};

interface OctetstringValue : Value {
  TString  getString ();
  void     setString (in TString value);
  TInteger getOctet (in TInteger position);
  void     setOctet (
           in TInteger position,
           in TInteger value
           );
  TInteger getLength ();
  void     setLength (in TInteger len);
};

interface FloatValue : Value {
  TFloat getFloat ();
  void  setFloat (in TFloat value);
};

interface HexstringValue : Value {
  TString  getString ();
  void     setString (in TString value);
  TInteger getHex (in TInteger position);
  void     setHex (
           in TInteger position,
           in TInteger value
           );
  TInteger getLength ();
  void     setLength (in TInteger len);
};

interface ObjidValue : Value {
  TObjid getObjid ();
  void   setObjid (in TObjid value);
};

interface EnumeratedValue : Value {
  void    setEnum (in TString enumValue);
  TString getEnum ();
};

interface IntegerValue : Value {
  TInteger getInt ();
  void     setInt (in TInteger value);
};
```

```
interface CharValue : Value {
  TChar getChar ();
  void  setChar (in TChar value);
};

interface CharstringValue : Value {
  TString  getString ();
  void     setString (in TString value);
  TChar    getChar   (in TInteger position);
  void     setChar   (
           in TInteger position,
           in TChar value
           );
  TInteger getLength ();
  void     setLength (in TInteger len);
};

interface BooleanValue : Value {
  TBoolean getBoolean ();
  void    setBoolean (in TBoolean value);
};

interface UniversalCharValue : Value {
  TUniversalChar getUniversalChar ();
  void           setUniversalChar (in TUniversalChar value);

};

interface UniversalCharstringValue : Value {
  TString        getString ();
  void           setString (in TString value);
  TUniversalChar getChar   (in TInteger position);
  void           setChar   (
           in TInteger position,
           in TUniversalChar value
           );
  TInteger       getLength ();
  void           setLength (in TInteger len);
};

interface UnionValue : Value {
  Value      getVariant  (in TString variantName);
  void       setVariant  (
           in TString variantName,
           in Value value
           );
  TString    getPresentVariantName ();
  TStringSeq getVariantNames ();

};

// ********************************************************************************
// * Abstract Logging Types
// ********************************************************************************

interface TciValueTemplate : Value {
    boolean isOmit ();
    boolean isAny();
    boolean isAnyOrOmit();
    TString getTemplateDef();
};

interface TciNonValueTemplate {
    boolean isAny();
    boolean isAll();
    TString getTemplateDef();
};

typedef sequence <Value> TciValueList;

struct TciValueDifference
{
  TString desc;
  Value val;
  TciValueTemplate tmpl;
};

typedef sequence <TciValueDifference> TciValueDifferenceList;
```

```
// ********************************************************************************
// Coding Decoding Interface
// -  Required
// ********************************************************************************

interface TCI_CD_Required {
  Type getTypeForName (in TString typeName);
  Type getInteger ();
  Type getFloat ();
  Type getBoolean ();
  Type getChar ();
  Type getUniversalChar ();
  Type getObjid ();
  Type getCharstring ();
  Type getUniversalCharstring ();
  Type getHexstring ();
  Type getBitstring ();
  Type getOctetstring ();
  Type getVerdict ();
  void tciErrorReq (in TString message);
};

// ********************************************************************************
// Coding Decoding interface
// -  Provided
// ********************************************************************************

interface TCI_CD_Provided {
  Value decode (
        in TriMessageType message,
        in Type decodingHypothesis
        );
  TriMessageType encode (in Value value);
};


// ********************************************************************************
// Test Management Interface
// -  Required
// ********************************************************************************

interface TCI_TM_Required : TCI_CD_Required {
  void tciRootModule (in TciModuleIdType moduleName);
  TciModuleIdListType getImportedModules();
  TciModuleParameterListType tciGetModuleParameters (in TciModuleIdType moduleName);
  TciTestCaseIdListType tciGetTestCases ();
  TciParameterTypeListType tciGetTestCaseParameters (
          in TciTestCaseIdType testCaseId
          );
  TriPortIdListType tciGetTestCaseTSI (
          in TciTestCaseIdType testCaseId
          );
  void tciStartTestCase (
          in TciTestCaseIdType testCaseId,
          in TciParameterListType parameterList
          );
  void tciStopTestCase ();
  TriComponentIdType tciStartControl ();
  void tciStopControl ();
};

// ********************************************************************************
// Test Management Interface
// -  Provided
// ********************************************************************************

interface TCI_TM_Provided {
  void tciTestCaseStarted (
          in TciTestCaseIdType testCaseId,
          in TciParameterListType parameterList,
          in TFloat timer
          );
  void tciTestCaseTerminated (
          in VerdictValue verdict,
          in TciParameterListType parameterList
          );
  void tciControlTerminated ();
  Value tciGetModulePar (
          in TciModuleParameterIdType parameterId
          );
```

```
  void tciLog (
       in TriComponentIdType testComponentId,
       in TString message
       );
  void tciError (in TString message);
};


// ********************************************************************************
// Component Handling Interface
// -  Required
// ********************************************************************************

interface TCI_CH_Required : TCI_CD_Required {
  void tciEnqueueMsgConnected (
          in TriPortIdType sender,
          in TriComponentIdType receiver,
          in Value receivedMessage
          );
  void tciEnqueueCallConnected (
          in TriPortIdType sender,
          in TriComponentIdType receiver,
          in TriSignatureIdType signature,
          in TciParameterListType parameterList
          );
  void tciEnqueueReplyConnected (
          in TriPortIdType sender,
          in TriComponentIdType receiver,
          in TriSignatureIdType signature,
          in TciParameterListType parameterList,
          in Value returnValue
          );
  void tciEnqueueRaiseConnected (
          in TriPortIdType sender,
          in TriComponentIdType receiver,
          in TriSignatureIdType signature,
          in Value except
          );
  TriComponentIdType tciCreateTestComponent (
          in TciTestComponentKindType kind,
          in Type componentType,
          in String name
          );
  void tciStartTestComponent (
          in TriComponentIdType comp,
          in TciBehaviourIdType behavior,
          in TciParameterListType parameterList
          );
  void tciStopTestComponent (
          in TriComponentIdType comp
          );
  void tciConnect (
         in TriPortIdType fromPort,
         in TriPortIdType toPort
         );
  void tciDisconnect (
         in TriPortIdType fromPort,
         in TriPortIdType toPort
         );
  void tciTestComponentTerminated (
         in TriComponentIdType comp,
         in VerdictValue verdict
         );
  TBoolean tciTestComponentRunning (
         in TriComponentIdType comp
         );
  TriComponentIdType tciGetMTC ();
  void tciMap (
         in TriPortIdType fromPort,
         in TriPortIdType toPort
         );
  void tciUnmap (
         in TriPortIdType fromPort,
         in TriPortIdType toPort
         );
  void tciExecuteTestCase (
         in TciTestCaseIdType testCaseId,
         in TriPortIdListType tsiPortList
         );
```

```
  TBoolean tciTestComponentDone (
         in TriComponentIdType comp
         );
  void tciReset ();
};

// *****************************************************************************
// Component Handling Interface
// -  Provided
// *****************************************************************************

interface TCI_CH_Provided {
  void tciSendConnected (
          in TriPortIdType sender,
          in TriComponentIdType receiver,
          in Value sendMessage
          );
  void tciSendConnectedBC (
         in TriPortIdType sender,
         in Value sendMessage
         );
  void tciSendConnectedMC (
         in TriPortIdType sender,
         in TriComponentIdListType receivers,
         in Value sendMessage
         );

  void tciCallConnected (
          in TriPortIdType sender,
          in TriComponentIdType receiver,
          in TriSignatureIdType signature,
          in TciParameterListType parameterList
          );
  void tciCallConnectedBC (
        in TriPortIdType sender,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList
        );
  void tciCallConnectedMC (
        in TriPortIdType sender,
        in TriComponentIdListType receivers,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList
        );

  void tciReplyConnected (
          in TriPortIdType sender,
          in TriComponentIdType receiver,
          in TriSignatureIdType signature,
          in TciParameterListType parameterList,
          in Value returnValue
          );
  void tciReplyConnectedBC (
         in TriPortIdType sender,
         in TriSignatureIdType signature,
         in TciParameterListType parameterList,
         in Value returnValue
         );
  void tciReplyConnectedMC (
         in TriPortIdType sender,
         in TriComponentIdListType receivers,
         in TriSignatureIdType signature,
         in TciParameterListType parameterList,
         in Value returnValue
         );

  void tciRaiseConnected (
          in TriPortIdType      sender,
          in TriComponentIdType receiver,
          in TriSignatureIdType signature,
          in Value except
          );
  void tciRaiseConnectedBC (
        in TriPortIdType sender,
        in TriSignatureIdType signature,
        in Value except
        );
  void tciRaiseConnectedMC (
        in TriPortIdType sender,
        in TriComponentIdListType receivers,
```

```
        in TriSignatureIdType signature,
        in Value except
        );

  TriComponentIdType tciCreateTestComponentReq (
          in TciTestComponentKindType kind,
          in Type componentType,
          in String name
          );
  void tciStartTestComponentReq (
          in TriComponentIdType comp,
          in TciBehaviourIdType behavior,
          in TciParameterListType parameterList
          );
  void tciStopTestComponentReq (
          in TriComponentIdType comp
          );
  void tciConnectReq (
          in TriPortIdType fromPort,
          in TriPortIdType toPort
          );
  void tciDisconnectReq (
          in TriPortIdType fromPort,
          in TriPortIdType toPort
          );
  void tciTestComponentTerminatedReq (
          in TriComponentIdType comp,
          in VerdictValue verdict
          );
  TBoolean tciTestComponentRunningReq (
          in TriComponentIdType comp
          );
  TriComponentIdType tciGetMTCReq ();
  void tciMapReq (
         in TriPortIdType fromPort,
         in TriPortIdType toPort
         );
  void tciUnmapReq (
         in TriPortIdType fromPort,
         in TriPortIdType toPort
         );
  void tciExecuteTestCaseReq (
          in TciTestCaseIdType testCaseId,
          in TriPortIdListType tsiPortList
          );
  void tciResetReq ();
  TBoolean tciTestComponentDoneReq (
          in TriComponentIdType comp
          );
};

// *****************************************************************************
// Test Logging Interface
// -  Provided
// *****************************************************************************

interface TCI_TL_Provided {
  void tliTcExecute(
         in TString am, in TInteger ts, in TString src, in TInteger line,
         in TriComponentIdType c, in TciTestCaseIdType tcId,
         in TriParameterListType pars, in TriTimerDurationType dur
         );
  void tliTcStart(
         in TString am, in TInteger ts, in TString src, in TInteger line,
         in TriComponentIdType c, in TciTestCaseIdType tcId,
         in TriParameterListType pars, in TriTimerDurationType dur
         );
  void tliTcStop(
         in TString am, in TInteger ts, in TString src, in TInteger line,
         in TriComponentIdType c
         );
  void tliTcStarted(
         in TString am, in TInteger ts, in TString src, in TInteger line,
         in TriComponentIdType c, in TciTestCaseIdType tcId,
         in TriParameterListType pars, in TriTimerDurationType dur
         );
  void tliTcTerminated(
         in TString am, in TInteger ts, in TString src, in TInteger line,
         in TriComponentIdType c, in TciTestCaseIdType tcId,
         in TriParameterListType pars, in VerdictValue outcome);
```

```
void tliCtrlStart(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c
       );
void tliCtrlStop(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c
       );
void tliCtrlTerminated(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c);

void tliMSend_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TriAddressType address, in TriStatusType encoderFailure,
       in TriMessageType msg, in TriStatusType transmissionFailure
       );
 void tliMSend_m_BC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TriStatusType encoderFailure, in TriMessageType msg,
       in TriStatusType transmissionFailure
       );
 void tliMSend_m_MC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TriAddressListType addresses, in TriStatusType encoderFailure,
       in TriMessageType msg, in TriStatusType transmissionFailure
       );

void tliMSend_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TriComponentIdType to, in TriStatusType transmissionFailure
       );
 void tliMSend_c_BC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TriStatusType transmissionFailure
       );
 void tliMSend_c_MC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TriComponentIdListType toList, in TriStatusType transmissionFailure);

void tliMDetected_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in TriMessageType msg,
       in TriAddressType address
       );
void tliMDetected_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TriComponentIdType from
       );
void tliMMismatch_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TciValueTemplate msgTmpl, in TciValueDifferenceList diffs,
        in TriAddressType address, in TciValueTemplate addressTmpl
       );
void tliMMismatch_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TciValueTemplate msgTmpl, in TciValueDifferenceList diffs,
        in TriComponentIdType from, in TciNonValueTemplate fromTmpl
       );
void tliMReceive_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TciValueTemplate msgTmpl, in TriAddressType address,
       in TciValueTemplate addressTmpl
       );
void tliMReceive_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
       in TciValueTemplate msgTmpl, in TriComponentIdType from,
       in TciNonValueTemplate fromTmpl
```

```
       );

void tliPrCall_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TriAddressType address, in TriStatusType encoderFailure,
       in TriParameterListType pars, in TriStatusType transmissionFailure
       );
 void tliPrCall_m_BC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TriStatusType encoderFailure, in TriParameterListType pars,
       in TriStatusType transmissionFailure
       );
 void tliPrCall_m_MC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TriAddressListType addresses, in TriStatusType encoderFailure,
       in TriParameterListType pars,  in TriStatusType transmissionFailure
       );

void tliPrCall_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TriComponentIdType to, in TriStatusType transmissionFailure
       );
 void tliPrCall_c_BC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TriStatusType transmissionFailure
       );
    void tliPrCall_c_MC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TriComponentIdListType toList, in TriStatusType transmissionFailure
       );

void tliPrGetCallDetected_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TriParameterListType pars,
       in TriAddressType address
       );
void tliPrGetCallDetected_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TriComponentIdType from
       );
void tliPrGetCallMismatch_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TciValueTemplate parsTmpl, in TciValueDifferenceList diffs,
       in TriAddressType address, in TciValueTemplate addressTmpl
       );
void tliPrGetCallMismatch_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TciValueTemplate parsTmpl, in TciValueDifferenceList diffs,
       in TriComponentIdType from, in TciNonValueTemplate fromTmpl
       );
void tliPrGetCall_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in TciValueTemplate parsTmpl, in TriAddressType address,
       in TciValueTemplate addressTmpl
       );
void tliPrGetCall_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
```

```
        in TciValueTemplate parsTmpl, in TriComponentIdType from,
        in TciNonValueTemplate fromTmpl
        );

    void tliPrReply_m(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in TciParameterListType parsValue,
        in Value replValue, in TriAddressType address,
        in TriStatusType encoderFailure, in TriParameterType repl,
        in TriStatusType transmissionFailure
        );
     void tliPrReply_m_BC(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in Value parsValue, in Value replValue,
        in TriStatusType encoderFailure, in TriParameterType repl,
        in TriStatusType transmissionFailure
        );
     void tliPrReply_m_MC(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in Value parsValue, in Value replValue,
        in TriAddressListType addresses, in TriStatusType encoderFailure,
        in TriParameterType repl, in TriStatusType transmissionFailure
        );

    void tliPrReply_c(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in TciParameterListType parsValue,
        in Value replValue, in TriComponentIdType to,
        in TriStatusType transmissionFailure
        );
     void tliPrReply_c_BC(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in Value parsValue, in Value replValue,
        in TriStatusType transmissionFailure
        );
     void tliPrReply_c_MC(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in Value parsValue, in Value replValue,
        in TriComponentIdListType toList, in TriStatusType transmissionFailure
        );

    void tliPrGetReplyDetected_m(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in TriParameterType repl,
        in TriAddressType address
        );
    void tliPrGetReplyDetected_c(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in Value replValue,
        in TriComponentIdType from
        );
    void tliPrGetReplyMismatch_m(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in TciParameterListType parsValue,
        in Value replValue, in TciValueTemplate replyTmpl,
        in TciValueDifferenceList diffs, in TriAddressType address,
        in TciValueTemplate addressTmpl
        );
    void tliPrGetReplyMismatch_c(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in TciParameterListType parsValue,
        in Value replValue, in TciValueTemplate replyTmpl,
        in TciValueDifferenceList diffs, in TriComponentIdType from,
        in TciNonValueTemplate fromTmpl
        );
    void tliPrGetReply_m(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in TciParameterListType parsValue,
        in Value replValue, in TciValueTemplate replyTmpl,
```

```
       in TriAddressType address, in TciValueTemplate addressTmpl
       );
void tliPrGetReply_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in Value replValue, in TciValueTemplate replyTmpl,
       in TriComponentIdType from, in TciNonValueTemplate fromTmpl
       );

void tliPrRaise_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in Value excValue, in TriAddressType address, in TriStatusType encoderFailure,
       in TriExceptionType exc, in TriStatusType transmissionFailure
       );
 void tliPrRaise_m_BC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in Value excValue, in TriStatusType encoderFailure, in TriExceptionType exc,
       in TriStatusType transmissionFailure
       );
 void tliPrRaise_m_MC(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in Value excValue, in TriAddressListType addresses,
       in TriStatusType encoderFailure, in TriExceptionType exc,
       in TriStatusType transmissionFailure
       );

void tliPrRaise_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in Value excValue, in TriComponentIdType to,
       in TriStatusType transmissionFailure
       );
void tliPrCatchDetected_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TriExceptionType exc,
       in TriAddressType address
       );
void tliPrCatchDetected_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in Value excValue,
       in TriComponentIdType from
       );
void tliPrCatchMismatch_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in Value excValue, in TciValueTemplate excTmpl,
       in TciValueDifferenceList diffs, in TriAddressType address,
       in TciValueTemplate addressTmpl
       );
void tliPrCatchMismatch_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in Value excValue, in TciValueTemplate excTmpl,
       in TciValueDifferenceList diffs, in TriComponentIdType from,
       in TciNonValueTemplate fromTmpl
       );
void tliPrCatch_m(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in Value excValue, in TciValueTemplate excTmpl,
       in TriAddressType address, in TciValueTemplate addressTmpl);

void tliPrCatch_c(
       in TString am, in TInteger ts, in TString src, in TInteger line,
       in TriComponentIdType c, in TriPortIdType port,
       in TriSignatureIdType signature, in TciParameterListType parsValue,
       in Value excValue, in TciValueTemplate excTmpl,
```

```
        in TriComponentIdType from, in TciNonValueTemplate fromTmpl
        );
    void tliPrCatchTimeoutDetected(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature
        );
    void tliPrCatchTimeout(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port,
        in TriSignatureIdType signature, in TciParameterListType parsValue
        );
    void tliCCreate(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriComponentIdType comp,
        in String name
        );
    void tliCStart(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriComponentIdType comp,
        in TciBehaviourIdType name, in TciParameterListType parsValue
        );
    void tliCRunning(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status
        );
     void tliCAlive(
            in TString am, in TInteger ts, in TString src, in TInteger line,
            in TriComponentIdType c,
            in TriComponentIdType comp, in TBoolean status
            );
     void tliCStop(
            in TString am, in TInteger ts, in TString src, in TInteger line,
            in TriComponentIdType c, in TriComponentIdType comp
            );
     void tliCKill(
            in TString am, in TInteger ts, in TString src, in TInteger line,
            in TriComponentIdType c, in TriComponentIdType comp
            );
     void tliCDoneMismatch(
            in TString am, in TInteger ts, in TString src, in TInteger line,
            in TriComponentIdType c, in TciNonValueTemplate compTmpl
            );
     void tliCKilledMismatch(
            in TString am, in TInteger ts, in TString src, in TInteger line,
            in TriComponentIdType c, in TciNonValueTemplate compTmpl
            );
     void tliCDone(in TString am, in TInteger ts, in TString src, in TInteger line,
            in TriComponentIdType c, in TciNonValueTemplate compTmpl
            );
     void tliCKilled(
            in TString am, in TInteger ts, in TString src, in TInteger line,
            in TriComponentIdType c, in TciNonValueTemplate compTmpl
            );
    void tliCTerminated(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in VerdictValue verdict
        );
    void tliPConnect(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1,
        in TriComponentIdType c2, in TriPortIdType port2
        );
    void tliPDisconnect(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1,
        in TriComponentIdType c2, in TriPortIdType port2
        );
    void tliPMap(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1,
        in TriComponentIdType c2, in TriPortIdType port2
        );
    void tliPUnmap(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1,
        in TriComponentIdType c2, in TriPortIdType port2
        );
    void tliPClear(
        in TString am, in TInteger ts, in TString src, in TInteger line,
```

```
        in TriComponentIdType c, in TriPortIdType port
        );
void tliPStart(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port
        );
void tliPStop(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port
        );
void tliPHalt(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriPortIdType port
        );
void tliEncode(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in Value val, in TriStatusType encoderFailure,
        in TriMessageType msg, in TString codec
        );
void tliDecode(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriMessageType msg,
        in TriStatusType decoderFailure, in Value val, in TString codec
        );
void tliTTimeoutDetected(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriTimerIdType timer
        );
void tliTTimeoutMismatch(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TciNonValueTemplate timerTmpl
        );
void tliTTimeout(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TciNonValueTemplate timerTmpl
        );
void tliTStart(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriTimerIdType timer,
        in TriTimerDurationType dur
        );
void tliTStop(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriTimerIdType timer
        );
void tliTRead(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriTimerIdType timer,
        in TriTimerDurationType elapsed
        );
void tliTRunning(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriTimerIdType timer, in TBoolean status
        );
void tliSEnter(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TString name, in TciParameterListType parsValue,
        in TString kind
        );
void tliSLeave(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TString name, in Value returnValue,
        in TString kind
        );
void tliVar(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TString name, in Value varValue
        );
void tliModulePar(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TString name, in Value parValue
        );
void tliGetVerdict(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in VerdictValue verdict
        );
void tliSetVerdict(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in VerdictValue verdict
        );
```

```
    void tliLog(
           in TString am, in TInteger ts, in TString src, in TInteger line,
           in TriComponentIdType c, in TciValueList log
           );
    void tliAEnter(
           in TString am, in TInteger ts, in TString src, in TInteger line,
           in TriComponentIdType c
           );
    void tliALeave(
           in TString am, in TInteger ts, in TString src, in TInteger line,
           in TriComponentIdType c
           );
    void tliADefaults(
           in TString am, in TInteger ts, in TString src, in TInteger line,
           in TriComponentIdType c
           );
    void tliAActivate(
           in TString am, in TInteger ts, in TString src, in TInteger line,
           in TriComponentIdType c, in TString name, in TciParameterListType pars,
           in Value ref
           );
    void tliADeactivate(
           in TString am, in TInteger ts, in TString src, in TInteger line,
           in TriComponentIdType c, in Value ref
           );
    void tliANomatch(
           in TString am, in TInteger ts, in TString src, in TInteger line,
           in TriComponentIdType c
           );
    void tliARepeat(
           in TString am, in TInteger ts, in TString src, in TInteger line,
           in TriComponentIdType c
           );
    void tliAWait(
           in TString am, in TInteger ts, in TString src, in TInteger line,
           in TriComponentIdType c
           );
    };
};
```

الملحق B

تقابل XML لـــ TCI TL موفر

يعرف هذا الملحق تقابل سطح بيني التسجيل لـــ TCI باستخدام تعاريف تخطيط XML.

1.B تخطيط TCI-TL XML لأنماط بسيطة

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
            xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
            elementFormDefault="qualified">

    <!-- Basic definitions -->
    <xsd:simpleType name="xpath">
        <!-- this string should be XPATH compliant -->
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>

    <xsd:simpleType name="TBoolean">
        <xsd:restriction base="xsd:boolean"/>
    </xsd:simpleType>

    <xsd:simpleType name="TString">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>

    <xsd:simpleType name="TInteger">
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>

    <!-- Miscellaneous -->
    <xsd:simpleType name="TriTimerDurationType">
        <xsd:restriction base="xsd:float"/>
    </xsd:simpleType>

    <xsd:simpleType name="TciParameterPassingModeType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="in"/>
            <xsd:enumeration value="inout"/>
            <xsd:enumeration value="out"/>
        </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="TriStatusType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="TRI_Ok"/>
            <xsd:enumeration value="TRI_Error"/>
        </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="TciStatusType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="TCI_Ok"/>
            <xsd:enumeration value="TCI_Error"/>
        </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

2.B تخطيط TCI-TL XML للأنماط

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
    xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
    xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
    xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
        elementFormDefault="qualified">

    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values.xsd"
     schemaLocation="Values.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes.xsd"
```

```
     schemaLocation="SimpleTypes.xsd"/>

    <!-- Connection -->
    <xsd:complexType name="TriPortIdType">
        <xsd:sequence>
            <xsd:element name="comp" type="Types:TriComponentIdType" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="port" type="Types:Port"  minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Port">
        <xsd:sequence>
            <xsd:element name="id" type="Types:Id" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="index" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="TriComponentIdType">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="null"/>
                <xsd:element name="id" type="Types:Id" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="TriComponentIdListType">
        <xsd:sequence>
            <xsd:element name="comp" type="Types:TriComponentIdType" minOccurs="0"
             maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>

    <!-- Communication -->
    <xsd:complexType name="TriMessageType">
        <xsd:attribute name="val" type="xsd:hexBinary"/>
    </xsd:complexType>

    <xsd:complexType name="TriParameterType">
        <xsd:sequence>
            <xsd:element name="val" type="Values:Value" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="SimpleTypes:TString"/>
        <xsd:attribute name="mode" type="SimpleTypes:TciParameterPassingModeType"/>
    </xsd:complexType>

    <xsd:complexType name="TriParameterListType">
        <xsd:sequence>
            <xsd:element name="par" type="Types:TriParameterType" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="TriExceptionType">
        <xsd:attribute name="val" type="SimpleTypes:TString"/>
    </xsd:complexType>

    <xsd:complexType name="TriSignatureIdType">
        <xsd:attribute name="val" type="SimpleTypes:TString"/>
    </xsd:complexType>

    <xsd:complexType name="TriAddressType">
        <xsd:attribute name="val" type="SimpleTypes:TString"/>
    </xsd:complexType>

    <xsd:complexType name="TriAddressListType">
        <xsd:sequence>
            <xsd:element name="addr" type="Types:TriAddressType" minOccurs="0"
             maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>


    <!-- Miscellaneous -->
    <xsd:complexType name="TriTimerIdType">
        <xsd:sequence>
            <xsd:element name="id" type="Types:Id" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
```

```
    <xsd:complexType name="TriTimerDurationType">
        <xsd:attribute name="val" type="SimpleTypes:TriTimerDurationType"/>
    </xsd:complexType>

    <!-- Basic definitions -->
    <xsd:complexType name="QualifiedName">
        <xsd:attribute name="moduleName" type="SimpleTypes:TString" use="required"/>
        <xsd:attribute name="baseName" type="SimpleTypes:TString" use="required"/>
    </xsd:complexType>

    <!-- general TCI abstract data types -->
    <xsd:complexType name="TciBehaviourIdType">
        <xsd:sequence>
            <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="TciTestCaseIdType">
        <xsd:sequence>
            <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="TciParameterType">
        <xsd:sequence>
            <xsd:element name="val" type="Values:Value" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="SimpleTypes:TString"/>
        <xsd:attribute name="mode" type="SimpleTypes:TciParameterPassingModeType"/>
    </xsd:complexType>

    <xsd:complexType name="TciParameterListType">
        <xsd:sequence>
            <xsd:element name="par" type="Types:TciParameterType" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>

    <!-- general identifier structure for test components, ports and timer -->
    <xsd:complexType name="Id">
        <xsd:sequence>
           <xsd:element name="name" type="SimpleTypes:TString" minOccurs="1" maxOccurs="1"/>
           <xsd:element name="id" type="SimpleTypes:TInteger" minOccurs="1" maxOccurs="1"/>
           <xsd:element name="type" type="SimpleTypes:TString" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

</xsd:schema>
```

3.B تخطيط TCI-TL XML لقيم

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
    xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
    xmlns:Templates="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates"
    xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
    xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
elementFormDefault="qualified">

    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates.xsd"
     schemaLocation="Templates.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types.xsd"
schemaLocation="Types.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes.xsd"
     schemaLocation="SimpleTypes.xsd"/>

    <xsd:attributeGroup name="ValueAtts">
        <xsd:attribute name="name" type="SimpleTypes:TString" use="optional"/>
        <xsd:attribute name="type" type="SimpleTypes:TString" use="optional"/>
        <xsd:attribute name="module" type="SimpleTypes:TString" use="optional"/>
    </xsd:attributeGroup>

    <xsd:complexType name="Value" mixed="true">
        <xsd:choice>
            <xsd:element name="integer" type="Values:IntegerValue"/>
            <xsd:element name="float" type="Values:FloatValue"/>
            <xsd:element name="boolean" type="Values:BooleanValue"/>
            <xsd:element name="objid" type="Values:ObjidValue"/>
```

```
        <xsd:element name="verdicttype" type="Values:VerdictValue"/>
        <xsd:element name="bitstring" type="Values:BitstringValue"/>
        <xsd:element name="hexstring" type="Values:HexstringValue"/>
        <xsd:element name="octetstring" type="Values:OctetstringValue"/>
        <xsd:element name="charstring" type="Values:CharstringValue"/>
        <xsd:element name="universal_charstring" type="Values:UniversalCharstringValue"/>
        <xsd:element name="record" type="Values:RecordValue"/>
        <xsd:element name="record_of" type="Values:RecordOfValue"/>
        <xsd:element name="set" type="Values:SetValue"/>
        <xsd:element name="set_of" type="Values:SetOfValue"/>
        <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
        <xsd:element name="union" type="Values:UnionValue"/>
        <xsd:element name="anytype" type="Values:AnytypeValue"/>
        <xsd:element name="address" type="Values:AddressValue"/>
            </xsd:choice>
        <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>

<!-- general event elements -->
<xsd:complexType name="IntegerValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="FloatValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="BooleanValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="ObjidValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="VerdictValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="BitstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="HexstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="OctetstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
```

```
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="CharstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="UniversalCharstringValue">
    <xsd:simpleContent>
        <xsd:extension base="SimpleTypes:TString">
                <xsd:attributeGroup ref="Values:ValueAtts"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="RecordValue">
    <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="integer" type="Values:IntegerValue"/>
            <xsd:element name="float" type="Values:FloatValue"/>
            <xsd:element name="boolean" type="Values:BooleanValue"/>
            <xsd:element name="objid" type="Values:ObjidValue"/>
            <xsd:element name="verdicttype" type="Values:VerdictValue"/>
            <xsd:element name="bitstring" type="Values:BitstringValue"/>
            <xsd:element name="hexstring" type="Values:HexstringValue"/>
            <xsd:element name="octetstring" type="Values:OctetstringValue"/>
            <xsd:element name="charstring" type="Values:CharstringValue"/>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue"/>
            <xsd:element name="record" type="Values:RecordValue"/>
            <xsd:element name="record_of" type="Values:RecordOfValue"/>
            <xsd:element name="set" type="Values:SetValue"/>
            <xsd:element name="set_of" type="Values:SetOfValue"/>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
            <xsd:element name="union" type="Values:UnionValue"/>
            <xsd:element name="anytype" type="Values:AnytypeValue"/>
            <xsd:element name="address" type="Values:AddressValue"/>
        </xsd:choice>
    </xsd:sequence>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>

<xsd:complexType name="RecordOfValue">
    <xsd:choice>
        <xsd:sequence>
            <xsd:element name="integer" type="Values:IntegerValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="float" type="Values:FloatValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="boolean" type="Values:BooleanValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="objid" type="Values:ObjidValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="bitstring" type="Values:BitstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="hexstring" type="Values:HexstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="octetstring" type="Values:OctetstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="charstring" type="Values:CharstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
```

```
        <xsd:sequence>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="record" type="Values:RecordValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="record_of" type="Values:RecordOfValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="set" type="Values:SetValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="set_of" type="Values:SetOfValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="union" type="Values:UnionValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="anytype" type="Values:AnytypeValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="address" type="Values:AddressValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:choice>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>

<xsd:complexType name="SetValue">
    <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="integer" type="Values:IntegerValue"/>
            <xsd:element name="float" type="Values:FloatValue"/>
            <xsd:element name="boolean" type="Values:BooleanValue"/>
            <xsd:element name="objid" type="Values:ObjidValue"/>
            <xsd:element name="verdicttype" type="Values:VerdictValue"/>
            <xsd:element name="bitstring" type="Values:BitstringValue"/>
            <xsd:element name="hexstring" type="Values:HexstringValue"/>
            <xsd:element name="octetstring" type="Values:OctetstringValue"/>
            <xsd:element name="charstring" type="Values:CharstringValue"/>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue"/>
            <xsd:element name="record" type="Values:RecordValue"/>
            <xsd:element name="record_of" type="Values:RecordOfValue"/>
            <xsd:element name="set" type="Values:SetValue"/>
            <xsd:element name="set_of" type="Values:SetOfValue"/>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
            <xsd:element name="union" type="Values:UnionValue"/>
            <xsd:element name="anytype" type="Values:AnytypeValue"/>
            <xsd:element name="address" type="Values:AddressValue"/>
        </xsd:choice>
    </xsd:sequence>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>

<xsd:complexType name="SetOfValue">
    <xsd:choice>
        <xsd:sequence>
            <xsd:element name="integer" type="Values:IntegerValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="float" type="Values:FloatValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="boolean" type="Values:BooleanValue" minOccurs="0"
```

```
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="objid" type="Values:ObjidValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="bitstring" type="Values:BitstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="hexstring" type="Values:HexstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="octetstring" type="Values:OctetstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="charstring" type="Values:CharstringValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="record" type="Values:RecordValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="record_of" type="Values:RecordOfValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="set" type="Values:SetValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="set_of" type="Values:SetOfValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="union" type="Values:UnionValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="anytype" type="Values:AnytypeValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:sequence>
            <xsd:element name="address" type="Values:AddressValue" minOccurs="0"
                maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:choice>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>

<xsd:complexType name="EnumeratedValue">
    <xsd:sequence>
        <xsd:element name="element" type="SimpleTypes:TString"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>

<xsd:complexType name="UnionValue">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="integer" type="Values:IntegerValue"/>
        <xsd:element name="float" type="Values:FloatValue"/>
        <xsd:element name="boolean" type="Values:BooleanValue"/>
        <xsd:element name="objid" type="Values:ObjidValue"/>
        <xsd:element name="verdicttype" type="Values:VerdictValue"/>
        <xsd:element name="bitstring" type="Values:BitstringValue"/>
        <xsd:element name="hexstring" type="Values:HexstringValue"/>
        <xsd:element name="octetstring" type="Values:OctetstringValue"/>
```

```
            <xsd:element name="charstring" type="Values:CharstringValue"/>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue"/>
            <xsd:element name="record" type="Values:RecordValue"/>
            <xsd:element name="record_of" type="Values:RecordOfValue"/>
            <xsd:element name="set" type="Values:SetValue"/>
            <xsd:element name="set_of" type="Values:SetOfValue"/>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
            <xsd:element name="union" type="Values:UnionValue"/>
            <xsd:element name="anytype" type="Values:AnytypeValue"/>
            <xsd:element name="address" type="Values:AddressValue"/>
        </xsd:choice>
        <xsd:attributeGroup ref="Values:ValueAtts"/>
    </xsd:complexType>

    <xsd:complexType name="AnytypeValue">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="integer" type="Values:IntegerValue"/>
            <xsd:element name="float" type="Values:FloatValue"/>
            <xsd:element name="boolean" type="Values:BooleanValue"/>
            <xsd:element name="objid" type="Values:ObjidValue"/>
            <xsd:element name="verdicttype" type="Values:VerdictValue"/>
            <xsd:element name="bitstring" type="Values:BitstringValue"/>
            <xsd:element name="hexstring" type="Values:HexstringValue"/>
            <xsd:element name="octetstring" type="Values:OctetstringValue"/>
            <xsd:element name="charstring" type="Values:OctetstringValue"/>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue"/>
            <xsd:element name="record" type="Values:RecordValue"/>
            <xsd:element name="record_of" type="Values:RecordOfValue"/>
            <xsd:element name="set" type="Values:SetValue"/>
            <xsd:element name="set_of" type="Values:SetOfValue"/>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
            <xsd:element name="union" type="Values:UnionValue"/>
            <xsd:element name="address" type="Values:AddressValue"/>
        </xsd:choice>
        <xsd:attributeGroup ref="Values:ValueAtts"/>
    </xsd:complexType>

    <xsd:complexType name="AddressValue">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="integer" type="Values:IntegerValue"/>
            <xsd:element name="float" type="Values:FloatValue"/>
            <xsd:element name="boolean" type="Values:BooleanValue"/>
            <xsd:element name="objid" type="Values:ObjidValue"/>
            <xsd:element name="verdicttype" type="Values:VerdictValue"/>
            <xsd:element name="bitstring" type="Values:BitstringValue"/>
            <xsd:element name="hexstring" type="Values:HexstringValue"/>
            <xsd:element name="octetstring" type="Values:OctetstringValue"/>
            <xsd:element name="charstring" type="Values:OctetstringValue"/>
            <xsd:element name="universal_charstring"
                type="Values:UniversalCharstringValue"/>
            <xsd:element name="record" type="Values:RecordValue"/>
            <xsd:element name="record_of" type="Values:RecordOfValue"/>
            <xsd:element name="set" type="Values:SetValue"/>
            <xsd:element name="set_of" type="Values:SetOfValue"/>
            <xsd:element name="enumerated" type="Values:EnumeratedValue"/>
            <xsd:element name="union" type="Values:UnionValue"/>
            <xsd:element name="anytype" type="Values:AnytypeValue"/>
        </xsd:choice>
        <xsd:attributeGroup ref="Values:ValueAtts"/>
    </xsd:complexType>
</xsd:schema>
```

4.B تخطيط TCI-TL XML لمقاسات

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates"
    xmlns:Templates="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates"
    xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
    xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
    xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
elementFormDefault="qualified">

    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values.xsd"
     schemaLocation="Values.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types.xsd"
schemaLocation="Types.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes.xsd"
     schemaLocation="SimpleTypes.xsd"/>
```

```
    <xsd:complexType name="TciValueTemplate">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Values:Value">
            <xsd:choice>
                <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                <xsd:element name="float" type="Templates:FloatTemplate"/>
                <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                <xsd:element name="universal_charstring"
type="Templates:UniversalCharstringTemplate"/>
            <xsd:element name="record" type="Templates:RecordTemplate"/>
            <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
            <xsd:element name="set" type="Templates:SetTemplate"/>
            <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
            <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
            <xsd:element name="union" type="Templates:UnionTemplate"/>
            <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
            <xsd:element name="address" type="Templates:AddressTemplate"/>
                    <xsd:element name="omit" type="Templates:omit"/>
                    <xsd:element name="any" type="Templates:any"/>
                    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
            </xsd:choice>
             </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="omit">
        <xsd:simpleContent>
            <xsd:extension base="SimpleTypes:TString">
                    <xsd:attributeGroup ref="Values:ValueAtts"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>

    <xsd:complexType name="any">
        <xsd:simpleContent>
            <xsd:extension base="SimpleTypes:TString">
                    <xsd:attributeGroup ref="Values:ValueAtts"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>

    <xsd:complexType name="anyoromit">
        <xsd:simpleContent>
            <xsd:extension base="SimpleTypes:TString">
                    <xsd:attributeGroup ref="Values:ValueAtts"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>


    <xsd:complexType name="TciNonValueTemplate">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="any" type="Templates:any"/>
                <xsd:element name="all" type="Templates:all"/>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="all">
        <xsd:simpleContent>
            <xsd:extension base="SimpleTypes:TString">
                    <xsd:attributeGroup ref="Values:ValueAtts"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>


    <xsd:complexType name="TciValueDifference">
            <xsd:attribute name="desc" type="SimpleTypes:TString" use="optional"/>
            <xsd:attribute name="val" type="SimpleTypes:xpath" use="required"/>
            <xsd:attribute name="tmpl" type="SimpleTypes:xpath" use="required"/>
```

```
    </xsd:complexType>

    <xsd:complexType name="TciValueDifferenceList">
            <xsd:sequence>
                    <xsd:element name="diff" type="Templates:TciValueDifference" minOccurs="1"
maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>


    <xsd:complexType name="IntegerTemplate">
        <xsd:complexContent>
            <xsd:extension base="Values:IntegerValue">
            <xsd:choice>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                        <xsd:element name="omit" type="Templates:omit"/>
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="null" type="xsd:string"/>
            </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="FloatTemplate">
        <xsd:complexContent>
            <xsd:extension base="Values:FloatValue">
            <xsd:choice>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                        <xsd:element name="omit" type="Templates:omit"/>
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="null" type="xsd:string"/>
            </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="BooleanTemplate">
        <xsd:complexContent>
            <xsd:extension base="Values:BooleanValue">
            <xsd:choice>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                        <xsd:element name="omit" type="Templates:omit"/>
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="null" type="xsd:string"/>
            </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="ObjidTemplate">
        <xsd:complexContent>
            <xsd:extension base="Values:ObjidValue">
            <xsd:choice>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                        <xsd:element name="omit" type="Templates:omit"/>
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="null" type="xsd:string"/>
            </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="BitstringTemplate">
        <xsd:complexContent>
            <xsd:extension base="Values:BitstringValue">
            <xsd:choice>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                        <xsd:element name="omit" type="Templates:omit"/>
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="null" type="xsd:string"/>
            </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
```

```
<xsd:complexType name="HexstringTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:BitstringValue">
        <xsd:choice>
            <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                    <xsd:element name="omit" type="Templates:omit"/>
                    <xsd:element name="any" type="Templates:any"/>
                    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
            <xsd:element name="null" type="xsd:string"/>
        </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="OctetstringTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:OctetstringValue">
        <xsd:choice>
            <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                    <xsd:element name="omit" type="Templates:omit"/>
                    <xsd:element name="any" type="Templates:any"/>
                    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
            <xsd:element name="null" type="xsd:string"/>
        </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CharstringTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:CharstringValue">
        <xsd:choice>
            <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                    <xsd:element name="omit" type="Templates:omit"/>
                    <xsd:element name="any" type="Templates:any"/>
                    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
            <xsd:element name="null" type="xsd:string"/>
        </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="UniversalCharstringTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:UniversalCharstringValue">
        <xsd:choice>
            <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                    <xsd:element name="omit" type="Templates:omit"/>
                    <xsd:element name="any" type="Templates:any"/>
                    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
            <xsd:element name="null" type="xsd:string"/>
        </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="RecordTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:RecordValue">
            <xsd:sequence>
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                    <xsd:element name="float" type="Templates:FloatTemplate"/>
                    <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                    <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                    <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                    <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                    <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                    <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                    <xsd:element name="universal_charstring"
                        type="Templates:UniversalCharstringTemplate"/>
                <xsd:element name="record" type="Templates:RecordTemplate"/>
                <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
                <xsd:element name="set" type="Templates:SetTemplate"/>
                <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
                <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
                <xsd:element name="union" type="Templates:UnionTemplate"/>
                <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
                <xsd:element name="address" type="Templates:AddressTemplate"/>
                        <xsd:element name="omit" type="Templates:omit"/>
```

```
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                    <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                    </xsd:choice>
                </xsd:sequence>
    </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="RecordOfTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:RecordOfValue">
            <xsd:choice>
                <xsd:sequence>
                    <xsd:element name="integer" type="Templates:IntegerTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="float" type="Templates:FloatTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="boolean" type="Templates:BooleanTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="objid" type="Templates:ObjidTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="bitstring" type="Templates:BitstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="hexstring" type="Templates:HexstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="octetstring" type="Templates:OctetstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="charstring" type="Templates:CharstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="universal_charstring"
                        type="Templates:UniversalCharstringTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="record" type="Templates:RecordTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="record_of" type="Templates:RecordOfTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="set" type="Templates:SetTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="set_of" type="Templates:SetOfTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="union" type="Templates:UnionTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="anytype" type="Templates:AnytypeTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="address" type="Templates:AddressTemplate" minOccurs="0"
```

```
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                    <xsd:element name="omit" type="Templates:omit"/>
                    <xsd:element name="any" type="Templates:any"/>
                    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
            <xsd:element name="templateDef" type="SimpleTypes:TString"/>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SetTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:SetValue">
            <xsd:sequence>
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                    <xsd:element name="float" type="Templates:FloatTemplate"/>
                    <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                    <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                    <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                    <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                    <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                    <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                    <xsd:element name="universal_charstring"
                        type="Templates:UniversalCharstringTemplate"/>
                <xsd:element name="record" type="Templates:RecordTemplate"/>
                <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
                <xsd:element name="set" type="Templates:SetTemplate"/>
                <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
                <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
                <xsd:element name="union" type="Templates:UnionTemplate"/>
                <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
                <xsd:element name="address" type="Templates:AddressTemplate"/>
                        <xsd:element name="omit" type="Templates:omit"/>
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                </xsd:choice>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SetOfTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:SetOfValue">
            <xsd:choice>
                <xsd:sequence>
                    <xsd:element name="integer" type="Templates:IntegerTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="float" type="Templates:FloatTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="boolean" type="Templates:BooleanTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="objid" type="Templates:ObjidTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="bitstring" type="Templates:BitstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="hexstring" type="Templates:HexstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="octetstring" type="Templates:OctetstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="charstring" type="Templates:CharstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
```

```
                <xsd:sequence>
                    <xsd:element name="universal_charstring"
                        type="Templates:UniversalCharstringTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="record" type="Templates:RecordTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="record_of" type="Templates:RecordOfTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="set" type="Templates:SetTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="set_of" type="Templates:SetOfTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="union" type="Templates:UnionTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="anytype" type="Templates:AnytypeTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="address" type="Templates:AddressTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                    <xsd:element name="omit" type="Templates:omit"/>
                    <xsd:element name="any" type="Templates:any"/>
                    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
            <xsd:element name="templateDef" type="SimpleTypes:TString"/>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="EnumeratedTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:EnumeratedValue">
            <xsd:choice>
                    <xsd:element name="omit" type="Templates:omit"/>
                    <xsd:element name="any" type="Templates:any"/>
                    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
            <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="UnionTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:UnionValue">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                <xsd:element name="float" type="Templates:FloatTemplate"/>
                <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                <xsd:element name="universal_charstring"
                    type="Templates:UniversalCharstringTemplate"/>
            <xsd:element name="record" type="Templates:RecordTemplate"/>
            <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
            <xsd:element name="set" type="Templates:SetTemplate"/>
            <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
            <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
            <xsd:element name="union" type="Templates:UnionTemplate"/>
            <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
```

```
                <xsd:element name="address" type="Templates:AddressTemplate"/>
                        <xsd:element name="omit" type="Templates:omit"/>
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
            </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="AnytypeTemplate">
        <xsd:complexContent>
            <xsd:extension base="Values:AnytypeValue">
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                    <xsd:element name="float" type="Templates:FloatTemplate"/>
                    <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                    <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                    <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                    <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                    <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                    <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                    <xsd:element name="universal_charstring"
                        type="Templates:UniversalCharstringTemplate"/>
                <xsd:element name="record" type="Templates:RecordTemplate"/>
                <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
                <xsd:element name="set" type="Templates:SetTemplate"/>
                <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
                <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
                <xsd:element name="union" type="Templates:UnionTemplate"/>
                <xsd:element name="address" type="Templates:AddressTemplate"/>
                        <xsd:element name="omit" type="Templates:omit"/>
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="AddressTemplate">
        <xsd:complexContent>
            <xsd:extension base="Values:AnytypeValue">
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                    <xsd:element name="float" type="Templates:FloatTemplate"/>
                    <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                    <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                    <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                    <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                    <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                    <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                    <xsd:element name="universal_charstring"
                        type="Templates:UniversalCharstringTemplate"/>
                <xsd:element name="record" type="Templates:RecordTemplate"/>
                <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
                <xsd:element name="set" type="Templates:SetTemplate"/>
                <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
                <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
                <xsd:element name="union" type="Templates:UnionTemplate"/>
                <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
                        <xsd:element name="omit" type="Templates:omit"/>
                        <xsd:element name="any" type="Templates:any"/>
                        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                </xsd:choice>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
</xsd:schema>
```

5.B تخطيط TCI-TL XML لأحداث

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Events"
    xmlns:Events="http://uri.etsi.org/ttcn-3/3.0.0/tci/Events"
    xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
    xmlns:Templates="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates"
    xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
```

```
    xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values" elementFormDefault="qualified">

    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes.xsd"
     schemaLocation="SimpleTypes.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types.xsd"
schemaLocation="Types.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values.xsd"
     schemaLocation="Values.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates.xsd"
     schemaLocation="Templates.xsd"/>

    <!-- common definition for all events -->
    <xsd:complexType name="Event" mixed="true">
        <xsd:sequence>
            <xsd:element name="am" type="SimpleTypes:TString"/>
        </xsd:sequence>
        <xsd:attribute name="ts" type="xsd:time" use="required"/>
        <xsd:attribute name="src" type="SimpleTypes:TString" use="optional"/>
        <xsd:attribute name="line" type="SimpleTypes:TInteger" use="optional"/>

        <!-- general identifier structure for test components, ports and timer -->
        <xsd:attribute name="name" type="SimpleTypes:TString" use="required"/>
        <xsd:attribute name="id" type="SimpleTypes:TInteger" use="required"/>
        <xsd:attribute name="type" type="SimpleTypes:TString" use="required"/>
    </xsd:complexType>

    <!-- this event is extended by all port configuration events -->
    <xsd:complexType name="PortConfiguration">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port1" type="Types:TriPortIdType" minOccurs="1"
maxOccurs="1"/>
                    <xsd:element name="port2" type="Types:TriPortIdType" minOccurs="1"
maxOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- this event is extended by all port status events -->
    <xsd:complexType name="PortStatus">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- testcases -->
    <xsd:complexType name="tliTcExecute">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="tcId" type="Types:TciTestCaseIdType"/>
                    <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
                    <xsd:element name="dur" type="Types:TriTimerDurationType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliTcStart">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="tcId" type="Types:TciTestCaseIdType"/>
                    <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
                    <xsd:element name="dur" type="Types:TriTimerDurationType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliTcStop">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
```

```
    </xsd:complexType>

    <xsd:complexType name="tliTcStarted">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="tcId" type="Types:TciTestCaseIdType"/>
                    <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
                    <xsd:element name="dur" type="Types:TriTimerDurationType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliTcTerminated">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="tcId" type="Types:TciTestCaseIdType"/>
                    <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
                    <xsd:element name="outcome" type="Values:VerdictValue"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- control -->
    <xsd:complexType name="tliCtrlStart">
        <xsd:complexContent>
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliCtrlStop">
        <xsd:complexContent>
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliCtrlTerminated">
        <xsd:complexContent>
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- asynchronous communication -->
    <xsd:complexType name="tliMSend_m">
         <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="msgValue" type="Values:Value"/>
                    <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                        <xsd:sequence>
                            <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>
                            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                        </xsd:sequence>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMSend_m_BC">
         <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="msgValue" type="Values:Value"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                        <xsd:sequence>
                            <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>
                            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                        </xsd:sequence>
                    </xsd:choice>
```

```
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMSend_m_MC">
         <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="msgValue" type="Values:Value"/>
                    <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                        <xsd:sequence>
                            <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>
                            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                        </xsd:sequence>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMSend_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="msgValue" type="Values:Value"/>
                    <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMSend_c_BC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="msgValue" type="Values:Value"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMSend_c_MC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="msgValue" type="Values:Value"/>
                    <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMDetected_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="msgValue" type="Types:TriMessageType"/>
                    <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMDetected_c">
        <xsd:complexContent mixed="true">
```

```
            <xsd:extension base="Events:Event">
                    <xsd:sequence>
                        <xsd:element name="port" type="Types:TriPortIdType"/>
                        <xsd:element name="msgValue" type="Types:TriMessageType"/>
                        <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
                    </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMMismatch_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                    <xsd:sequence>
                        <xsd:element name="port" type="Types:TriPortIdType"/>
                        <xsd:element name="msgValue" type="Values:Value"/>
                        <xsd:element name="msgTmpl" type="Templates:TciValueTemplate"/>
                        <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
                        <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                        <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"
minOccurs="0"/>
                    </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMMismatch_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                    <xsd:sequence>
                        <xsd:element name="port" type="Types:TriPortIdType"/>
                        <xsd:element name="msgValue" type="Values:Value"/>
                        <xsd:element name="msgTmpl" type="Templates:TciValueTemplate"/>
                        <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
                        <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
                        <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"
minOccurs="0"/>
                    </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMReceive_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                    <xsd:sequence>
                        <xsd:element name="port" type="Types:TriPortIdType"/>
                        <xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>
                        <xsd:element name="msgTmpl" type="Templates:TciValueTemplate"
minOccurs="0"/>
                        <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                        <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"
minOccurs="0"/>
                    </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliMReceive_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                    <xsd:sequence>
                        <xsd:element name="port" type="Types:TriPortIdType"/>
                        <xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>
                        <xsd:element name="msgTmpl" type="Templates:TciValueTemplate"
minOccurs="0"/>
                        <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
                        <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"
minOccurs="0"/>
                    </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

     <!-- synchronous communication -->
    <xsd:complexType name="tliPrCall_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
```

```
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType" minOccurs="0"/>
                    <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"
minOccurs="0"/>
                        <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrCall_m_BC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType" minOccurs="0"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"
minOccurs="0"/>
                        <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrCall_m_MC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType" minOccurs="0"/>
                    <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"
minOccurs="0"/>
                        <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrCall_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType" minOccurs="0"/>
                    <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrCall_c_BC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType" minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
```

```
    <xsd:complexType name="tliPrCall_c_MC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType" minOccurs="0"/>
                    <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetcallDetected_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="pars" type="Types:TriParameterListType"/>
                    <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetcallDetected_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="pars" type="Types:TciParameterListType"/>
                    <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetcallMismatch_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                    <xsd:element name="pars" type="Types:TriParameterListType"/>
                    <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetcallMismatch_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                    <xsd:element name="pars" type="Types:TciParameterListType"/>
                    <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetcall_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="from" type="Types:TriAddressType" minOccurs="0"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                    <xsd:element name="pars" type="Types:TriParameterListType"/>
```

```
                    <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetcall_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="from" type="Types:TriAddressType" minOccurs="0"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                    <xsd:element name="pars" type="Types:TciParameterListType"/>
                    <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrReply_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                        <xsd:sequence>
                            <xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>
                            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                        </xsd:sequence>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrReply_m_BC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                        <xsd:sequence>
                            <xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>
                            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                        </xsd:sequence>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrReply_m_MC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                        <xsd:sequence>
                            <xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>
                            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                        </xsd:sequence>
```

```
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrReply_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrReply_c_BC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrReply_c_MC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetReplyDetected_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="address" type="Types:TriAddressType"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetReplyDetected_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="from" type="Types:TriComponentIdType"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetReplyMismatch_m">
```

```
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="replTmpl" type="Values:Value"/>
                    <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
                    <xsd:element name="address" type="Types:TriAddressType"/>
                    <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetReplyMismatch_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="replTmpl" type="Values:Value"/>
                    <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
                    <xsd:element name="from" type="Types:TriComponentIdType"/>
                    <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetReply_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="replTmpl" type="Values:Value"/>
                    <xsd:element name="address" type="Types:TriAddressType"/>
                    <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrGetReply_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="replValue" type="Values:Value"/>
                    <xsd:element name="replTmpl" type="Values:Value"/>
                    <xsd:element name="from" type="Types:TriComponentIdType"/>
                    <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrRaise_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                    <xsd:element name="excValue" type="Values:Value"/>
                    <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                        <xsd:sequence>
                            <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>
                            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
```

```
                        </xsd:sequence>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrRaise_m_BC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                    <xsd:element name="excValue" type="Values:Value"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                        <xsd:sequence>
                            <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>
                            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                        </xsd:sequence>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrRaise_m_MC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                    <xsd:element name="excValue" type="Values:Value"/>
                    <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>
                    <xsd:choice>
                        <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                        <xsd:sequence>
                            <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>
                            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                        </xsd:sequence>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrRaise_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="excValue" type="Values:Value"/>
                    <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrRaise_c_BC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="excValue" type="Values:Value"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
```

```
    <xsd:complexType name="tliPrRaise_c_MC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="excValue" type="Values:Value"/>
                    <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrCatchDetected_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="exc" type="Types:TriExceptionType"/>
                    <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrCatchDetected_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="exc" type="Types:TriExceptionType"/>
                    <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrCatchMismatch_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="exc" type="Values:Value"/>
                    <xsd:element name="excTmpl" type="Templates:TciValueTemplate"/>
                    <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
                    <xsd:element name="address" type="Types:TriAddressType"/>
                    <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrCatchMismatch_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                    <xsd:element name="exc" type="Values:Value"/>
                    <xsd:element name="excTmpl" type="Templates:TciValueTemplate"/>
                    <xsd:element name="address" type="Types:TriAddressType"/>
                    <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliPrCatch_m">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
```

```
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                <xsd:element name="exception" type="Values:Value"/>
                <xsd:element name="exceptionTmpl" type="Templates:TciValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatch_c">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                <xsd:element name="exception" type="Values:Value"/>
                <xsd:element name="exceptionTmpl" type="Templates:TciValueTemplate"/>
                <xsd:element name="from" type="Types:TriComponentIdType"/>
                <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatchTimeoutDetected">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatchTimeout">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="parsValue" type="Types:TriParameterListType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<!-- components -->
<xsd:complexType name="tliCCreate">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="name" type="SimpleTypes:TString"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCStart">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="name" type="Types:TciBehaviourIdType"/>
                <xsd:element name="pars" type="Types:TciParameterListType" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCRunning">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="status" type="SimpleTypes:TBoolean"/>
            </xsd:sequence>
```

```
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>


<xsd:complexType name="tliCAlive">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="status" type="SimpleTypes:TBoolean"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCStop">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCKill">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCDoneMismatch">
   <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
            <xsd:attribute name="done" type="SimpleTypes:TBoolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCKilledMismatch">
   <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
            <xsd:attribute name="done" type="SimpleTypes:TBoolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCDone">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
            <xsd:attribute name="done" type="SimpleTypes:TBoolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

 <xsd:complexType name="tliCKilled">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="compTmpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
            <xsd:attribute name="done" type="SimpleTypes:TBoolean"/>
```

```
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCTerminated">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="verdict" type="Values:VerdictValue" maxOccurs="1"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<!-- ports -->
<xsd:complexType name="tliPConnect">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPDisconnect">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPMap">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPUnmap">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPClear">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPStart">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPStop">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPHalt">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<!-- codec -->
<xsd:complexType name="tliEncode">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="val" type="Values:Value"/>
                <xsd:choice>
                    <xsd:element name="msg" type="Types:TriMessageType"/>
                    <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                </xsd:choice>
            </xsd:sequence>
            <xsd:attribute name="codec" type="SimpleTypes:TString" use="optional"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

```
    <xsd:complexType name="tliDecode" mixed="true">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:choice>
                        <xsd:element name="val" type="Values:Value"/>
                        <xsd:element name="decoder-failure" type="SimpleTypes:TciStatusType"/>
                    </xsd:choice>
                    <xsd:element name="msg" type="Types:TriMessageType"/>
                </xsd:sequence>
                <xsd:attribute name="codec" type="SimpleTypes:TString" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- timers -->
    <xsd:complexType name="tliTTimeoutDetected">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1"
minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliTTimeoutMismatch">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1"
minOccurs="1"/>
                    <xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" maxOccurs="1"
minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliTTimeout">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1"
minOccurs="1"/>
                    <xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" maxOccurs="1"
minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliTStart">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="timer" type="Types:TriTimerIdType"/>
                    <xsd:element name="dur" type="Types:TriTimerDurationType"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliTStop">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="timer" type="Types:TriTimerIdType"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliTRead">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="timer" type="Types:TriTimerIdType"/>
                    <xsd:element name="elapsed" type="Types:TriTimerDurationType"/>
```

```
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliTRunning">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="timer" type="Types:TriTimerIdType"/>
                </xsd:sequence>
                <xsd:attribute name="status" type="SimpleTypes:TBoolean"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- scope -->
    <xsd:complexType name="tliSEnter">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                    <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
                    <xsd:element name="kind" type="SimpleTypes:TString"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliSLeave">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                    <xsd:element name="return" type="Values:Value" minOccurs="0"/>
                    <xsd:element name="kind" type="SimpleTypes:TString"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- variables and module parameter -->
    <xsd:complexType name="tliVar">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                    <xsd:element name="val" type="Values:Value" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliModuleParr">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                    <xsd:element name="val" type="Values:Value" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- verdicts -->
    <xsd:complexType name="tliGetVerdict">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="verdict" type="Values:VerdictValue"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliSetVerdict">
```

```
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="verdict" type="Values:VerdictValue"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- log -->
    <xsd:complexType name="tliLog">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="log" type="SimpleTypes:TString"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <!-- alt -->
    <xsd:complexType name="tliAEnter">
        <xsd:complexContent>
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliALeave">
        <xsd:complexContent>
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliADefaults">
        <xsd:complexContent>
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliAActivate">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                    <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
                    <xsd:element name="ref" type="Values:Value"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliADeactivate">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="ref" type="Values:Value"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliANomatch">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliARepeat">
        <xsd:complexContent>
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tliAWait">
        <xsd:complexContent>
            <xsd:extension base="Events:Event"/>
        </xsd:complexContent>
    </xsd:complexType>

</xsd:schema>
```

6.B تخطيط TCI-TL XML لسجل

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/TLI"
    xmlns:TLI="http://uri.etsi.org/ttcn-3/3.0.0/tci/TLI"
    xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
    xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
    xmlns:Events="http://uri.etsi.org/ttcn-3/3.0.0/tci/Events" elementFormDefault="qualified">

    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types.xsd"
schemaLocation="Types.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values.xsd"
     schemaLocation="Values.xsd"/>
    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Events.xsd"
     schemaLocation="Events.xsd"/>

    <xsd:element name="logfile" type="TLI:LogModule"/>
    <xsd:complexType name="LogModule">
        <xsd:sequence>
            <xsd:element name="header" type="TLI:Header"/>
            <xsd:element name="body" type="TLI:Body"/>
            <xsd:element name="trailer" type="TLI:Trailer"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Header">
        <xsd:sequence>
           <!-- logging version -->
           <xsd:element name="version" type="xsd:string"/>
          <!-- begin of the log -->
            <xsd:element name="ts" type="xsd:time"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Trailer">
        <xsd:sequence/>
    </xsd:complexType>

    <xsd:complexType name="Body">
        <xsd:choice maxOccurs="unbounded">

            <!-- test cases operations -->
            <xsd:element name="tliTcExecute" type="Events:tliTcExecute"/>
            <xsd:element name="tliTcStart" type="Events:tliTcStart"/>
            <xsd:element name="tliTcStop" type="Events:tliTcStop"/>
            <xsd:element name="tliTcStarted" type="Events:tliTcStarted"/>
            <xsd:element name="tliTcTerminated" type="Events:tliTcTerminated"/>

            <!-- control operations -->
            <xsd:element name="tliCtrlStart" type="Events:tliCtrlStart"/>
            <xsd:element name="tliCtrlStop" type="Events:tliCtrlStop"/>
            <xsd:element name="tliCtrlTerminated" type="Events:tliCtrlTerminated"/>

            <!-- asynchronous communication -->
            <xsd:element name="tliMSend_m" type="Events:tliMSend_m"/>
            <xsd:element name="tliMSend_c" type="Events:tliMSend_c"/>
            <xsd:element name="tliMDetected_m" type="Events:tliMDetected_m"/>
            <xsd:element name="tliMDetected_c" type="Events:tliMDetected_c"/>
            <xsd:element name="tliMMismatch_m" type="Events:tliMMismatch_m"/>
            <xsd:element name="tliMMismatch_c" type="Events:tliMMismatch_c"/>
            <xsd:element name="tliMReceive_m" type="Events:tliMReceive_m"/>
            <xsd:element name="tliMReceive_c" type="Events:tliMReceive_c"/>

            <!-- synchronous communication -->
            <xsd:element name="tliPrCall_m" type="Events:tliPrCall_m"/>
            <xsd:element name="tliPrCall_c" type="Events:tliPrCall_c"/>

            <xsd:element name="tliPrGetcallDetected_m" type="Events:tliPrGetcallDetected_m"/>
            <xsd:element name="tliPrGetcallDetected_c" type="Events:tliPrGetcallDetected_c"/>
            <xsd:element name="tliPrGetcallMismatch_m" type="Events:tliPrGetcallMismatch_m"/>
            <xsd:element name="tliPrGetcallMismatch_c" type="Events:tliPrGetcallMismatch_c"/>
            <xsd:element name="tliPrGetcall_m"     type="Events:tliPrGetcall_m"/>
            <xsd:element name="tliPrGetcall_c"     type="Events:tliPrGetcall_c"/>

            <xsd:element name="tliPrReply_m"    type="Events:tliPrReply_m"/>
            <xsd:element name="tliPrReply_c"    type="Events:tliPrReply_c"/>

            <xsd:element name="tliPrGetReplyDetected_m" type="Events:tliPrGetReplyDetected_m"/>
            <xsd:element name="tliPrGetReplyDetected_c" type="Events:tliPrGetReplyDetected_c"/>
            <xsd:element name="tliPrGetReplyMismatch_m" type="Events:tliPrGetReplyMismatch_m"/>
```

```
            <xsd:element name="tliPrGetReplyMismatch_c" type="Events:tliPrGetReplyMismatch_c"/>
            <xsd:element name="tliPrGetReply_m"     type="Events:tliPrGetReply_m"/>
            <xsd:element name="tliPrGetReply_c"     type="Events:tliPrGetReply_c"/>

            <xsd:element name="tliPrRaise_m"    type="Events:tliPrRaise_m"/>
            <xsd:element name="tliPrRaise_c"    type="Events:tliPrRaise_c"/>

            <xsd:element name="tliPrCatchDetected_m"    type="Events:tliPrCatchDetected_m"/>
            <xsd:element name="tliPrCatchDetected_c"    type="Events:tliPrCatchDetected_c"/>
            <xsd:element name="tliPrCatchMismatch_m"    type="Events:tliPrCatchMismatch_m"/>
            <xsd:element name="tliPrCatchMismatch_c"    type="Events:tliPrCatchMismatch_c"/>
            <xsd:element name="tliPrCatch_m"    type="Events:tliPrCatch_m"/>
            <xsd:element name="tliPrCatch_c"    type="Events:tliPrCatch_c"/>

            <xsd:element name="tliPrCatchTimeout"    type="Events:tliPrCatchTimeout"/>

            <!-- components -->
            <xsd:element name="tliCCreate" type="Events:tliCCreate"/>
            <xsd:element name="tliCStart" type="Events:tliCStart"/>
            <xsd:element name="tliCRunning" type="Events:tliCRunning"/>
            <xsd:element name="tliCAlive" type="Events:tliCRunning"/>
            <xsd:element name="tliCStop" type="Events:tliCStop"/>
            <xsd:element name="tliCKill" type="Events:tliCStop"/>
            <xsd:element name="tliCDoneMismatch" type="Events:tliCDone"/>
            <xsd:element name="tliCDone" type="Events:tliCDone"/>
            <xsd:element name="tliCKilledMismatch" type="Events:tliCDone"/>
            <xsd:element name="tliCKilled" type="Events:tliCDone"/>
            <xsd:element name="tliCTerminated" type="Events:tliCTerminated"/>

            <!-- ports -->
            <xsd:element name="tliPConnect" type="Events:tliPConnect"/>
            <xsd:element name="tliPDisconnect" type="Events:tliPDisconnect"/>
            <xsd:element name="tliPMap" type="Events:tliPMap"/>
            <xsd:element name="tliPUnmap" type="Events:tliPUnmap"/>
            <xsd:element name="tliPClear" type="Events:tliPClear"/>
            <xsd:element name="tliPStart" type="Events:tliPStart"/>
            <xsd:element name="tliPStop" type="Events:tliPStop"/>
            <xsd:element name="tliPHalt" type="Events:tliPStop"/>

            <!-- codec -->
            <xsd:element name="tliDecode" type="Events:tliDecode"/>
            <xsd:element name="tliEncode" type="Events:tliEncode"/>

            <!-- timers -->
            <xsd:element name="tliTTimeoutDetected" type="Events:tliTTimeoutDetected"/>
            <xsd:element name="tliTTimeoutMimatch" type="Events:tliTTimeoutMismatch"/>
            <xsd:element name="tliTTimeout" type="Events:tliTTimeout"/>
            <xsd:element name="tliTStart" type="Events:tliTStart"/>
            <xsd:element name="tliTStop" type="Events:tliTStop"/>
            <xsd:element name="tliTRead" type="Events:tliTRead"/>
            <xsd:element name="tliTRunning" type="Events:tliTRunning"/>

            <!-- scopes -->
            <xsd:element name="tliSEnter" type="Events:tliSEnter"/>
            <xsd:element name="tliSLeave" type="Events:tliSLeave"/>

            <!-- statements -->
            <xsd:element name="tliVar" type="Events:tliVar"/>
            <xsd:element name="tliGetVerdict" type="Events:tliGetVerdict"/>
            <xsd:element name="tliSetVerdict" type="Events:tliSetVerdict"/>
            <xsd:element name="tliLog" type="Events:tliLog"/>

            <!-- alt  -->
            <xsd:element name="tliAEnter" type="Events:tliAEnter"/>
            <xsd:element name="tliALeave" type="Events:tliALeave"/>
            <xsd:element name="tliADefaults" type="Events:tliADefaults"/>
            <xsd:element name="tliAActivate" type="Events:tliAActivate"/>
            <xsd:element name="tliADeactivate" type="Events:tliADeactivate"/>
            <xsd:element name="tliANomatch" type="Events:tliANomatch"/>
            <xsd:element name="tliARepeat" type="Events:tliARepeat"/>
            <xsd:element name="tliAWait" type="Events:tliAWait"/>

        </xsd:choice>
    </xsd:complexType>
</xsd:schema>
```

بيبليوغرافيا

- INTOOL CGI/NPL038 (V2.2): *Generic Compiler/Interpreter interface; GCI Interface Specification Infrastructural Tools for Informational Technology and Telcomunications Conference Testing*, December 1996.
- ITU-T Recommendation X.292 (2002), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – The Tree and Tabular Combined Notation* (*TTCN*).

  ISO/IEC 9646-3:1998, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 3*: *The Tree and Tabular combined Notation* (*TTCN*).
- ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set* (*UCS*).
- OMG CORBA v2.2: *The Common Object Request Broker*: *Architecture and Specification*, Section 3, February 1998.

سلاسل التوصيات الصادرة عن قطاع تقييس الاتصالات

السلسلة A	تنظيم العمل في قطاع تقييس الاتصالات
السلسلة D	المبادئ العامة للتعريفة
السلسلة E	التشغيل العام للشبكة والخدمة الهاتفية وتشغيل الخدمات والعوامل البشرية
السلسلة F	خدمات الاتصالات غير الهاتفية
السلسلة G	أنظمة الإرسال ووسائطه والأنظمة والشبكات الرقمية
السلسلة H	الأنظمة السمعية المرئية والأنظمة متعددة الوسائط
السلسلة I	الشبكة الرقمية متكاملة الخدمات
السلسلة J	الشبكات الكبلية وإرسال إشارات تلفزيونية وبرامج صوتية وإشارات أخرى متعددة الوسائط
السلسلة K	الحماية من التداخلات
السلسلة L	إنشاء الكبلات وغيرها من عناصر المنشآت الخارجية وتركيبها وحمايتها
السلسلة M	إدارة الاتصالات بما في ذلك شبكة إدارة الاتصالات (TMN) وصيانة الشبكات
السلسلة N	الصيانة: الدارات الدولية لإرسال البرامج الإذاعية الصوتية والتلفزيونية
السلسلة O	مواصفات تجهيزات القياس
السلسلة P	نوعية الإرسال الهاتفي والمنشآت الهاتفية وشبكات الخطوط المحلية
السلسلة Q	التبديل والتشوير
السلسلة R	الإرسال البرقي
السلسلة S	التجهيزات المطرافية للخدمات البرقية
السلسلة T	المطاريف الخاصة بالخدمات التلماتية
السلسلة U	التبديل البرقي
السلسلة V	اتصالات المعطيات على الشبكة الهاتفية
السلسلة X	شبكات المعطيات والاتصالات بين الأنظمة المفتوحة ومسائل الأمن
السلسلة Y	البنية التحتية العالمية للمعلومات وملامح بروتوكول الإنترنت وشبكات الجيل التالي
السلسلة Z	**اللغات والجوانب العامة للبرمجيات في أنظمة الاتصالات**

طُبعت في سويسرا
جنيف، 2010