الاتحـــــــــاد الدولـــــــــي للاتصـــــــــالات

ITU-T Z.144

(2006/03)

قطـــاع تقييس الاتصـــالات
في الاتحاد الدولي للاتصالات

السلسلة Z: اللغات والجوانب العامة للبرمجيات في أنظمة الاتصالات

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

النسخة 3 من الاختبار وترميز ضبط الاختبار (TTCN-3): السطح بيني لوقت التسيير (TRI)

التوصيـــة ITU-T Z.144

الاتحاد الدولي للاتصالات

توصيات السلسلة Z الصادرة عن قطاع تقييس الاتصالات

اللغات والجوانب العامة للبرمجيات في أنظمة الاتصالات

تقنيات الوصف الشكلي (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
لغات البرمجة	
CHILL: لغة المستوى الرفيع لدى قطاع تقييس الاتصالات	Z.209–Z.200
لغة الإنسان–الآلة	
مبادئ عامة	Z.309–Z.300
قواعد النظم 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
الجودة	
جودة برمجيات الاتصالات	Z.409–Z.400
مظاهر الجودة للتوصيات المرتبطة بالبروتوكولات	Z.459–Z.450
الطرائق	
طرائق للتثبت من الصلاحية وللاختبار	Z.519–Z.500
البرمجيات الوسيطة	
بيئة المعالجة الموزعة	Z.609–Z.600

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

النسخة 3 من الاختبار وترميز ضبط الاختبار (TTCN): السطح بيني لوقت التسيير (TRI)

ملخص

تقدم هذه التوصية مواصفات السطح بيني لوقت التسيير لتنفيذ نظام اختبار للنسخة الثالثة من (*Testing and Test Control Notation 3 (الاختبار وترميز ضبط الاختبار)*). ويوفر السطح بيني لوقت التسيير التكييف الموصى به للتوقيت والاتصالات المتعلقة بنظام الاختبار لمنتدى تجهيز معين والنظام موضع الاختبار على التوالي. وتعرف هذه التوصية السطح بيني بأنه مجموعة من العمليات المستقلة عن اللغة المستهدفة.

ويعرف السطح بيني بأنه يتواءم مع التوصية Z.140. وتستخدم هذه التوصية للغة تعريف السطح بيني CORBA لتمديد السطح بيني لوقت التسيير لترميز عملية الاختبار وضبط الاختبار بصورة كاملة. ويحدد البندان 6 و7 التقابلات اللغوية للمواصفات المجردة مع اللغتين المستهدفين Java وANSI-C. ويتضمن المرفق ألف موجزاً لمواصفات السطح بيني المعتمد على لغة تعريف السطح بيني

المصدر

وافقت لجنة الدراسات 17 (2005-2008) لقطاع تقييس الاتصالات على التوصية ITU-T Z.144 بتاريخ 16 مارس 2006 بموجب الإجراء المحدد في التوصية ITU-T A.8.

تمهيـــد

الاتحاد الدولي للاتصالات وكالة متخصصة للأمم المتحدة في ميدان الاتصالات. 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 2006

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
1.1 التطابق 1
2 المراجع 1
3 التعاريف والمختصرات 2
1.3 التعاريف 2
2.3 المختصرات 3
4 البنية العامة لنظام اختبار TTCN-3 3
1.4 الكيانات في نظام اختبار TTCN-3 4
2.4 السطوح البينية في نظام اختبار TTCN-3 7
3.4 متطلبات تنفيذ نظام اختبار TTCN-3 7
5 السطح بيني لوقت تسيير TTCN-3 وعملياته 8
1.5 عرض عام للسطح بيني لوقت تسيير TTCN-3 (TRI) 8
2.5 مناولة الأخطاء 10
3.5 السطح بيني للبيانات 10
4.5 أوصاف العمليات 12
5.5 عمليات السطح بيني للاتصالات 12
6.5 عمليات السطح بيني للمنتدى 25
6 تقابل لغة جافا 28
1.6 مقدمة 28
2.6 الأسماء والنطاقات 28
3.6 تقابل النمط 28
4.6 الثوابت 36
5.6 تقابل السطوح البينية 36
6.6 معلمات اختيارية 39
7.6 تدميث السطح بيني لوقت تسيير TTCN-3 39
8.6 مناولة الأخطاء 40
7 تقابل لغة ANSI-C 40
1.7 مقدمة 40
2.7 الأسماء والنطاقات 40
3.7 إدارة الذاكرة 45
4.7 مناولة الأخطاء 45
8 سيناريوهات الاستخدام 45
1.8 السيناريو الأول 46
2.8 السيناريو الثاني 49
3.8 السيناريو الثالث 51
الملحق ألف (معياري) – موجز لغة تعريف السطوح البينية (IDL) 53
البيبليوغرافيا 56

المقدمـــة

تتألف هذه التوصية من جزأين مختلفين، حيث يصف الجزء الأول بنية تنفيذ نظام الاختبار في النسخة 3 من الاختبار وترميز ضبط الاختبارات (TTCN-3)، ويقدم الجزء الثاني مواصفات السطح بيني لوقت تسيير TTCN-3.

ويقدم الجزء الأول تقسيم نظام الاختبار TTCN-3 إلى أربعة كيانات رئيسية هي:

- إدارة الاختبار (TM)؛
- مجموعة الاختبارات القابلة للتنفيذ في TTCN-3 (TE)؛
- مكيف النظام قيد الاختبار (SA)؛
- مكيف المنتدى.

وعلاوة على ذلك، يجرى تعريف التفاعل بين هذه الكيانات مثل السطوح البينية المقابلة.

ويحدد الجزء الثاني من التوصية السطح بيني لوقت تسيير TTCN-3. ويعرف السطح بيني على أساس العمليات التي تنفذ كجزء من أحد الكيانات وتستدعى من جانب الكيانات الأخرى في نظام الاختبار. وفي كل عملية، تتولى مواصفات السطح بيني تعريف بنيات البيانات ذات الصلة، والتأثير المتوخى على نظام الاختبار وأية قيود على استخدام العملية. ويلاحظ أن مواصفات السطح بيني لا تعّرف سوى التفاعلات بين السطح بيني لنظام الاختبار TSI ومكيف النظام قيد الاختبار فضلاً عن عمليات المؤقت.

النسخة 3 من الاختبار وترميز ضبط الاختبار (TTCN): السطح بيني لوقت التسيير (TRI)

1 مجال التطبيق

تقدم هذه التوصية مواصفات السطح بيني لوقت تسيير عمليات تنفيذ نظام الاختبار في النسخة 3 من ترميز عملية الاختبار وضبط الاختبارات TTCN-3. ويوفر السطح بيني لوقت تسيير TTCN-3 مواءمة مقيسة للتوقيت والاتصالات ذات الصلة بنظام الاختبار لمنتدى تجهيز معين، والنظام قيد الاختبار على التوالي. وتعرف هذه التوصية السطح بيني بأنه مجموعة من العمليات المستقلة عن اللغة المستهدفة.

ويعرف السطح بيني بأنه يتطابق مع المعيار TTCN-3 (انظر أدناه). وتستخدم هذه التوصية للغة تعريف السطح بيني CORBA لتمديد السطح بي لوقت التسيير لترميز عملية الاختبار وضبط الاختبار بصورة كاملة. ويحدد البندان 6 و7 التقابلات اللغوية للمواصفات المجردة مع اللغتين المستهدفين Java وANSI-C . ويتضمن المرفق ألف موجزا لمواصفات السطح بيني المعتمد على لغة تعريف السطح بيني.

1.1 التطابق

يهدف الشرط المتعلق بأن يتطابق نظام اختبار TTCN-3 مع سطح بيني لوقت التسيير TRI إلى الالتزام بمواصفات السطح بيني بين الواردة في هذه التوصية فضلاً عن أحد تقابلات اللغة المستهدفة المدرجة.

مثال: إذا كان المورد يساند Java، فإن عملية السطح بيني لوقت التسيير تتطلب التطابق مع لغة تعريف السطح بيني لتقابل Java المحدد في هذه التوصية. ويتعين على عمليات التنفيذ، التي هي جزء من قابلية تنفيذ TTCN-3 فعل ذلك.

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، نحث جميع المستعملين لهذه التوصية على السعي إلى تطبيق أحدث طبعة للتوصيات 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 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*.

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

[3] 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)*.

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

3 التعاريف والمختصرات

1.3 التعاريف

لأغراض هذه التوصية، تسري المصطلحات والتعاريف الواردة في التوصية [2] Z.140 وتلك الواردة فيما يلي:

1.1.3 مجموعة اختبارات مجردة (ATS): انظر التوصية [1] X.290.

2.1.3 منفذ الاتصالات: آلية مجردة لتسيير الاتصالات بين مكونات الاختبار.

ملاحظة: يجرى تشكيل منفذ الاتصالات في شكل صف FIFO في اتجاه المستقبل. وقد تكون المنافذ معتمدة على الرسائل أو الإجراءات أو على خليط من الاثنين.

3.1.3 متوالية الاختبارات القابلة للتنفيذ (ETS): (انظر التوصية [1] X.290).

4.1.3 مؤقت صريح: المؤقت المعلن في مجموعة اختبار مجردة من TTCN-3 والتي يمكن النفاذ إليها عن طريق عمليات مؤقت TTCN-3.

5.1.3 تنفيذ المعلومات الإضافية لعلمية الاختبار (IXIT): انظر التوصية [1] X.290.

6.1.3 مؤقت ضمني: مؤقت نظام تستحدثه نسخة [1] X.290 القابلة للتنفيذ لحراسة نداء [1] X.290 أو عملية تنفيذ.

ملاحظة – المؤقتات الضمنية لا يمكن أن ينفذ إليه مستعمل TTCN-3.

7.1.3 مكيف المنتدى (PA): كيان يكيف TTCN-3 القابلة للتنفيذ لتوافق منتدى تنفيذ معين.

ملاحظة: يستحدث مكيف المنتدى رمزاً واحداً من الوقت لنظام اختبار TTCN-3، وينفذ وظائف خارجية فضلاً عن المؤقتات الصريحة والضمنية.

8.1.3 مكيف النظام قيد الاختبار (SA): كيان يكيف عمليات اتصالات TTCN-3 مع النظام قيد الاختبار المعتمد على السطح بيني لنظام الاختبار المجرد وينفذ السطح بيني لنظام الاختبار الحقيقي.

9.1.3 النظام قيد الاختبار (SUT): انظر التوصية [1] X.290.

ملاحظة: جميع الأنماط معروفة في وقت التصريف أي أنها مقيدة بصورة ثابتة.

10.1.3 حالة اختبار: انظر التوصية [1] X.290.

11.1.3 حدث اختبار: ترسل أو تستقبل بيانات اختبار (رسالة أو استدعاء إجراء) على منفذ الاتصالات الذي هو جزء من السطح بيني لنظام الاختبار.

12.1.3 إدارة الاختبارات (TM): كيان يوفر سطح بيني المستعمل ويدير نظام اختبار TTCN-3.

13.1.3 نظام الاختبار: انظر التوصية [1] X.290.

14.1.3 السطح بيني لنظام الاختبار: مكون اختبار يوفر تقابلا للمنافذ المتاحة في نظام اختبار TTCN-3 (المجرد) مع تلك المقدمة من نظام الاختبار الحقيقي.

15.1.3 تعريف المؤقت (TID): تعريف فريد لحالات المؤقت الصريح أو الضمني التي تستحدثها TTCN-3 القابلة للتنفيذ.

16.1.3 السطح بيني للتحكم في TTCN-3 (TCI): سطح بيني خاص في الوقت الحاضر يحدد التفاعل بين إدارة اختبار وTTCN-3 القابلة للتنفيذ نظام اختبار.

17.1.3 الجوانب القابلة للتنفيذ في TTCN-3 (TE): جزء من نظام الاختبار يتناول تفسير تنفيذ متوالية الاختبار القابلة للتنفيذ في TTCN-3.

18.1.3 السطح بيني لوقت تسيير TTCN-3 (TRI): سطح بيني يعرف تفاعل TTCN-3 القابلة للاختبار مع النظام قيد الاختبار ومكيف المنتدى في نظام الاختبار.

2.3 المختصرات

تستخدم هذه التوصية المختصرات التالية:

ATS مجموعة الاختبارات المجردة *(Abstract Test Suite)*

CH مناول المكونات *(Component Handler)*

ECD الكودكات الخارجية *(External CoDecs)*

EDS نظام التشفير/وفك التشفير *(Encoding/Decoding System)*

ETS متوالية الاختبارات القابلة للتنفيذ *(Executable Test Suite)*

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

IXIT معلومات إضافية للتنفيذ من أجل الاختبار *(Implementation eXtra Information for Testing)*

MSC مخطط تتابع الرسائل *(Message Sequence Chart)*

MTC مكون الاختبار الرئيسي *(Main Test Component)*

OMG فريق إدارة الأهداف *(Object Management Group)*

PA مكيف المنتدى *(Platform Adapter)*

SA مكيف النظام قيد الاختبار *(SUT Adapter)*

SUT النظام قيد الاختبار *(System Under Test)*

T3RTS نظام وقت تسيير TTCN-3 *(TTCN-3 Runtime System)*

TC ضبط الاختبار *(Test Control)*

TCI السطح بيني للتحكم في TTCN-3 *(TTCN-3 Control Interface)*

TE الجوانب القابلة للتنفيذ في TTCN-3 *(TTCN-3 Executable)*

TID تعريف المؤقت *(Timer IDentification)*

TL تسجيل الاختبار *(Test Logging)*

TM إدارة الاختبار *(Test Management)*

TRI السطح بيني لوقت تسيير TTCN-3 *(TTCN-3 Runtime Interface)*

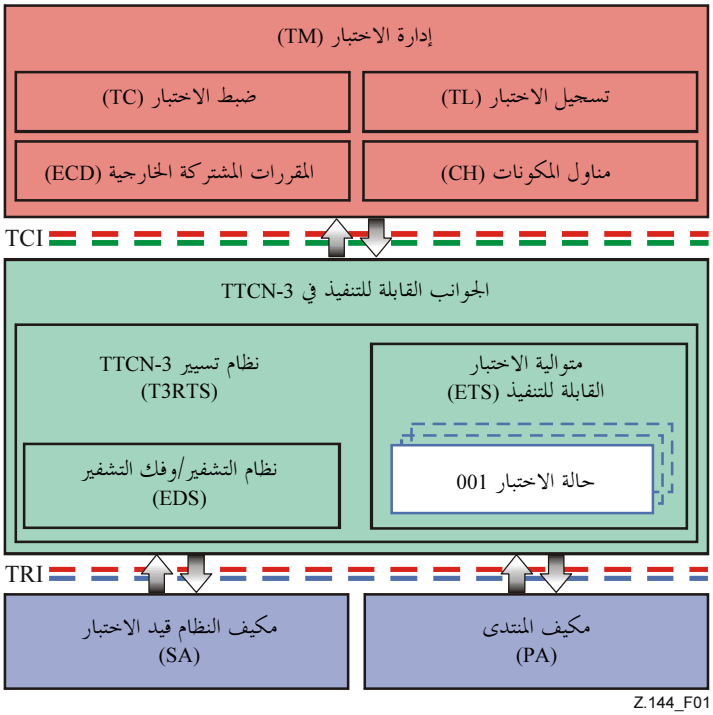
TSI السطح بيني لنظام الاختبار *(Test System Interface)*

TTCN ترميز الاختبار والتحكم في الاختبار *(Testing and Test Control Notation)*

TTCN-3 النسخة 3 من الترميز المختلط على شكل شجرة وجداول *(Tree and Tabular Combined Notation version 3)*

4 البنية العامة لنظام اختبار TTCN-3

يمكن النظر إلى نظام الاختبار في TTCN-3 من الناحية المفاهيمية على أنه مجموعة من الكيانات المتفاعلة حيث يتطابق كل كيان مع جانب خاص من الوظائفية في تنفيذ نظام الاختبار. وتدير هذه الكيانات عملية إنجاز الاختبار، وتفسير أو تنفيذ شفرة مجمعة من TTCN-3، وتحقق الاتصالات الصحيحة مع النظام قيد الاختبار وتنفذ الوظائف الخارجية ومناولة عمليات المؤقت (انظر الشكل 1).



الشكل Z.144/1 – البنية العامة لنظام اختبار TTCN-3

1.4 الكيانات في نظام اختبار TTCN-3

يبين الشكل 1 بنية تنفيذ نظام اختبار TTCN-3. وتجدر الملاحظة بأن زيادة تنقيح إدارة الاختبار لتصبح كيانات أصغر حجماً على النحو المبين في الشكل 1 والمستخدم في البنود التالية في هذه التوصية ليس سوى مساعدة في تحديد السطوح البينية لنظام اختبار TTCN-3.

ويشكل الجزء من نظام الاختبار الذي يتناول تفسير وإنجاز وحدات TTCN-3 أي متوالية الاختبارات القابلة للتنفيذ، جزءاً من متوالية الاختبارات القابلة للتنفيذ في TTCN-3 ويتواءم هذا إما مع الشفرة القابلة للتنفيذ التي أعدها معرف TTCN-3 أو مع مفسر TTCN-3 في تنفيذ نظام الاختبار. ومن المفترض أن تنفيذ نظام الاختبار يشمل مجموعة الاختبار القابلة للتنفيذ، على النحو المستخلص من متوالية الاختبارات المجردة في TTCN-3.

أما الجزء المتبقي من نظام اختبار TTCN-3 الذي يتناول أية جوانب لا يمكن إدراجها في المعلومات المقدمة في متوالية الاختبارات المجردة الأصلية وحدها، فيمكن تقسيمه وتوزيعه على كيانات إدارة الاختبار ومكيف النظام قيد الاختبار ومكيف المنتدى. وعموماً فإن هذه الكيانات تغطي السطح بيني لمستعمل نظام الاختبار والتحكم في إنجاز الاختبار وتسجيل حدث الاختبار فضلاً عن الاتصالات مع النظام قيد الاختبار وتنفيذ المؤقت.

1.1.4 إدارة الاختبار (TM)

يمكن أن نفرق، في الكيان الخاص بإدارة الاختبار TM بين الوظيفية المتعلقة بالتحكم في إنجاز الاختبار، وتسجيل حدث الاختبار.

1.1.1.4 ضبط الاختبار (TC)

يتولى كيان ضبط الاختبار (TC) المسؤولية عن الإدارة الشاملة لنظام الاختبار. فبعد بدء نظام الاختبار، يبدأ إنجاز الاختبار داخل كيان ضبط الاختبار. وهذا الكيان مسؤول عن التنفيذ السليم لوحدات TTCN-3 أي نشر معلمات الوحدات و/أو المعلومات عن تنفيذ المعلومات الإضافية للاختبار على الجوانب القابلة للتنفيذ في TTCN-3 إذا اقتضى الأمر. ويقوم هذا الكيان عادة بتنفيذ سطح بيني لمستعمل نظام الاختبار.

2.1.1.4 تسجيل الاختبار (TL)

كيان تسجيل الاختبار مسؤول عن المحافظة على سجل الاختبار. وتقوم صراحة الجوانب القابلة للتنفيذ في TTCN-3 بتسجيل أحداث الاختبار. ولهذا الكيان سطح بيني لا اتجاهي حيث يمكن لأي كيان يشكل جزءاً من الجوانب القابلة للتنفيذ في TTCN-3 أن يقدم طلباً للتسجيل إلى كيان تسجيل الاختبار. كما يمكن استخدام السطح بيني الداخلي لإدارة الاختبار في تسجيل معلومات إدارة الاختبار التي يستحدثها التحكم في الاختبار.

3.1.1.4 كيانات الكودك الخارجية

تتولى كيانات الكودك الخارجية، بصورة اختبارية، مسؤولية تشفير وفك شفرة البيانات المرتبطة الاتصالات المعتمدة على الرسائل أو المعتمدة على الإجراءات داخل الجوانب القابلة للتنفيذ في TTCN-3. ويمكن استخدام الكودكات الخارجية بالتوازي مع تلك المدمجة المرتبطة بالجوانب القابلة للتنفيذ في TTCN-3 أو كبديل لها. وعلى العكس من الكودكات المدمجة، تنطوي الكودكات الخارجية على سطوح بينية مقيسة مما يجعلها قابلة للحركة بين مختلف أنظمة وأدوات TTCN-3.

4.1.1.4 مناول المكونات (CH)

كيان مناول المكونات مسؤول عن توزيع مكونات الاختبار الموازية. ويمكن أن يكون هذا التوزيع عن طريق نظام واحد أو عدة أنظمة مادية. ويسمح كيان مناول المكونات لإدارة الاختبار باستحداث أنظمة اختبار موزعة والتحكم فيها بطريقة تتسم بالشفافية ومستقلة عن الجوانب القابلة للتنفيذ حتى TTCN-3.

2.1.4 الجوانب القابلة للتنفيذ في TTCN-3

هذا الكيان مسؤول عن تفسير أو إنجاز متوالية الاختبار المجرد في TTCN-3. ويمكن من الناحية المفاهيمية تقسيم الجوانب القابلة للتنفيذ في TTCN-3 إلى ثلاثة كيانات متفاعلة هي متوالية اختبار قابلة للتنفيذ ونظام وقت تسيير TTCN-3 وكيان نظام التشفير وفك الشفرة. ويلاحظ أن هذا التنقيح للجوانب القابلة للتنفيذ بتقسيمها إلى كيانات أصغر ليس سوى مساعدة مفاهيمية لتعريف السطوح البينية لنظام اختبار TTCN-3 ولا يوجد ما يدعو إلى إدراج هذا التفريق في عمليات تنفيذ السطح بيني لوقت تسيير TTCN-3.

وتعرف البنود التالية مسؤوليات كل كيان وتناقش كذلك طرق مناولة المؤقتات في السطح بيني لوقت تسيير TTCN-3.

1.2.1.4 متوالية الاختبارات القابلة للتنفيذ (ETS)

يتولى كيان متوالية الاختبارات القابلة للتنفيذ ETS إنجاز أو تفسير حالات الاختبار، وتتابع ومواءمة أحداث الاختبار على النحو المعرف في وحدات TTCN-3 المقابلة في التوصية Z.140 [2]. ويتفاعل هذا الكيان مع كيان نظام وقت التسيير T3RTS لإرسال ومحاولة استقبال وحذف مكونات اختبار TTCN-3 فضلاً عن مناولة بداءات الوظيفة الخارجية وعمليات الإجراءات والمؤقتات. ويلاحظ أن الكيان الخاص بمتوالية الاختبار القابل للتنفيذ ETS لا تتفاعل بصورة مباشرة مع مكيف النظام قيد الاختبار عن طريق السطح بيني لوقت تسيير TTCN-3.

2.2.1.4 نظام وقت تسيير النسخة 3 من الاختبار وترميز ضبط الاختبار (T3RTS)

يتفاعل هذا الكيان T3RTS مع كيانات إدارة الاختبار TM ومكيف النظام قيد الاختبار (SA)، ومكيف المنتدى (PA) عن طريق السطح بيني للتحكم في TTCN والسطح بيني لوقت التسيير (TRI)، ويدير كيان متوالية الاختبار القابلة للتنفيذ، ونظام لتشفير وفك التشفير. ويقوم هذا الكيان بتدميث المكيفات فضلاً عن كياني متوالية الاختبار القابلة للتنفيذ ونظام التشفير وفك التشفير. ويؤدي هذا الكيان جميع الإجراءات اللازمة لبدء إنجاز حالة اختبار أو العمل مع المعلمات الواردة في كيان متوالية الاختبار القابلة للتنفيذ. ويقوم بصف كيان إدارة الاختبار (TM) لتحديد قيم معلمات النماذج التي تحتاجها متوالية الاختبار القابلة للتنفيذ، ويرسل لها معلومات التسجيل. ويقوم كذلك بتجميع وتسوية الأحكام ذات الصلة العائدة من كيان متوالية الاختبار القابلة للتنفيذ على النحو المعرف في التوصية Z.140 [2].

ويقوم الكيان T3RTS بتنفيذ عمليات استحداث وحذف مكونات اختبار TTCN-3، فضلاً عن التركيبات اللغوية للاتصالات المعتمدة على الرسائل والإجراءات ونداءات الوظيفة الخارجية وعمليات الإجراءات والمؤقتات. ويشمل ذلك إبلاغ مكيف النظام قيد الاختبار ومكيف المنتدى بالوظيفة الخارجية التي ستنفذ والمؤقتات التي ستبدأ، أو توقف أو تصطف أو تُقرأ. كذلك فإن T3RTS تبلغ كيان متوالية الاختبار القابلة للتنفيذ بالرسائل القادمة أو نداءات الإجراءات من النظام قيد الاختبار فضلاً عن أحداث المهلة.

وقبيل إرسال أو استقبال الرسائل ونداءات الإجراءات إلى أو من مكيف النظام قيد الاختبار أو مناولة نداءات الوظائف وعمليات الإجراءات في مكيف المنتدى لكيان متوالية الاختبار القابلة للتنفيذ، يقوم الكيان T3RTS بتنفيذ الكيان الخاص بنظام التشفير وفك التشفير لاستخدام وظائفها في التشفير وفك التشفير. وينبغي أن ينفذ الكيان T3RTS جميع عمليات الاتصالات المعتمدة على الرسائل والمعتمدة على الإجراءات فيما بين مكونات الاختبار باستثناء التركيبات اللغوية للنسخة TTCN-3 الخاصة بالاتصالات المعتمدة على الإجراءات مع النظام قيد الاختبار أي إمكانية وقف وإلغاء وقف تنفيذ مكون الاختبار والحماية بالمؤقتات الضمنية، والاستثناءات الخاصة بالمهلة الناجمة عن عمليات الاتصال. ويتعين أن تتحقق جميع عمليات الاتصالات المعتمدة على الإجراءات مع النظام قيد الاختبار، وتعّرف (في حالة عملية مستقبلة) في مكيف النظام قيد الاختبار حيث إنها تنفذ بأقصى قدر من الكفاءة بطريقة خاصة بالمنتدى. ويلاحظ أن توقيت أي عملية لنداء إجراء مثل المؤقتات الضمنية ينفذ في مكيف المنتدى.

ويطلب من الجوانب القابلة للتنفيذ في TTCN-3 بالمحافظة على صفوفها في المنفذ (التي تختلف عن تلك التي قد تتوافر في مكيف النظام قيد الاختبار أو مكيف المنتدى) لأحداث اختبار الدخل لأداء اللقطات الخاصة بعمليات الاستقبال على النحو المعرف في التوصية [2] Z.140. ويتعين الإبقاء على أحداث المهلة، التي استحدثتها علميات تنفيذ مؤقت TTCN-3 ومؤقت النداءات أو مؤقت حالة الاختبار في قائمة المهلة على النحو المحدد في التوصية [3] X.292. وقد أسندت جميع هذه الوظيفية في الشكل 2 للكيان T3RTS. فهو مسؤول عن حفظ الأحداث التي أبلغ مكيف النظام قيد الاختبار ومكيف المنتدى لها الكيان الخاص بالجوانب القابلة للتنفيذ في TTCN-3 والتي مازال يتعين تجهيزها.

3.2.1.4 نظام التشفير وفك التشفير (EDS)

يتولى هذا الكيان EDS مسؤولية تشفير وفك تشفير بيانات الاختبار التي تتضمن البيانات المستخدمة في عمليات الاتصالات مع النظام قيد الاختبار على النحو المحدد في تنفيذ وحدة TTCN-3. وفي حالة عدم تحديد أي تشفير في وحدة TTCN-3، يكون تشفير قيم البيانات قائماً على أساس الأداة. وينفذ هذا الكيان بواسطة الكيان T3RTS ويعود إليه. ويلاحظ أن الكيان EDS لا يتفاعل بصورة مباشرة مع مكيف النظام قيد الاختبار عن طريق السطح بيني لوقت تسيير TTCN-3.

4.2.1.4 المؤقتات في الجوانب القابلة للتنفيذ في TTCN-3

يمكن تصنيف المؤقتات التي تم إعلانها وتسميتها في متوالية الاختبار القابلة للتنفيذ في TTCN-3 من الناحية المفاهيمية على أنها صريحة في الجوانب القابلة للتنفيذ في TTCN-3. وتعرف المؤقتات التي استحدثتها تلك الجوانب لحراسة نداءات إجراء TTCN-3 أو تنفيذ عمليات بأنها ضمنية في هذه الجوانب. ويجرى استحداث كل من المؤقتات الصريحة والضمنية داخل الجوانب القابلة للتنفيذ في TTCN-3 إلا أنها تنفذ بواسطة مكيف المنتدى. ويتحقق ذلك بواسطة استحداث تعريف فريد للمؤقتات لأي مؤقت يستحدث في هذه الجوانب TE. وينبغي أن يمكّن هذا التعريف الفريد للمؤقت الجوانب المشار إليها TE من التفرقة بين مختلف المؤقتات. وسوف يستخدم التعريف الفريد للمؤقت في TE للتفاعل مع تنفيذ المؤقت المقابل في مكيف المنتدى.

ويلاحظ أنه تقع على عاتق TE مسؤولية تنفيذ مختلف التركيبات اللغوية في TTCN-3 لكل من المؤقتات الصريحة والضمنية بصورة صحيحة على النحو المعرف في التوصية [2] Z.140 مثل أن استخدام الكلمات الرئيسية الخاصة بأي من المؤقتات أو جميعها لا يسري إلاّ على المؤقتات الصريحة. وتستحدث جميع المؤقتات الخاصة بمكيف المنتدى، أي الصريحة والضمنية، بنفس الطريقة.

3.1.4 مكيف النظام قيد الاختبار (SA)

يتولى مكيف SA بتكييف الاتصالات المعتمدة على الرسائل والمعتمدة على الإجراءات في نظام اختبار TTCN-3 مع النظام قيد الاختبار (SUT) في منتدى إنجاز معين في نظام الاختبار. ويكون هذا الكيان على دراية بتقابل منافذ اتصالات مكون اختبار TTCN-3 مع منافذ السطح بيني لنظام الاختبار وينفذ السطح بيني الحقيقي لنظام الاختبار على النحو المعرف في التوصية Z.140 [2]. ويتحمل مسؤولية نشر طلبات الإرسال وعمليات إجراء النظام قيد الاختبار من الجوانب القابلة للتنفيذ في TTCN-3 إلى النظام قيد الاختبار، وإبلاغ TE بأي أحداث اختبار تستقبل من خلال إرفاقها بصفوف المنفذ الخاصة بالجوانب القابلة للتنفيذ في TE.

وتنفذ عمليات الاتصالات المعتمدة على الإجراءات مع النظام قيد الاختبار في مكيف النظام قيد الاختبار. ويتولى هذا الكيان الأخير SA المسؤولية عن التفريق بين مختلف الرسائل في الاتصالات المعتمدة على الإجراءات (أي النداء والرد والاستثناء) ونشرها بالطريقة الملائمة إما إلى النظام قيد الاختبار أو إلى TE. ويتولى الكيان TE مناولة التركيبات اللغوية للاتصالات المعتمدة على الإجراءات في TTCN-3 أي تأثيرات هذه العملية على إنجاز مكون اختبار TTCN-3.

ولمكيف النظام قيد الاختبار (SUT) سطح بيني مع TE الذي يستخدم في إرسال رسائل النظام قيد الاختبار (الصادرة في عمليات إجراء SUT) إلى SA وتبادل بيانات لاختبار المشفرة بين الكيانين في عمليات الاتصالات مع SUT.

4.1.4 مكيف المنتدى (PA)

ينفذ هذا الكيان الوظائف الخارجية للنسخة TTCN-3 ويوفر نظام اختبار TTCN-3 برمز زمني واحد. وفي هذا الكيان، سوف تنفذ الوظائف الخارجية فضلاً عن جميع المؤقتات، ويلاحظ أن حالات المؤقت تستحدث في TE. ولا يمكن تمييز المؤقت في مكيف المنتدى إلاّ بواسطة تعريف معرف المؤقت. ولذا فإن مكيف المنتدى (PA) يتعامل مع كل من المؤقتات الصريحة والضمنية بنفس الطريقة.

ويتيح السطح بيني مع TE تنفيذ الوظائف الخارجية وبدء المؤقتات وقراءتها ووقفها فضلاً عن السؤال عن المؤقتات باستخدام معرف المؤقت. ويبلغ مكيف المنتدى TE بالمؤقتات المنتهية.

2.4 السطوح البينية في نظام اختبار TTCN-3

كما سبق بيانه في الشكل 1، لنظام الاختبار في TTCN-3 سطحان بينيان هما السطح بيني للتحكم في TTCN-3 والسطح بيني لوقت تسيير TTCN-3 مما يحدد السطح بيني كياني إدارة الاختبار وكيانات الجوانب القابلة للتنفيذ في TTCN-3 وTE ومكيف النظام قيد الاختبار ومكيف المنتدى، على التوالي.

وتعرف هذه التوصية السطح بيني لوقت تسيير TTCN-3. وسيجرى تعريف TE مع SA وPA هنا من حيث عمليات TRI. وعلى الرغم من أنه يتعين تعريف كلا السطحين البينين أي TRI و TCI لتحقيق التنفيذ الكامل لنظام اختبار TTCN-3، فإن مواصفات TCI وتنفيذها تعتبر في الوقت الحاضر ملكية خاصة.

3.4 متطلبات تنفيذ نظام اختبار TTCN-3

سوف يعامل كل نداء عمليات السطح بيني لوقت تسيير TTCN-3 TRIعلى أنه عملية ذرية في الكيان الذي يقوم بالنداء. وسيعيد الكيان المنادى، الذي ينفذ عملية TRI، التحكم إلى الكيان الذي يقوم بالنداء بمجرد تحقيق تأثيره المتوخى أو في حالة عدم التمكن من استكمال العملية بنجاح ولن يقوم الكيان المنادى بوقف تنفيذ الاتصالات المعتمدة على الإجراءات. غير أن هذا الكيان سوف يقوم بالوقف بعد الانتهاء من تنفيذ الوظيفة الخارجية وانتظار قيمة عودتها. ويلاحظ، اعتماداً على تنفيذ نظام الاختبار، أن فشل العودة من تنفيذ الوظيفة الخارجية قد يؤدي إلى وقف لا نهائي لتنفيذ مكون الاختبار، والجوانب القابلة للتنفيذ في TTCN-3 ومكيف المنتدى أو حتى نظام الاختبار بأكمله.

ويمكن تحقيق متطلبات الإنجاز الواردة أعلاه خلال عملية تنفيذ نظام اختبار مندمج بصورة وثيقة فهنا ينفذ كامل نظام الاختبار في TTCN-3 من جانب أي عملية واحدة قابلة للتنفيذ حيث يخصص لكل كيان لنظام الاختبار تتابع واحد على الأقل من تتابعات الإنجاز. ويمكن تنفيذ عملية TRI هنا بوصفها من نداءات الإجراءات.

ويلاحظ أن الاندماج غير الوثيق لتنفيذ نظام الاختبار يظل ممكناً مثل تنفيذ نظام اختبار TTCN-3 بمكيفات SUT متعددة في بيئة حاسوبية موزعة. وفي هذه الحالة، فإن جزءاً صغيراً فقط من مكيف النظام قيد الاختبار SUT سوف يندمج بصورة وثيقة مع بقية نظام اختبار TTCN-3 حيث يمكن تحقيق مكيفات SA بعمليات منفصلة. ولن يمكن لهذا الجزء الصغير من SA إلاّ تنفيذ مسار للمعلومات المقدمة من عمليات TRI إلى عمليات مكيف النظام قيد الاختبار SUT التي ربما تكون في حالة تنفيذ على مضيف بعيد والعكس.

5 السطح بيني لوقت تسيير TTCN-3 وعملياته

يعرف هذا البند عمليات TRI من حيث الوقت الذي يمكن أن تستخدم فيه والتأثيرات المتوخاة في تنفيذ نظام اختبار. كما يجرى تعريف مجموعة من أنماط البيانات المجردة التي تستخدم بعد ذلك في تعريف عمليات TRI ويتضمن هذا التعريف أيضاً وصفاً أكثر تفصيلاً لمعلمات الدخل اللازمة لكل نداء عملية من عمليات TRI وقيم العودة الخاصة به.

1.5 عرض عام للسطح بيني لوقت تسيير TTCN-3 (TRI)

يعرف TRI التفاعل بين كيانات الجوانب القابلة للتنفيذ في TTCN-3 (TE) ومكيف النظام قيد الاختبار SUT ومكيف المنتدى (PA) داخل عملية تنفيذ نظام اختبار TTCN-3. ويوفر من الناحية المفاهيمية وسيلة تمكن TE من إرسال بيانات الاختبار إلى النظام قيد الاختبار أو معالجة المؤقتات، وكذلك إبلاغ TE ببيانات الاختبار المرسلة وأوقات المهلة.

ويمكن النظر إلى TRI على أنها تتألف من سطحين بينيين فرعيين، الاتصالات بشأن السطح بيني لوقت تسيير TTCN-3 والسطح بيني للمنتدى الخاص بهذا السطح البيني. ويعالج السطح بيني الخاص باتصالات بشأن السطح بيني لوقت تسيير TTCN-3 اتصالات متوالية الاختبار القابلة للتنفيذ في TTCN-3 مع النظام قيد الاختبار الذي ينفذ في SA. ويمثل السطح بيني لمنتدى السطح بيني لوقت تسيير TTCN-3 مجموعة من العمليات التي تكيف ETS على منتدى تنفيذ معين.

وكلا السطحين مزدوج الاتجاه حيث توحد الإجراء المنادية والمنادى عليها في الكيانات TE وA وPA في نظام الاختبار. ويبين الجدول بقدر أكبر من التفصيل علاقات المنادى والمنادى عليه فيما بين الكيانات ذات الصلة. ويلاحظ إن هذا الجدول لا يبين سوى التفاعلات المرئية عند TRI. ولم يبين الاتصالات الداخلية بين أجزاء نفس الكيان حيث إن البنية الداخلية لكل من TE وSA وPA قد تختلف في تنفيذ نظام اختبار TTCN-3.

الجدول Z.144/1 – عرض عام للسطوح البينية

السطح بيني	الاتجاه (الكيان المنادي ← والكيان المنادى عليه)	
الاسم	TE ← SA أو PA	SA أو PA ← TE
الاتصالات TRI	SA ← TE	TE ← SA
المنتدى TRI	PA ← TE	TE ← PA

1.1.5 السطح بيني للاتصالات TRI

يتألف هذا السطح البيني من عمليات ضرورية لتنفيذ اتصالات متوالية الاختبار القابلة للتنفيذ في TTCN-3 مع SUT ويتضمن عمليات لتدميث السطح بيني لنظام الاختبار وإقامة توصيلات إلى SUT ومناولة الاتصالات المعتمدة على الرسائل والمعتمدة على الإجراءات مع SUT. وعلاوة على ذلك، يقدم السطح بيني للاتصالات TRI عملية لإعادة تدميث مكيف SUT.

2.1.5 السطح بيني لمنتدى TRI

يتضمن هذا السطح بيني جميع العمليات اللازمة لتكييف الجوانب القابلة للتنفيذ في TTCN-3 مع منتدى إنجاز معين. ويوفر السطح بيني لمنتدى TRI وسائل لبدء ووقف وقراءة أحد المؤقتات، والسؤال عن حالته وإضافة أحداث مهلة لقائمة المؤقت المنتهي. وعلاوة على ذلك، يوفر عمليات لنداء الوظائف الخارجية في TTCN-3 وإعادة تدميث مكيف المنتدى. ويلاحظ أنه لا يوجد تفريق بين المؤقتات الصريحة والضمنية اللازمة عند السطح بيني لمنتدى TRI. وبدلاً من ذلك، فإن كل مؤقت سوف يعالج بصورة موحدة مع معرف المؤقت الخاص به.

3.1.5 الارتباط بين تنفيذ عمليات TTCN-3 وTRI

يتوافر لتنفيذ بعض عمليات TTCN-3 ارتباط مباشر بتنفيذ عملية TRI واحدة (أو ربما اثنتين في حالة عمليات تنفيذ ونداء TTCN-3) على النحو المبين في الجدول 2. وبالنسبة لتنفيذ جميع عمليات TRI، قد لا يكون هناك ارتباط مباشر.

ولا يستمر الارتباط بالنسبة لعمليات اتصالات TTCN-3 (أي الإرسال والنداء والرد والإصدار) إلاّ إذا نفذت هذه العمليات على منفذ مكونات اختبار يقابل منفذ TSI. وعلى ذلك، فإن هذا الارتباط يستمر بالنسبة لتنفيذ جميع هذه العمليات في حالة عدم تحديد مكون نظام في حالة الاختبار أي لا يستحدث سوى مكون اختبار MTC لحالة الاختبار وليس مكونات اختبار أخرى.

الجدول Z.144/2 – عرض الارتباط بين تنفيذ عمليات TTCN-3 وTRI (* = إذا كانت قابلة للتطبيق)

اسم العملية TTCN-3	اسم العملية TRI	السطح بيني TRI
execute	triExecuteTestCase	TriCommunication
	triStartTimer*	TriPlatform
map	triMap	TriCommunication
unmap	triUnmap	TriCommunication
send	triSend (انظر الملاحظة 1)	TriCommunication
	triSendBC (انظر الملاحظة 2)	
	triSendMC (انظر الملاحظة 3)	
call	triCall (انظر الملاحظة 1)	TriCommunication
	triCallBC (انظر الملاحظة 2)	
	triCallMC (انظر الملاحظة 3)	
	triStartTimer*	TriPlatform
reply	triReply (انظر الملاحظة 1)	TriCommunication
	triReply (انظر الملاحظة 2)	
	triReply (انظر الملاحظة 3)	
raise	triRaise (انظر الملاحظة 1)	TriCommunication
	triRaise (انظر الملاحظة 2)	
	triRaise (انظر الملاحظة 3)	
action	triSUTactionInformal	TriCommunication
start (timer)	triStartTimer	TriPlatform
stop (timer)	triStopTimer	TriPlatform
read (timer)	triReadTimer	TriPlatform
running (timer)	triTimerRunning	TriPlatform
TTCN-3 external function	triExternalFunction	TriPlatform
الملاحظة 1 – للاتصالات وحيدة الإذاعة.		
الملاحظة 2 – للاتصالات عريضة الإذاعة.		
الملاحظة 3 – للاتصالات متعددة الإذاعات.		

ويلاحظ أن جميع عمليات TRI المدرجة في الجدول 2 تستخدم بواسطة TE وأن هذه الأخيرة قد تقوم بتنفيذ هذه العمليات بصورة مختلفة لدى تقييم لقطاع TTCN داخل متوالية الاختبار القابلة للتنفيذ في TTCN-3.

2.5 مناولة الأخطاء

لا تحدد عملية مناولة الأخطاء الصريحة إلاّ لعمليات TRI التي تتطلبها الجوانب القابلة للتنفيذ في TTCN-3 وتبلغ SA أو PA عن حالة عملية TRI في القيمة الراجعة لعملية TRI. ويمكن أن تدل قيمة الحالة على النجاح المحلي (***TRI_OK***) أو الفشل (***TRI_Error***) في عملية TRI. ولذا قد تتصرف TE إزاء خطأ حدث إما داخل SA أو PA والإصدار مثل خطأ حالة اختبار.

وبالنسبة لعمليات TRI التي تطلبها SA أو PA لا تطلب أية مناولة للخطأ الصريح حيث لن هذه العمليات تنفذ في TE. وهنا تكون TE في وضع تحكم في إنجاز الاختبار في حالة حدوث خطأ في مثل هذه العملية TRI.

ويلاحظ أن شفرات الخطأ المحددة فضلاً عن اكتشاف ومناولة الأخطاء في أي كيان من كيانات نظام الاختبار تتجاوز نطاق مواصفات TRI الحالية.

3.5 السطح بيني للبيانات

لن تمر في عمليات TRI سوى بيانات الاختبار المشفرة. وتتولى الجوانب القابلة للتنفيذ في TTCN-3 مسؤولية تشفير بيانات الاختبار التي سترسل، وفك تشفير بيانات الاختبار المستقبلية في عمليات TRI المتعلق بكل منها حيث أن يمكن تحديد قواعد التشفير الخاصة بوحدة TTCN-3 أو داخلها. ويلاحظ أنه يتعين على TE أن تشفر بيانات الاختبار حتى لو لم تكن قد تدمث معلومات تشفير في متوالية الاختبار القابلة للتنفيذ في TTCN-3. وفي هذه الحالة، يتعين على مورد الأداء أن يعرف التشفير.

وبدلاً من تعريف سطح بيني صريح للبيانات لكل من نمطي بيانات TTCN-3 وASN.1، يعرف معيار TRI مجموعة من أنماط البيانات المجردة لكل تبين المعلومات التي يتعين أن تمر من الكيان المنادي إلى الكيان المنادى عليه والعكس. ويجرى في البندين 6 و7 تعريف التمثيل الملموس لهذه الأنماط من البيانات المجرد فضلاً عن تعريف أنماط البيانات الأساسية في تقابل لغات كل منها.

ويلاحظ أن القيم الخاصة بنمط بيانات المعرف ستكون فريدة في تنفيذ نظام الاختبار حيث يجرى تعريف الطابع الفريد بأنه الطابع المميز عالمياً في أي نقطة زمنية.

ويجرى تحديد واستخدام أنماط البيانات المجردة التالية لتعريف عمليات TRI.

1.3.5 التوصيل

`TriComponentIdType` تتضمن قيمة النمط `TriComponentIdType` معرفاً واسماً ونمط المكون. والقيمة المتميزة الأخيرة هي اسم المكون على النحو المحدد في متوالية الاختبار القابلة للتنفيذ في TTCN-3. ويستخدم هذا النمط المجرد بالدرجة الأولى على منافذ TSI لتسوية عمليات اتصالات TRI على منافذ TSI التي لها تقابلات مع الكثير من منافذ مكونات الاختبار.

`TriComponentIdListType` قيمة النمط `TriComponentIdListType` عبارة عن قائمة بهذا النمط `TriComponentIdType`. ويستخدم هذا النمط المجرد في الاتصالات متعددة الإذاعات في TCI.

`TriPortIdType` تتضمن قيمة النمط `TriPortIdType` قيمة النمط `TriComponentIdType` لتمثيل المكون الذي ينتمي إليه المنفذ، والرقم الدليل للمنفذ (إن وجد) واسم المنفذ على النحو المحدد في متوالية الاختبار القابلة للتنفيذ في TTCN-3. ويطلب النمط `TriPortIdType` بالدرجة الأخرى لتحديد المعلومات عن TSI والتوصيلات إلى TSI من TE إلى معرف النظام قيد الاختبار (SA).

`TriPortIdListType` قيمة النمط `TriPortIdListType` عبارة عن قائمة من الأنماط `TriPortIdType`. ويستخدم هذا النمط المجرد في أغراض التدميث بعد تنفيذ حالة الاختبار TTCN-3.

2.3.5 الاتصالات

`TriMessageType` قيمة النمط `TriMessageType` عبارة عن بيانات اختبار مشفرة سوف ترسل إلى النظام قيد الاختبار أو إنها استقبلت من ذلك النظام.

`TriAddressType` تتضمن قيمة النمط `TriAddressType` عنوان المصدر أو المقصد في النظام قيد الاختبار. ويمكن استخدام هذا النمط المجرد في عمليات الاتصالات TRI وهو نمط مفتوح غير منفذ إلى TE.

`TriAddressListType` قيمة النمط `TriAddressListType` عبارة عن قائمة من الأنماط `TriAddressType`. ويستخدم هذا النمط المجرد للاتصالات متعددة الإذاعات في TRI.

`TriSignatureIdType` قيمة النمط `TriSignatureIdType` عبارة عن اسم توقيع إجراء على النحو المحدد في متوالية الاختبار القابلة للتنفيذ. ويستخدم هذا النمط المجرد في عمليات اتصالات TRI المعتمدة على الإجراءات.

`TriParameterType` تتضمن قيمة النمط `TriParameterType` معلمة مشفرة وقيمة النمط `TriParameterPassingModeType` المحدد في المعلمة في متوالية الاختبار القابلة للتنفيذ في TTCN-3.

`TriParameterPassingModeType` قيمة النمط `TriParameterPassingModeType` يكون إما *داخل* أو *داخل خارج* أو *خارج* (*in*, *inout*, or *out*). ويستخدم هذا النمط المجرد في عمليات اتصالات TRI المعتمدة على الإجراءات ولنداءات الوظائف الخارجة.

`TriParameterListType` قيمة النمط `TriParameterListType` عبارة عن قائمة أنماط معلمات `TriParameterType`. ويستخدم هذا النمط المجرد في عمليات لاتصالات TRI المعتمدة على الإجراءات ولنداءات الوظائف الخارجية.

`TriExceptionType` قيمة النمط `TriExceptionType` عبارة عن نمط مشفر، وقيمة استثناء إما سيرسل إلى النظام قيد الاختبار أو استقبل من ذلك النظام. ويستخدم هذا النمط المجرد في عمليات اتصالات TRI المعتمدة على الإجراءات.

3.3.5 المؤقت

`TriTimerIdType` تحدد قيمة النمط `TriTimerIdType` معرفاً لمؤقت. وهذا النمط المجرد مطلوب في جميع عمليات مؤقت TRI.

`TriTimerDurationType` تحدد قيمة النمط `TriTimerDurationType` مدة مؤقت بالثواني.

4.3.5 متنوعات

`TriTestCaseIdType` قيمة النمط `TriTestCaseIdType` هي اسم حالة اختبار على النحو المعرف في متوالية الاختبار القابلة للتنفيذ في TTCN-3.

`TriFunctionIdType` قيمة النمط `TriFunctionIdType` هي اسم الوظيفة الخارجية المحددة في متوالية الاختبار القابلة للتنفيذ في TTCN-3.

`TriStatusType` قيمة النمط `TriStatusType` إما ***TRI_OK*** أو ***TRI_Error*** مما يشير إلى نجاح أو فشل عمليات TRI.

4.5 أوصاف العمليات

تعرف جميع تعاريف العمليات باستخدام لغة تعريف السطوح البينية (IDL). وتعرف التقابلات اللغوية الملموسة في البندين 6 و7.

وبالنسبة لكل نداء عملية من عمليات TRI، فإن جميع معلمات *داخل* و*داخل خارج* و*خارج* (*in*, *inout*, and *out*) المدرجة في تعريف عملية معنية هي معلمات إلزامية. وتحدد قيمة معلمات *الداخل* بواسطة الكيان طالب النداء. كذلك فإن قيمة معلمة *خارج* يحددها الكيان المنادي عليه. وفي حالة المعلمة *داخل خارج*، تحدد القيمة أولاً بواسطة الكيان طالب النداء إلا انه يمكن الاستعاضة عنها بقيمة جديدة يحددها الكيان المنادى عليه. ويلاحظ إنه على الرغم من أن TTCN-3 تستخدم أيضاً *داخل* و*داخل خارج* و*خارج* (*in*, *inout*, and *out*) في تعاريف التوقيع، فإن المعاني المستخدمة في مواصفات TRI وIDL لا يتعلق بتلك الواردة في مواصفات TTCN-3.

وينبغي أن تستخدم نداءات العمليات قيمة محتجزة للدلالة على انعدام المعلمات التي تعرف على أنها اختيارية في الوصف المقابل لمعلمات TRI. وتعرف القيم المحتجزة لهذه الأنماط في كل تقابل لغوي ويشار إليها فيما بعد أنها القيمة `null`.

وتوصف جميع الوظائف في السطح بيني باستخدام المقاس التالي:

F.n.m	**اسم العملية** الكيان طالب النداء ← الكيان المنادى عليه
التوقيع	لفة تعريف السطح بيني IDL – التوقيع
معلمات ''داخل''	وصف للبيانات التي تمر لمعلمات إلى العملية من الكيان طالب النداء إلى الكيان المنادى عليه
معلمات ''خارج''	وصف للبيانات التي تمر لمعلمات إلى العملية من الكيان طالب النداء إلى الكيان المنادى عليه
معلمات ''داخل خارج''	وصف للبيانات التي تمر لمعلمات إلى العملية من الكيان طالب النداء إلى الكيان المنادى عليه ومن الكيان المنادى عليه للعودة إلى الكيان طالب النداء
قيمة العودة	وصف لبيانات العودة من عملية إلى الكيان طلب النداء
القيود	وصف لأية قيود تسري على نداء العملية
التأثير	السلوك المطلوب من الكيان المطلوب قبل عودة العملية

5.5 عمليات السطح بيني للاتصالات

1.5.5 triSAReset (SA ← TE)

التوقيع	`TriStatusType triSAReset()`
معلمات ''داخل''	لا تنطبق
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة العودة لعملية `triSAReset` معرف النظام قيد الاختبار. وتبين حالة العودة النجاح المحلي (***TRI_OK***) أو الفشل (***TRI_Error***) للعملية
القيود	يمكن أن تطلب هذه العملية من TE في أي وقت لإعادة تدميث معرف النظام قيد الاختبار
التأثير	يعني قيام معرف النظام قيد الاختبار بإعادة تدميث جميع الاتصالات يحتفظ، مثل من خلال إعادة تدميث توصيلات ثابتة إلى النظام قيد الاختبار، بتوصيلات دينامية إلى النظام قيد الاختبار مع إهمال أية نداءات رسائل أو إجراءات وشيكة. وتعيد عملية `triSAResetSA` قيد الاختبار ***TRI_OK*** في حالة تحقيق العملية بنجاح و***TRI_Error*** في حالة عكس ذلك.

2.5.5 عمليات مناولة التوصيلات

1.2.5.5 (SA ← TE) triExecuteTestCase

التوقيع	`TriStatusType triExecuteTestCase(` `in TriTestCaseIdType testCaseId,` `in TriPortIdListType tsiPortList)`
معلمات ''داخل''	`testCaseId` معرف حالة الاختبار المتجهة إلى التنفيذ. `tsiPortList` قائمة منافذ `Tri` عبارة عن قائمة منافذ السطوح البينية لنظام الاختبار المعرفة لنظام الاختبار
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة تنفيذ عملية `triExecuteTestCase`. وتبين حالة العودة النجاح المحلي (***TRI_OK***) أو فشل (***TRI_Error***) للعملية
القيود	تطلب العملية من TE قبل تنفيذ أية حالة اختبار مباشرة. وتبين حالة الاختبار التي ستنفذ بمعرف `testCaseId`. وتحتوي `tsiPortList` على جميع المنافذ التي أعلنت في تعريف مكون النظام لأغراض حالة الاختبار أي منافذ TSI (السطح بيني لنظام الاختبار) وفي حالة عدم تعريف مكون للنظام بصورة صريحة لحالة الاختبار في متوالية الاختبار القابلة للتنفيذ، تحتوي قائمة منافذ TSIعندئذ جميع منافذ الاتصالات في مكون الاختبار MTC. ويجري ترتيب `tsiPortList` بالصورة التي تظهر بها في إعلان مكون TTCN-3 لكل منها.
التأثير	يمكن أن ينشئ معرف النظام قيد لاختبار SA أية توصيلات ثابتة على النظام قيد الاختبار ويدمث أية وسائل `triExecuteTestCase`. وتعيد عملية تنفيذ حالة الاختبار ***TRI_OK*** في حالة إتمام العملية بنجاح و***TRI_Error*** في حالة الفشل.

2.2.5.5 (SA ← TE) triMap

التوقيع	`TriStatusType triMap(` `in TriPortIdType compPortId,` `in TriPortIdType tsiPortId)`
معلمات ''داخل''	`compPortId` معرف منفذ مكون الاختبار الذي سيجري تقابله `tsiPortId` ويقابل معرف منفذ لنظام الاختبار من أجل نظام الاختبار
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية تنفيذ `triMap`. وتبين حالة العودة النجاح المحلي (***TRI_OK***) أو حالة الفشل (***TRI_Error***) للعملية
القيود	تطلب هذه العملية TE بعد تنفيذ عملية تقابل TTCN-3
التأثير	يمكن أن تنشئ SA توصيلة دينامية إلى النظام قيد الاختبار لمنفذ TSI المشار إليه. وتعيد عملية `triMap` في حالة عدم القدرة على إنشاء التوصيلة بنجاح ***TRI_Error*** أو ترسل ***TRI_OK*** في حالة النجاح. وينبغي أن تعيد العملية ***TRI_OK*** في حالة عدم الحاجة إلى إنشاء توصيلة دينامية بواسطة نظام الاختبار.

3.2.5.5 (SA ← TE) triUnmap

التوقيع	`TriStatusType triUnmap(` `in TriPortIdType compPortId,` `in TriPortIdType tsiPortId)`
معلمات ''داخل''	`compPortId` معرف منفذ مكون الاختبار الذي لن يتم تقابله `tsiPortId` معرف منفذ السطح بيني لنظام الاختبار الذي لن يتم تقابله
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triUnmap`. وتبين حالة العودة النجاح المحلي (***TRI_OK***) أو حالة الفشل (***TRI_Error***) للعملية
القيود	تطلب هذه العملية TE عندما تنفذ أية عملية لإلغاء تقابل TTCN-3
التأثير	تغلق SA التوصيلة الدينامية إلى SUT لمنفذ TSI المشار إليه. وتعيد عملية `triUnmap` الرمز ***TRI_Error*** عندما لا يمكن إغلاق التوصيلة بنجاح أو عدم إنشاء هذه التوصيلة السابقة أو ***TRI_OK***. وينبغي أن تعيد العملية ***TRI_OK*** عندما لا يتعين إنشاء توصيلة دينامية بواسطة نظام الاختبار.

3.5.5 عمليات الاتصالات المعتمدة على الرسائل

1.3.5.5 triSend (SA ← TE)

التوقيع	`TriStatusType triSend( in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriMessageType sendMessage)`
معلمات ''داخل''	`componentId` معرف إرسال مكون الاختبار `tsiPortId` معرف منفذ السطح بيني لنظام الاختبار عن طريق الرسالة التي ترسل إلى مكيف SUT `SUTaddress` عنوان المقصد (اختيارية) في SUT `sendMessage` إرسال الرسالة: إرسال الرسالة المشفرة
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triSend`. وتبين حالة العددية النجاح المحلي ***(TRI_OK)*** للعملية أو فشلها ***(TRI_Error)***
القيود	تطلب هذه العملية TE لدى إنجازها عملية إرسال وحيدة الإذاعة من TTCN-3 على منفذ مكون كان قد تقابل مع منفذ TSI. وتطلب هذه العملية من TE لجميع عمليات إرسال TTCN-3 في حالة عدم تحديد أي مكون من مكونات النظام أو حالة اختبار أي أن مكون اختبار MTC فقط هو الذي يستحدث لحالة الاختبار. وينبغي أن تتم عملية تشفير `sendMessage` في TE مثل نداء عملية TRI هذه.
التأثير	يمكن أن ترسل SA رسالة إلى SUT. وتعيد عملية `triSend` ***TRI_OK*** في حالة استكمالها بنجاح وإلا ترسل ***TRI_Error*** في عكس ذلك. ويلاحظ أن قيمة العودة ***TRI_OK*** لا تعني أن SUT قد استقبلت `sendMessage`.

2.3.5.5 (SA ← TE) triSendBC

التوقيع	`TriStatusType triSendBC(` `in TriComponentIdType componentId,` `in TriPortIdType tsiPortId,` `in TriMessageType sendMessage)`
معلمات ''داخل''	`componentId` معرف مكون الاختبار المرسال `tsiPortId` ومعرف منفذ السطح بيني لنظام الاختبار الذي ترسل الرسالة عن طريقه إلى مكيف SUT `sendMessage` الرسالة المشفرة التي سترسل
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triSend`. وتبين حالة العودة النجاح المحلي (***TRI_OK***) أو فشل (***TRI_Error***) العملية.
القيود	تطلب هذه العملية TE لدى إنجازها عملية إرسال الإذاعات المتعددة من TTCN-3 على منفذ مكون كان قد تقابل مع منفذ TSI. وتطلب هذه العملية من TE لجميع عمليات إرسال TTCN-3 في حالة عدم تحديد أي مكون من مكونات النظام أو حالة اختبار أي أن مكون اختبار MTC فقط هو الذي يستحدث لحالة الاختبار. وينبغي أن تتم عملية تشفير `sendMessage` في TE مثل نداء عملية TRI هذه.
التأثير	يمكن إذاعة SA رسالة إلى SUT. وتعيد عملية `triSend` ***TRI_OK*** في حالة استكمالها بنجاح وإلا ترسل ***TRI_Error*** في عكس ذلك. ويلاحظ إن قيمة العودة ***TRI_OK*** لا تعني أن SUT قد استقبلت `sendMessage`.

3.3.5.5 (SA ← TE) triSendMC

التوقيع	`TriStatusType triSendMC(` `in TriComponentIdType componentId,` `in TriPortIdType tsiPortId,` `in TriAddressListType SUTaddresses,` `in TriMessageType sendMessage)`
معلمات ''داخل''	`componentId` معرف مكون الاختبار المرسل `tsiPortId` ومعرف منفذ السطح بيني لنظام الاختبار الذي ترسل الرسالة عن طريقه إلى مكيف SUT `SUTaddresses` عنوان SUT عنوان المقصد داخل STU `sendMessage` الرسالة المشفرة التي سترسل
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triSend`. وتبين حالة العودة النجاح المحلي ***(TRI_OK)*** أو فشل ***(TRI_Error)*** العملية.
القيود	تطلب هذه العملية TE لدى إنجازها عملية إرسال الإذاعات المتعددة من TTCN-3 على منفذ مكون كان قد تقابل مع منفذ TSI. وتطلب هذه العملية من TE لجميع عمليات إرسال TTCN-3 في حالة عدم تحديد أي مكون من مكونات النظام أو حالة اختبار أي أن مكون اختبار MTC فقط هو الذي يستحدث لحالة الاختبار. وينبغي أن تتم عملية تشفير `sendMessage` في TE مثل نداء عملية TRI هذه.
التأثير	يمكن إذاعة SA عن طريق متعدد الإذاعات رسالة SUT. وتعيد عملية `triSend` ***TRI_OK*** في حالة استكمالها بنجاح وإلا ترسل ***TRI_Error*** في عكس ذلك. ويلاحظ أن قيمة العودة ***TRI_OK*** لا تعني أن SUT قد استقبلت `sendMessage`.

4.3.5.5 (TE ← SA) triEnqueueMsg

التوقيع	`void triEnqueueMsg(`	`in TriPortIdType tsiPortId,`
		`in TriAddressType SUTaddress,`
		`in TriComponentIdType componentId,`
		`in TriMessageType receivedMessage)`

معلمات ''داخل''	`tsiPortId`	معرف منفذ السطح بيني لنظام الاختبار الذي ستصف عن طريقه الرسالة بواسطة مكيف SUT
	`UTaddress`	عنوان SUT (اختيارية) عنوان المصدر في SUT.
	`componentId`	معرف مكون الاختبار المستقبل
	`receivedMessage`	الرسالة المشفرة المستقبلة
معلمات ''خارج''	لا تنطبق	
قيمة العودة	خالية	
القيود	تطلب هذه العملية SA بعد أن تتلقى الرسالة SUT ولا يمكن استخدامها إلا إذا كان `tsiPortId` قد سبق تقابله مع منفذ معرف مكون أو كان قد أحيل في البيان السابق `triExecuteTestCase`. وفي حالة تنفيذ عملية `triEnqueueMsg`، الرسالة `receivedMessage` سوف تحتوي على قيمة مشفرة.	
التأثير	هذه العملية سوف تمرر الرسالة إلى TE مبينة معرف المكون الذي سيتم تقابل `tsiPortId` معه. ويتعين فك تشفير الرسالة `receivedMessage` في TE.	

4.5.5 عمليات الاتصالات المعتمدة على الإجراءات

1.4.5.5 (SA ← TE) triCall

التوقيع	`TriStatusType triCall(`	`in TriComponentIdType componentId,`
		`in TriPortIdType tsiPortId,`
		`in TriAddressType SUTaddress,`
		`in TriSignatureIdType signatureId,`
		`in TriParameterListType parameterList)`

معلمات ''داخل''	`componentId`	معرف مكون الاختبار الذي يصدر نداء الإجراء
	`tsiPortId`	معرف منفذ السطح بيني لنظام الاختبار الذي يرسل عن طريق نداء الإجراء إلى معرف SUT
	`SUTaddress`	عنوان المقصد (اختياري) في STU.
	`signatureId`	معرف توقيع نداء الإجراء
	`parameterList`	قائمة بالمعلمات المشفرة تشكل جزءاً من التوقيع المبين والمعلمات في `parameterList` مرتبة بالشكل الذي تظهر في إعلان توقيع TTCN-3.
معلمات ''خارج''	لا تنطبق	
قيمة العودة	حالة عودة عملية `triCall`. وتبين حالة العودة النجاح المحلي (***TRI_OK***) أو فشل (***TRI_Error***) العملية.	
القيود	تطلب هذه العملية من TE عندما ينفذ عملية نداء الوحيد الإذاعة TTCN-3 على منفذ المكونات الذي تم تقابله مع منفذ TSI. وتطلب هذه العملية من TE لجميع عمليات نداء TTCN-3 في حالة عدم تحديد مكون نظام لحالة اختبار أي لا يستحدث مكون اختبار MTC إلا لحالة الاختبار. وجميع معاير إجراء *داخل* و*داخل خارج* ترد في القيم المشفرة. ومعلمات الإجراء هي المعلمات المحددة في مقاس توقيع TTCN-3. ويتعين أن يتم تشفيرها في TE قبل نداء عملية TRI هذه.	
التأثير	يمكن لدى تنفيذ هذه العملية أن يدمث SA نداء الإجراء الذي يقابل `signatureId`، `tsiPortId`. وسوف تعود عملية نداء tri دون انتظار لعودة نداء الإجراء الصادر. وتعيد `triCall` لدى نجاح تدميث نداء الإجراء *TRI_OK* والرمز *TRI_Error* إذا لم ينجح. ولن يبين SA أي خطأ في حالة أن تكون قيمة أي معلمات ''خارج'' غير صفرية. ويلاحظ أن قيمة العودة لعملية TRI هذه لا يصدر أي بيان عن نجاح أو فشل نداء الإجراء. ويلاحظ أن هناك قيمة مهملة اختيارية، يمكن تحديدها في متوالية الاختبار القابلة للتنفيذ في TTCN-3 لعملية النداء، غير مدرجة في توقيع عملية `triCall`. وTE مسؤولة عن معالجة هذه المسألة من خلال بدء مؤقت لعملية نداء TTCN-3 في PA مع نداء إجراء TRI منفصل أي `triStartTimer`.	
ملاحظة: يمكن تحقيق ذلك، مثلاً من خلال وضع خيط أو عملية جديدة. غير أن مناولة نداء الإجراء هذه يعتمد على تنفيذ TE.		

(SA ← TE) triCallBC 2.4.5.5

التوقيع	`TriStatusType triCallBC(` `in TriComponentIdType componentId,`
	`in TriPortIdType tsiPortId,`
	`in TriSignatureIdType signatureId,`
	`in TriParameterListType parameterList)`

معلمات ''داخل''	`componentId` معرف مكون الاختبار الذي يصدر نداء الإجراءات `tsiPortId` معرف منفذ السطح بيني لنظام الاختبار الذي سيتم عن طريقة إرسال نداء الإجراء إلى مكيف النظام قيد الاختبار SUT. `signatureId` معرف توقيع نداء الإجراء `parameterList` قائمة بالمعلمات المشفرة التي هي جزء من التوقيع المبين. وتأتي المعلمات في `parameterList` مرتبة بالصورة التي تبدو بها في إعلان توقيع TTCN-3
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triCall`. وتشير حالة العودة إلى النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	تطلب TE هذه العملية عندما تنفذ عملية النداء الإذاعي العريض في TTCN-3 على منفذ مكونات كان قد تم تقابله مع منفذ TSI. وتطلب هذه العملية بواسطة TE لجميع عمليات نداء TTCN-3 إذا لم يكن قد تم تحديد مكون نظام لحالة اختبار أي أن مكون اختبار MTC هو الذي يستحدث فقط لحالة اختبار. وتتضمن جميع معلمات إجراءات *الداخل* و*الداخل الخارج* قيماً مشفرة. ومعلمات الإجراءات هي المعلمات المحددة في مقياس توقيع TTCN-3. ويتعين أن يتم تشفيرها في TE مثل النداء الخاص بعملية TRI.
التأثير	لدى تنفيذ هذه العملية، يستطيع SA أن يدمث ويذيع نداء الإجراءات الذي يتواءم مع `signatureId` و`tsiPortId`. وسوف تعود عملية `triCall` دون انتظار لعودة نداء الإجراءات المعلن (انظر الملاحظة). وتعيد عملية TRI هذه *TRI_OK* في حالة نجاح تدميث نداء الإجراءات وإلا فإنها تعيد *TRI_Error* وأن تبين SA أية أخطاء إذا كانت قيمة أي معلمة خارج غير صفرية. ويلاحظ أن قيمة عودة هذه العملية TRI لا تصدر أي بيان عن نجاح أو فشل نداء الإجراءات. ويلاحظ أن قيمة المهملة الاختيارية، التي يمكن تحديدها في متوالية الاختبار القابلة للتنفيذ لعملية النداء، ليست مدرجة في توقيع عملية `triCall`. وتتحمل TE المسؤولية عن معالجة هذه المسألة من خلال بدء مؤقت لعملية نداء TTCN-3 في مكيف المنتدى بنداء عملية TRI منفصل أي بدء مؤقت `triStartTimer`.
ملاحظة: يمكن تحقيق ذلك بواسطة وضع خيط أو عملية جديدة غير أن هذه المناولة لنداء الإجراءات يعتمد على تنفيذ TE.	

3.4.5.5 (SA ← TE) triCallMC

التوقيع	`TriStatusType triCallMC(` `in TriComponentIdType componentId,`
	`in TriPortIdType tsiPortId,`
	`in TriAddressListType SUTaddresses,`
	`in TriSignatureIdType signatureId,`
	`in TriParameterListType parameterList)`

معلمات ''داخل''	`componentId` معرف مكون الاختبار الذي يصدر نداء الإجراءات `tsiPortId` معرف منفذ السطح بيني لنظام الاختبار الذي سيتم عن طريقة إرسال نداء الإجراء إلى مكيف النظام قيد الاختبار SUT. `SUTaddresses` عناوين مقصد داخل SUT اختياري `signatureId` قائمة بالمعلمات المشفرة التي هي جزء من التوقيع المبين. `parameterList` وتأتي المعلمات في قائمة المعلمات مرتبة بالصورة التي تبدو بها في إعلان توقيع TTCN-3
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triCall`. وتشير حالة العودة إلى النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	تطلب TE هذه العملية عندما تنفذ عملية النداء الإذاعي العريض في TTCN-3 على منفذ مكونات كان قد تم تقابله مع منفذ TSI. وتطلب هذه العملية بواسطة TE لجميع عمليات نداء TTCN-3 إذا لم يكن قد تم تحديد مكون نظام لحالة اختبار أي أن مكون اختبار MTC هو الذي يستحدث فقط لحالة اختبار. وتتضمن جميع معلمات إجراءات الداخل والداخل الخارج قيماً مشفرة. ومعلمات الإجراءات هي المعلمات المحددة في مقياس توقيع TTCN-3. ويتعين أن يتم تشفيرها في TE مثل النداء الخاص بعملية TRI.
التأثير	لدى تنفيذ هذه العملية، يستطيع SA أن يدمث ويذيع نداء الإجراءات الذي يتواءم مع `signatureId` ومنفذ TSI `tsiPortId`. وسوف تعود عملية `triCall` دون انتظار لعودة نداء الإجراءات المعلن (انظر الملاحظة). وتعيد عملية TRI هذه *TRI_OK* في حالة نجاح تدميث نداء الإجراءات، وإلا فإنها تعيد *TRI_Error* وأن تبين SA أية أخطاء إذا كانت قيمة أي معلمة خارج غير صفرية. ويلاحظ أن قيمة عودة هذه العملية TRI لا تصدر أي بيان عن نجاح أو فشل نداء الإجراءات. ويلاحظ إن قيمة المهملة الاختيارية، التي يمكن تحديدها في متوالية الاختبار القابلة للتنفيذ لعملية النداء، ليست مدرجة في توقيع عملية `triCall`. وتتحمل TE المسؤولية عن معالجة هذه المسألة من خلال بدء مؤقت لعملية نداء TTCN-3 في مكيف المنتدى بنداء عمليات TRI منفصل أي `triStartTimer`.
ملاحظة: يمكن تحقيق ذلك بواسطة وضع خيط أو عملية جديدة غير أن هذه المناولة لنداء الإجراءات يعتمد على تنفيذ TE.	

4.4.5.5 triReply (TE ← SA)

التوقيع	`TriStatusType triReply(`
	`in TriComponentIdType componentId,`
	`in TriPortIdType tsiPortId,`
	`in TriAddressType SUTaddress,`
	`in TriSignatureIdType signatureId,`
	`in TriParameterListType parameterList,`
	`in TriParameterType returnValue)`

معلمات ''داخل''	`componentId` معرف مكون الاختبار الذي يصدر الرد `tsiPortId` معرف منفذ السطح بيني لنظام الاختبار الذي يرسل عن طريقه الرد إلى مكيف SUT `SUTaddress` عنوان المقصد داخل SUT (اختياري) `signatureId` معرف توقيع نداء الإجراءات `parameterList` قائمة بالمعلمات المشفرة التي هي جزء من التوقيع المبين. وتأتي المعلمات في قائمة `parameterList` بالصورة التي تبدو بها في إعلان توقيع TTCN-3 `returnValue` قيمة العودة المشفرة لنداء الإجراءات (اختياري).
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triReply`. وتشير حالة العودة إلى النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	تطلب TE هذه العملية عندما تنفذ عملية النداء الإذاعي العريض في TTCN-3 على منفذ مكونات كان قد تم تقابله مع منفذ TSI. وتطلب هذه العملية بواسطة TE لجميع عمليات نداء TTCN-3 إذا لم يكن قد تم تحديد مكون نظام لحالة اختبار أي أن مكون اختبار MTC هو الذي يستحدث فقط لحالة اختبار. وتتضمن جميع معلمات إجراءات *الداخل* و*الداخل الخارج* قيماً مشفرة. معلمات الإجراءات هي المعلمات وقيمة العودة المحددة في مقياس توقيع TTCN-3. وتحتوي قائمة المعلمات `parameterList` الإجراءات. وهي تلك المعلمات المحددة في مقياس توقيع TTCN-3. ويتعين أن يتم تشفيرها في TE مثل نداء العملية TRI. في حالة عدم تعريف نمط عودة توقيع الإجراءات في متوالية الاختبار القابلة للتنفيذ، وسيجرى تمرير القيمة المميزة `null` لقيمة العودة.
التأثير	يمكن أن تصدر SA، لدى تنفيذ هذه العملية، رداً على نداء الإجراءات يتواءم مع `signatureId`، `tsiPortId`. وسوف تعيد عملية `triReply` ***TRI_OK*** في حالة تنفيذ هذه العملية بنجاح و***TRI_Error*** في حالة فشلها. وسوف تبين SA عدم وجود أخطاء في حالة اختلاف قيمة أي معلمة من *داخل* أو أي قيمة عودة غير محددة عن القيمة null.

(SA ← TE) triReplyBC 5.4.5.5

التوقيع	`TriStatusType triReplyBC(` `in TriComponentIdType componentId,`
	`in TriPortIdType tsiPortId,`
	`in TriSignatureIdType signatureId,`
	`in TriParameterListType parameterList,`
	`in TriParameterType returnValue)`

معلمات ''داخل''	`componentId` معرف مكون الرد على الاختيار `tsiPortId` معرف منفذ السطح البيني لنظام الاختبار الذي عن طريقه يرسل الرد إلى مكيف SUT `signatureId` معرف توقيع نداء الإجراءات `parameterList` قائمة بالمعلمات المشفرة التي هي جزء من التوقيع المبين. وتأتي المعلمات في قائمة `parameterList` بالصورة التي تبدو بها في إعلان توقيع TTCN-3 `returnValue` قيمة العودة المشفرة لنداء الإجراءات (اختياري).
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triReply`. وتشير حالة العودة إلى النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	تطلب TE هذه العملية عندما تنفذ عملية النداء الإذاعي العريض في TTCN-3 على منفذ مكونات كان قد تم تقابله مع منفذ TSI. وتطلب هذه العملية بواسطة TE لجميع عمليات نداء TTCN-3 إذا لم يكن قد تم تحديد مكون نظام لحالة اختبار أي أن مكون اختبار MTC هو الذي يستحدث فقط لحالة اختبار. وتتضمن جميع معلمات إجراءات *الداخل* و*الداخل الخارج* قيماً مشفرة. وتحتوي قائمة المعلمات معلمات نداء الإجراءات. وهي تلك المعلمات المحددة في مقياس توقيع TTCN-3. ويتعين أن يتم تشفيرها في TE مثل نداء العملية TRI في حالة عدم تعريف نمط عودة توقيع الإجراءات في متوالية الاختبار القابلة للتنفيذ، وسيجرى تمرير القيمة المميزة صفر `null` لقيمة العودة .
التأثير	يمكن أن تصدر SA، لدى تنفيذ هذه العملية، رداً على نداء الإجراءات يتواءم مع `signatureId`، `tsiPortId`. وسوف تعيد عملية `triReply` ***TRI_OK*** في حالة تنفيذ هذه العملية بنجاح و***TRI_Error*** في حالة فشلها. وسوف تبين SA عدم وجود أخطاء في حالة اختلاف قيمة أي معلمة من *داخل* أو أي قيمة عودة غير محددة عن القيمة null.

6.4.5.5 triReplyMC (SA ← TE)

التوقيع	`TriStatusType triReplyMC(`
	`in TriComponentIdType componentId,`
	`in TriPortIdType tsiPortId,`
	`in TriAddressListType SUTaddresses,`
	`in TriSignatureIdType signatureId,`
	`in TriParameterListType parameterList,`
	`in TriParameterType returnValue)`

معلمات ''داخل''	`componentId` معرف مكون الرد على الاختبار `tsiPortId` معرف منفذ السطح بيني لنظام الاختبار الذي سيتم عن طريقة إرسال نداء الإجراء إلى مكيف النظام قيد الاختبار SUT. `SUTaddresses` عناوين المقصد داخل SUT `signatureId` معرف توقيع نداء الإجراءات `parameterList` قائمة بالمعلمات المشفرة التي هي جزء من التوقيع المبين. وتأتي المعلمات في قائمة المعلمات مرتبة بالصورة التي تبدو بها في إعلان توقيع TTCN-3 `returnValue` قيمة العودة المشفرة لنداء الإجراءات (اختياري).
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triReply`. وتشير حالة العودة إلى النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	تطلب TE هذه العملية عندما تنفذ عملية النداء الإذاعي العريض في TTCN-3 على منفذ مكونات كان قد تم تقابله مع منفذ TSI. وتطلب هذه العملية بواسطة TE لجميع عمليات نداء TTCN-3 إذا لم يكن قد تم تحديد مكون نظام لحالة اختبار أي أن مكون اختبار MTC هو الذي يستحدث فقط لحالة اختبار. وتتضمن جميع معلمات إجراءات *الداخل* و*الداخل الخارج* *out* و*inout* قيماً مشفرة. وتحتوي قائمة المعلمات `parameterList` معلمات نداء الإجراءات. وهي تلك المعلمات المحددة في مقياس توقيع TTCN-3. ويتعين أن يتم تشفيرها في TE مثل نداء العملية TRI في حالة عدم تعريف نمط عودة توقيع الإجراءات في متوالية الاختبار القابلة للتنفيذ، وسيجرى تمرير القيمة المميزة صفر `null` لقيمة العودة .
التأثير	يمكن أن تصدر SA، لدى تنفيذ هذه العملية، رداً على نداء الإجراءات يتواءم مع `signatureId`، `tsiPortId`. وسوف تعيد عملية `triReply` ***TRI_OK*** في حالة تنفيذ هذه العملية بنجاح و***TRI_Error*** في حالة فشلها. وسوف تبين SA عدم وجود أخطاء في حالة اختلاف قيمة أي معلمة من *داخل* أو أي قيمة عودة غير محددة عن القيمة null.

7.4.5.5 (SA ← TE) triRaise

التوقيع	`TriStatusType triRaise(`
	`in TriComponentIdType componentId,`
	`in TriPortIdType tsiPortId,`
	`in TriAddressType SUTaddress,`
	`in TriSignatureIdType signatureId,`
	`in TriExceptionType exc)`

معلمات ''داخل''	`componentId` معرف مكون الاختبار المثير للاستثناء `tsiPortId` معرف منفذ السطح بيني لنظام الاختبار الذي سيرسل عن طريقة الاستثناء إلى مكيف SUT. `SUTaddress` عنوان المقصد داخل SUT (اختياري) `signatureId` معرف توقيع نداء الإجراءات الذي يرتبط به `exc` الاستثناء المشفر
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة العودة لعملية `triRaise`. وتشير حالة العودة إلى النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***).
القيود	تطلب TE هذه العملية عند تنفيذها لعملية إصدار وحيدة الإذاعة TTCN-3 على منفذ مكونات يكون قد تم تقابله مع منفذ Tsi. تتطلب TE هذه العملية لجميع عمليات إصدار TTCN-3 إذا لم يكن قد تم تحديد مكون نظام لحالة اختبار أي أن لا يستحدث سوى مكون اختبار MTC لحالة اختبار. ويتعين أن يتم تشفير الاستثناء في TE قبل نداء إجراءات Tri هذا.
التأثير	لدى تنفيذ هذه العملية، تستطيع SA أن تصدر استثناء لنداء الإجراءات الذي يتواءم مع `signatureId` و`tsiPortId`. وعودة عملية `triRaise` هي ***TRI_OK*** في حالة نجاح الاستثناء في العملية *TRI_Error* في حالة حدوث عكس ذلك.

8.4.5.5 (SA ← TE) triRaiseBC

التوقيع	`TriStatusType triRaiseBC(`
	`in TriComponentIdType componentId,`
	`in TriPortIdType tsiPortId,`
	`in TriSignatureIdType signatureId,`
	`in TriExceptionType exc)`

معلمات ''داخل''	`componentId` معرف مكون الاختبار المثير للاستثناء `tsiPortId` معرف منفذ السطح بيني لنظام الاختبار الذي سيرسل عن طريقة الاستثناء إلى مكيف SUT. `signatureId` معرف توقيع نداء الإجراءات الذي يرتبط به `exc` الاستثناء المشفر
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة العودة لعملية `triRaise`. وتشير حالة العودة إلى النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	تطلب TE هذه العملية عند تنفيذها لعملية إصدار وحيدة الإذاعة TTCN-3 على منفذ مكونات يكون قد تم تقابله مع منفذ Tsi. تتطلب TE هذه العملية لجميع عمليات إصدار TTCN-3 إذا لم يكن قد تم تحديد مكون نظام لحالة اختبار أي أن لا يستحدث سوى مكون اختبار MTC لحالة اختبار. ويتعين أن يتم تشفير الاستثناء في TE قبل نداء إجراءات Tri هذا.
التأثير	لدى تنفيذ هذه العملية، تستطيع SA أن تصدر استثناء لنداء الإجراءات الذي يتواءم مع `signatureId` و`tsiPortId`. وعودة عملية `triRaise` هي *TRI_OK* في حالة نجاح الاستثناء في العملية *TRI_Error* في حالة حدوث عكس ذلك.

9.4.5.5 triRaiseMC (SA ← TE)

التوقيع	`TriStatusType triRaiseMC( in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressListType SUTaddresses, in TriSignatureIdType signatureId, in TriExceptionType exc)`
معلمات ''داخل''	`componentId` معرف مكون الاختبار المثير للاستثناء `tsiPortId` معرف منفذ السطح بيني لنظام الاختبار الذي سيرسل عن طريقة الاستثناء إلى مكيف SUT `SUTaddresses` عناوين المقصد داخل SUT `signatureId` معرف توقيع نداء الإجراءات الذي يرتبط به `exc` الاستثناء المشفر
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة العودة لعملية `triRaise`. وتشير حالة العودة إلى النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	تطلب TE هذه العملية عند تنفيذها لعملية إصدار وحيدة الإذاعة TTCN-3 على منفذ مكونات يكون قد تم تقابله مع منفذ Tsi. تتطلب TE هذه العملية لجميع عمليات إصدار TTCN-3 إذا لم يكن قد تم تحديد مكون نظام لحالة اختبار أي أن لا يستحدث سوى مكون اختبار MTC لحالة اختبار. ويتعين أن يتم تشفير الاستثناء في TE قبل نداء إجراءات Tri هذا.
التأثير	لدى تنفيذ هذه العملية، تستطيع SA أن تصدر استثناء لنداء الإجراءات الذي يتواءم مع `signatureId`، `tsiPortId`. وعودة عملية `triRaise` هي *TRI_OK* في حالة نجاح الاستثناء في العملية *TRI_Error* في حالة حدوث عكس ذلك.

9.4.5.5 triEnqueueCall (TE ← SA)

التوقيع	`void triEnqueueCall( in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriComponentIdType componentId, in TriSignatureIdType signatureId, in TriParameterListType parameterList)`
معلمات ''داخل''	`tsiPortId` معرف منفذ السطح بيني لنظام الاختبار الذي عن طريقه سيصطف نداء الإجراءات بواسطة مكيف SUT `SUTaddress` هو عنوان المصدر داخل SUT (اختياري) `componentId` معرف مكون الاختبار المستقبل `signatureId` معرف توقيع نداء الإجراءات `parameterList` قائمة المعلمات المشفرة التي هي جزء من التوقيع المبين. وتأتي المعلمات في قائمة المعلمات مرتبة بالشكل الذي تظهر به في إعلان توقيع TTCN-3. وصف البيانات التي تمر لمعلمات إلى العملية من الكيان طالب النداء إلى الكيان المنادى عليه.
معلمات ''خارج''	لا تنطبق
قيمة العودة	فارغة
القيود	يمكن أن تطلب SA هذه العملية بعد تلقي نداء إجراءات من SUT. ولا يمكن استخدامها إلا عندما يكون `tsiPortId` إما سبق تعامله مع منفذ `componentId` أو أشير إليه في بيان سابق عن `triExecuteTestCase`. وأدى تنفيذ عملية `triEnqueueCall`، تحتوي جميع معلمات إجراءات *داخل وداخل خارج* قيماً مشفرة.
التأثير	ويمكن أن يضع TE نداء الإجراءات هذا في صف مع `signatureId` عند منفذ `componentId` التي تقابل معه `tsiPortId`. ويتعين أن يتم فك تشفير معلمات الإجراءات بواسطة TE. وستعين TE عدم وجود خطأ في حالة أن تكون قيمة أي معلمة *خارج* مختلفة عن null.

11.4.5.5 triEnqueueReply (TE ← SA)

التوقيع	`void triEnqueueReply(`	`in TriPortIdType tsiPortId,`
		`in TriAddressType SUTaddress,`
		`in TriComponentIdType componentId,`
		`in TriSignatureIdType signatureId,`
		`in TriParameterListType parameterList,`
		`in TriParameterType returnValue)`

معلمات ''داخل''	`tsiPortId`	معرف منفذ السطح بيني لنظام الاختبار الذي عن طريقه سيصطف الرد بواسطة مكيف SUT
	`SUTaddress`	عنوان المصدر داخل SUT (اختياري)
	`componentId`	معرف مكون الاختبار المستقبل
	`signatureId`	معرف توقيع نداء الإجراءات
	`parameterList`	قائمة المعلمات هي قائمة تضم المعلمات المشفرة التي هي جزء من التوقيع المبين. وتأتي المعلمات `parameterList` في قائمة المعلمات مرتبة بالشكل الذي تظهر به في إعلان توقيع
	`returnValue`	قيمة العودة (اختيارية) هي قيمة العودة المشفرة لنداء الإجراءات.
معلمات ''خارج''	لا تنطبق	
قيمة العودة	فارغة	
القيود	يمكن أن تطلب SA هذه العملية بعد تلقي نداء إجراءات من SUT. ولا يمكن استخدامها إلا عندما يكون `tsiPortId` إما سبق تعامله مع منفذ `componentId` أو أشير إليه في بيان سابق عن `triExecuteTestCase`. وأدى تنفيذ عملية `triEnqueueReply`، تحتوي جميع معلمات إجراءات *داخل وداخل خارج* قيماً مشفرة. وفي حالة عدم تعريف نمط عودة لتوقيع الإجراءات في متوالية الاختبار القابلة للتنفيذ، تستخدم القيمة المميزة `null` لقيمة العودة.	
التأثير	ويمكن أن يضع TE نداء الإجراءات هذا في صف مع `signatureId` عند منفذ `componentId` التي تقابل معه منفذ TSI `tsiPortId`. ويتعين أن يتم فك تشفير معلمات الإجراءات بواسطة TE. وستعين TE عدم وجود خطأ في حالة أن تكون قيمة أي معلمة *خارج* مختلفة عن null.	

12.4.5.5 triEnqueueException (TE ← SA)

التوقيع	`void triEnqueueException(`	`in TriPortIdType tsiPortId,`
		`in TriAddressType SUTaddress,`
		`in TriComponentIdType componentId,`
		`in TriSignatureIdType signatureId,`
		`in TriExceptionType exc)`

معلمات ''داخل''	`tsiPortId`	معرف منفذ السطح بيني لنظام الاختبار الذي عن طريقه سيصطف نداء الاستثناء بواسطة مكيف SUT
	`SUTaddress`	عنوان المصدر داخل SUT
	`componentId`	معرف بدء مكونة استقبال الاختبار
	`signatureId`	معرف توقيع نداء الإجراءات الذي يرتبط به
	`exc`	الاستثناء المشفَّر
معلمات ''خارج''	لا تنطبق	
قيمة العودة	فارغة	
القيود	يمكن أن تطلب SA هذه العملية بعد تلقي الرد من SUT. ولا يمكن استخدامها إلا عندما يكون معرف المنفذ `tsiPortId` سبق تقابله لمنفذ معرف المكونات `componentId` أو أشير إليه في بيان سابق `triExecuteTestCase` عن تنفيذ حالة اختبار. وأدى تنفيذ عملية نداء صفوف tri، تحتوي جميع معلمات إجراءات داخل وداخل خارج قيماً مشفرة. وعند تنفيذ عملية استثناء اصطفاف `triEnqueueException`، يحتوي الاستثناء على قيمة مشفرة.	
التأثير	يمكن أن تصف TE هذا الاستثناء لنداء الإجراءات مع معرف التوقيع `signatureId` عند منفذ معرف المكونات `componentId` الذي يتقابل معه منفذ TSI. ويتعين أن تتم عملية فك تشفير الاستثناء داخل TE.	

5.5.5 عمليات متنوعة

1.5.5.5 (SA ← TE) triSUTactionInformal

التوقيع	`TriStatusType triSUTactionInformal(in string description)`
معلمات ''داخل''	`description` وصف غير رسمي لإجراء يعتمد على SUT
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triSUTactionInformal`. تبين حالة العودة إلى النجاح المحلي للعملية (***TRI_OK***) والفشل (***TRI_Error***).
القيود	تطلب TE هذه العملية عند تنفيذ عملية إجراء SUT في TTCN-3 التي لا تحتوي إلا على سلسلة.
التأثير	لدى تنفيذ هذه العملية، تدمث SA الإجراء الموصوف الذي سيتخذ على SUT مثل التحول إلى، والتدميث أو إرسال رسالة إلى SUT. وتعيد عملية الإجراء غير الرسمي `triSUTactionInformal` عبارة *TRI_OK* في حالة تنفيذ العملية بنجاح، و*TRI_Error* إذا كان الأمر عكس ذلك. ويلاحظ أن قيمة العودة لهذه العملية TRI لا تقدم أي بيان بشأن نجاح أو فشل الاجارءات التي ستتخذ بشأن SUT.

2.5.5.5 (SA ← TE) triSUTactionTemplate

تقادمت.

6.5 عمليات السطح بيني للمنتدى

1.6.5 (PA ← TE) triPAReset

التوقيع	`TriStatusType triPAReset()`
معلمات ''داخل''	لا تنطبق
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triPAReset`. وتبين حالة العودة النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	يمكن أن تطلب هذه العملية بواسطة TE في أي وقت لإعادة تدميث معرف المنتدى
التأثير	يقوم معرف المنتدى بإعادة تدميث جميع أنشطة التدميث التي تعمل في الوقت الحاضر مثل وقف جميع مؤقتات العودة، وإهمال عمليات مهملة وشيكة للمؤقتات المنتهية وتعيد عملية إعادة `triResetSA` عبارة *TRI_OK* في حالة أداء العملية بنجاح و*TRI_Error* في حالة الفشل

2.6.5 عمليات المؤقت

1.2.6.5 (PA ← TE) triStartTimer

التوقيع	`TriStatusType triStartTimer( in TriTimerIdType timerId,` `in TriTimerDurationType timerDuration)`
معلمات ''داخل''	`timerId` معرف حالة المؤقت `timerDuration` خلال عمل المؤقت بالثواني
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triStartTimer`. وتبين حالة العودة النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	يطلب هذه العملية TE عندما يتعين المؤقت
التأثير	سوف يبدأ معرف المنتدى لدى تنفيذ هذه العملية، المؤقت المبين بالمدة المبينة. ويعمل المؤقت من القيمة صفر (0,0) حتى الحد الأقصى الذي تحدده مدة المؤقت. وإذا يبين المؤقت من خلال `timerId` أنه يعمل بالفعل، يتعين إعادة بدئه، وعندما ينتهي المؤقت، يطلب معرف المنتدى عملية `triTimeout()` مع `timerId`. وتعيد عملية بدء المؤقت `triStartTimer` عبارة *TRI_OK* في حالة البدء بنجاح و*TRI_Error* في حالة فشلها

2.2.6.5 triStopTimer (PA ← TE)

التوقيع	`TriStatusType triStopTimer( in TriTimerIdType timerId)`
معلمات ''داخل''	`timerId` معرف حالة المؤقت
معلمات ''خارج''	لا تنطبق
قيمة العودة	حالة عودة عملية `triStopTimer`. وتبين حالة العودة النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	تطلب TE هذه العملية عندما يتعين وقف المؤقت
التأثير	سوف يبدأ معرف المنتدى لدى تنفيذ هذه العملية، المؤقت المبين بالمدة المبينة. ويعمل المؤقت من القيمة صفر (0,0) حتى الحد الأقصى الذي تحدده مدة المؤقت. وإذا يبين المؤقت من خلال `elapsedTime` أنه يعمل بالفعل، يتعين إعادة بدئه، وعندما ينتهي المؤقت، يطلب معرف المنتدى عملية مهملة () مع معرف المؤقت. وتعيد عملية وقف المؤقت `triStopTimer` عبارة ***TRI_OK*** في حالة البدء بنجاح و***TRI_Error*** في حالة فشلها ويلاحظ أن وقف المؤقت المفعل يُعتبر عملية سليمة وفي هذه الحالة تكون العودة ***TRI_OK***.

3.2.6.5 triReadTimer (PA ← TE)

التوقيع	`TriStatusType triReadTimer( in TriTimerIdType timerId, out TriTimerDurationType elapsedTime)`
معلمات ''داخل''	`timerId` معرف حالة المؤقت
معلمات ''خارج''	`elapsedTime` قيمة الوقت المنقضي منذ أن بدأ المؤقت بالثواني
قيمة العودة	حالة عودة عملية `triReadTimer`. وتبين حالة العودة النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***)
القيود	يمكن طلب هذه العملية بواسطة TE عندما يتعين تنفيذ يمكن قراءة مؤقت على المؤقت المشار إليه (انظر 1.3.5)
التأثير	لدى تنفيذ هذه العملية، يستخدم معرف المنتدى `timerId` للنفاذ إلى الوقت الذي انقضى منذ أن بدأ هذا المؤقت. وسوف تقدم قيمة `elapsedTime` بالثواني. وسوف تعيد قراءة مؤقت معطل أي المؤقت الذي لم يبدأ بعد أو الذي انتهى، قيمة الوقت المنقضي البالغ صفرا. وتعيد عملية قراءة المؤقت `triReadTimer` عبارة ***TRI_OK*** عند أداء هذه العملية بنجاح و***TRI_Error*** إذا كان الأمر غير كذلك.

4.2.6.5 triTimerRunning (PA ← TE)

التوقيع	`TriStatusType triTimerRunning( in TriTimerIdType timerId, out boolean running)`
معلمات داخل	`timerId` معرف حالة المؤقت
معلمات خارج	`running` حالة المؤقت
قيمة العودة	حالة عودة. عملية `triTimerRunning`. وتبين حالة العودة النجاح المحلي للعملية بأنه (***TRI_OK***) والعكس من ذلك (***TRI_Error***)
القيود	يمكن أن تطلب TE هذه العملية عندما يتعين تنفيذ عملية مؤقت تسيير TTCN-3 على المؤقت المبين (انظر 1.3.5).
التأثير	لدى تنفيذ هذه العملية، يستخدم معرف المنتدى `timerId` للنفاذ إلى حالة المؤقت. وتدمث العملية التسيير إلى القيمة البولينية إذا، وإذا فقط أعادت عملية تسيير المؤقت `triTimerRunning` إذا كانت حالة المؤقت قد تحددت بنجاح ***TRI_OK*** أو ***TRI_Error*** في حالة عكس ذلك.

(TE ← PA) triTimeout 5.2.6.5

التوقيع	`void triTimeout(in TriTimerIdType timerId)`
معلمات داخل	`timerId` معرف حالة المؤقت
معلمات خارج	لا تنطبق
قيمة العودة	فارغة
القيود	هذه العملية تستدعي من PA بعد انتهاء مدة المؤقت الذي كان قد بُدئ باستخدام عملية `triStartTimer` – أي أنه بلغ أقصى قيمة للمدة.
التأثير	يطلب PA هذه العملية بعد أن يكون المؤقت، الذي سبق أن بدأ باستخدام عملية بدء `timerId`، قد انتهى. ويحصل أن تنفذ هذه العملية في TE بطريقة يمكن بها معالجة مختلف التركيبات اللغوية للنسخة TTCN-3 بالنسبة للمؤقتات المعرفة في التوصية Z.143 [4] (انظر أيضاً 1.3.5)

3.6.5 عمليات متنوعة

(TE ← PA) triExternalFunction 1.3.6.5

التوقيع	`TriStatusType triExternalFunction( in TriFunctionIdType functionId, inout TriParameterListType parameterList, out TriParameterType returnValue)`
معلمات داخل	`functionId` معرف الوظيفة الخارجية
معلمات خارج	`returnValue` قيمة العودة المشفرة (اختيارية)
معلمات دخل خارج	`ParameterList` قائمة من المعلمات المشفرة للوظيفة المبينة. وتأتي المعلمات في `parameterList` مرتبة بالصورة التي تظهر بها في إعلان وظيفة TTCN-3.
قيمة العودة	حالة عودة عملية `triExternalFunction`. وتبين حالة العودة النجاح المحلي للعملية (***TRI_OK***) أو فشلها (***TRI_Error***).
القيود	تطلب TE هذه العملية عندما تنفذ وظيفة معرفة بأنها خارجية عن TTCN-3 (أي أن جميع الوظائف غير الخارجية تنفذ داخل TE). ولدى تنفيذ عملية `triExternalFunction` بواسطة TE تحتوي جميع معلمات وظيفة *داخل* و*داخل خارج* قيماً مشفرة. ولن يشير PA إلى أي خطأ في حالة أن تكون قيمة أي معلمات خارج غير صفرية.
التأثير	لكل وظيفة خارجية محددة في متوالية لاحتجاز القابلة للتنفيذ على TTCN-3، سينفذ PA السلوك. ولدى تنفيذ هذه العملية، سينفذ PA الوظيفة المبينة بواسطة المعرف `functionId`. وسوف يتم نفاذها إلى معلمات وظيفته *داخل* و*داخل خارج* في `parameterList`، وتقييم الوظيفة الخارجية باستخدام قيم هذه المعلمات وحساب القيم الخاصة بكل من معلمات *داخل خارج*، و*خارج* في `parameterList`، وتعيد العملية عندئذ القيم المشفرة لجميع معلمات وظائف داخل خارج وخارج وقيمة العودة المشفرة للوظيفة الخارجية. وفي حالة عدم تعريف أي نمط عودة لهذه الوظيفة الخارجية على متوالية الاختبار القابلة للتنفيذ، تستخدم القيمة المميزة `null` للأخيرة. وتعيد عملية `triExternalFunction` ***TRI_OK*** إذا استكمل PA تقييم الوظيفة الخارجية بنجاح و***TRI_Error*** إذا كان غير ذلك. يلاحظ إنه في حين أن جميع عمليات Tri تعتبر غير مثيرة للاعتراض، فإن عملية `triExternalFunction` تعتبر مسببة لذلك. ويعني ذلك أن العملية لن تعود قبل تقييم الوظيفة الخارجية المبينة بصورة كاملة. ويتعين تنفيذ الوظائف الخارجية بعناية حيث إنها قد تتسبب في *وقف* تنفيذ مكونات الاختبار أو حتى تنفيذ نظام الاختبار بأكمله.

6 تقابل لغة جافا

1.6 مقدمة

يتناول هذا البند تقابل لغة جافا على السطح بيني لوقت تسيير TTCN-3. وتحقيقاً للكفاءة، يقدم تقابل لغة مخصص بدلاً من استخدام لغة تعريف السطح بيني في مجموعة إدارة الأهداف بالنسبة للغة جافا.

ويعرّف تقابل لغة جافا للسطح بيني لوقت تسيير TTCN-3 الكيفية التي يتم بها تقابل التعاريف الواردة في لغة تعريف السطح بيني التي يرد وصف لها في البند 5 مع لغة جافا. وتقابل اللغات مستقل عن نسخة جافا المستخدمة حيث لا تستخدم سوى بنيات لغة جافا الأساسية.

2.6 الأسماء والنطاقات

1.2.6 الأسماء

على الرغم من عدم وجود تضارب بين المعرفات في تعريف لغة تعريف السطح بيني، ولغة جافا، فإن بعض قواعد ترجمة التسميات تطبق على معرفات تعريف السطح بيني.

- تبدأ معرفات معلمات جافا بحروف صغيرة في حين تبدأ الأجزاء اللاحقة التي تقيم معرف المعلمات بحروف كبيرة. فعلى سبيل المثال، فإن معرف معلمات لغة تعريف السطح البيني `sutAddress` تتقابل مع **`SUTaddress`** في جافا.
- تحذف سطوح جافا البينية أو معرفات الفئة النمط المتخالف المستخدم في تعريف لغة تعريف السطح بيني IDL. مثال ذلك نمط IDL وهو **`TriPortIdType`** يقابل `TriPortId` في جافا.

ويتواءم التقابل الناجم مع مبادئ تشفير جافا المعيارية.

2.2.6 النطاقات

يجرى تقابل وحدة **`triInterface`** في IDL مع حزمة جافا `org.etsi.ttcn3.tri`. ويجرى تقابل جميع الإعلانات من نمط IDL داخلي هذه الوحدة مع إعلانات الفئات أو السطح البينية لجافا داخل هذه الحزمة.

3.6 تقابل النمط

1.3.6 تقابل نمط أساسي

يقدم الجدول 3 عرضاً عاماً للكيفية التي يتم تقابل أنماط IDL الأساسية المستخدمة مع أنماط جافا.

الجدول Z.144/3 – تقابلات النمط الأساسي

نمط IDL	نمط جافا
بولانية	`org.etsi.ttcn.tri.TriBoolean`
سلسلة	`java.lang.String`

ولا تستخدم أنماط IDL الأساسية الأخرى في تعريف IDL

1.1.3.6 النمط البولاني

يتقابل النمط البولاني في IDL مع السطح بيني `org.etsi.ttcn.tri.TriBoolean` حتى يمكن للأهداف التي تنفذ هذا السطح بيني أن تعمل كأهداف ماسكة.

ويعرف السطح بيني الآتي من أجل: org.etsi.ttcn.tri.TriBoolean

```
// TriBoolean
package org.etsi.ttcn.tri;
public interface TriBoolean {
    public void setBooleanValue(boolean value);
    public boolean getBooleanValue();
}
```

1.1.1.3.6 الطرائق

setBooleanValue(boolean value) -
تدميث TriBoolean على القيمة البولانية value

getBooleanValue() -
إعادة القيمة البولانية الممثلة في هذا النمط البولاني TriBoolean

2.1.3.6 السلسلة

يقابل نمط **string** في IDL مع فئة سلسلة لغة جافا دون التحقق من المدى أو قيود بالنسبة للواسمات في السلسلة. ويمكن تحويل جميع السلاسل الممكنة في TTCN-3 إلى سلسلة لغة جافا.

2.3.6 تقابل النمط المبنين

يعرف الوصف في TRI IDL الأنماط المعرفة من قبل المستعملين بوصفها أنماطا محلية. وفي تقابل لغة جافا، تقابل هذه الأنماط مع السطوح البينية لجافا. وتعرف السطوح البينية الطرائق والنعوت المتيسرة للأهداف التي تقدم بتنفيذ هذا السطح بيني.

TriPortIdTyp 1.2.3.6

نمط معرف منفذ **TriPortIdType**:

```
// TRI IDL TriPortIdType// TRI IDL TriPortIdType
package org.etsi.ttcn.tri;
public interface TriPortId {
    public String getPortName();
    public TriComponentId getComponent();
    public boolean isArray();
    public int getPortIndex();
}
```

1.1.2.3.6 الطرائق

getPortName() -
يعيد اسم المنفذ على النحو المعرف في مواصفات TTCN-3.

getComponent() -
يعيد معرف المكون الذي ينتمي له TriPortId على النحو الوارد في مواصفات TTCN-3

isArray() -
يعيد true إذا كان هذا المنفذ جزءاً من صفيف المنفذ وfalse إذا كان العكس.

getPortIndex() -
يعيد الرقم الدليلي للمنفذ إذا كان هذا المنفذ جزءاً من صفيف المنفذ الذي يبدأ من صفر. أما إذا لم يكن المنفذ جزءاً من صفيف المنفذ عندئذ يعادل 1.

TriPortIdListType 2.2.3.6

يقابل **TriPortIdListType** مع السطوح البينية التالية:

```
// TRI IDL TriPortIdListType
package org.etsi.ttcn.tri;
public interface TriPortIdList {
    public int size();
    public boolean isEmpty();
    public java.util.Enumeration getPortIds();
    public TriPortId get(int index);
}
```

الطرائق 1.2.2.3.6

`size()`

يعيد عدد المنافذ في هذه القائمة.

`isEmpty()`

يعيد `true` إذا كانت القائمة لا تحتوي على منافذ.

`getPortIds()`

يعيد `Enumeration` على المنافذ في القائمة. ويقدم الترقيم المنافذ بنفس الترتيب الذي تظهر به في القائمة.

`get(int index)`

يعيد معرف `TriPortId Tri` عند الوضع المحدد.

TriComponentIdType 3.2.3.6

TriComponentIdType تقابل مع السطوح البينية التالية:

```
// TRI IDL TriComponentIdType
package org.etsi.ttcn.tri;
public interface TriComponentId {
    public String getComponentId();
    public String getComponentName();
    public String getComponentTypeName();
    public TriPortIdList getPortList();
    public boolean equals(TriComponentId port);
}
```

الطرائق 1.3.2.3.6

`getComponentId()` -

تعيد تمثيل معرف المكون الفريد المشار إليه.

`getComponentName()` -

تعيد اسم المكون على النحو المعرف في مواصفات TTCN-3 وفي حالة عدم تقديم اسم، تعاد سلسلة فارغة.

`getComponentTypeName()` -

تعيد اسم نمط المكون على النحو المعرف في مواصفات TTCN-3.

`getPortList()` -

تعيد قائمة منفذ المكون على النحو المعرف في مواصفات TTCN-3.

`equals(TriComponentId component)` -

تقارن المكون لمعرف `TriComponentId` المشار إليه للتأكد من التكافؤ وتعيد `true` إذا، وإذا فقط، كان لكلا المكونين نفس تمثيل معرف المكونات الفريد المشار إليه أو `false` إذا كان غير ذلك.

TriComponentIdListType 4.2.3.6

TriComponentIdListType تقابل مع السطوح البينية التالية:

```
// TRI IDL TriComponentIdListType
package org.etsi.ttcn.tri;
public interface TriComponentIdListType {
    public int size();
    public boolean isEmpty();
    public java.util.Enumeration getComponents();
    public TriComponentId get(int index);
    public void clear();
    public void add(TriComponentId comp);
}
```

1.4.2.3.6 الطرائق

`size()`
يعيد عدد المكونات في هذه القائمة.

`isEmpty()`
يعيد `true` إذا كانت القائمة لا تحتوي مكونات.

`getComponents()`
تعيد `Enumeration` على المكونات في القائمة. ويوفر الترقيم المكونات بنفس الترتيب الذي تظهر به في القائمة.

`get(int index)`
يعيد معرف `TriComponentId` عند موقع معين.

`clear()`
يحذف جميع المكونات من `TriComponentIdList`.

`add(TriComponentId comp)`
تضاف مكونات في نهاية هذه `TriComponentIdList`.

5.2.3.6 TriMessageType

`TriMessageType` تقابل مع السطوح البينية التالية:

```
// TRI IDL TriMessageType
package org.etsi.ttcn.tri;
public interface TriMessage {
    public byte[] getEncodedMessage();
    public void setEncodedMessage(byte[] message);
    public boolean equals(TriMessage message);
}
```

1.5.2.3.6 الطرائق

`getEncodedMessage()` -
تعيد الرسالة مشفرة وفقاً لقواعد التشفير المعرفة في مواصفات TTCN-3.

`setEncodedMessage(byte[] message)` -
تدمث تمثل الرسالة المشفرة لهذه `TriMessage` على `message`.

`equals(TriMessage message)` -
تقارن `message` بهذه `TriMessage` للتأكد من التكافؤ. وتعيد `true` إذا، وإذا فقط للرسالتين نفس التمثيل المشفر، و`false` إذا كانت عكس ذلك.

6.2.3.6 TriAddressType

`TriAddressType` تقابل مع السطوح البينية التالية:

```
// TRI IDL TriAddressType
package org.etsi.ttcn.tri;
public interface TriAddress {
    public byte[] getEncodedAddress();
    public void setEncodedAddress(byte[] address);
    public boolean equals(TriAddress address);
}
```

1.6.2.3.6 الطرائق

`getEncodedAddress()` -
يعيد العنوان المشفر

`setEncodedAddress(byte[] address)` -
يدمث العنوان المشفر لهذا `TriAddress` على `address`.

`equals(TriAddress address)` -
يقارب `address` بالعنوان `TriAddress` للتأكد من التكافؤ ويعيد `true` إذا، وإذا فقط، كان للعنوانين نفس التمثيل المشفر و`false` إذا كانت غير ذلك.

TriAddressListType 7.2.3.6

TriAddressListType تقابل مع السطوح البينية التالية:

```
// TRI IDL TriAddressListType
package org.etsi.ttcn.tri;
public interface TriAddressListType {
    public int size();
    public boolean isEmpty();
    public java.util.Enumeration getAddresses();
    public TriAddress get(int index);
    public void clear();
    public void add(TriAddress addr);
}
```

الطرائق 1.7.2.3.6

`size()`

يعيد عدد المكونات في هذه القائمة.

`isEmpty()`

يعيد `true` إذا كانت هذه القائمة لا تحتوي على أية مكونات.

`getAddresses()`

يعيد `Enumeration` على المكونات في القائمة. ويقدم الترقيم العناوين بنفس الترتيب الذي ظهر به في القائمة.

`get(int index)`

يعيد `TriAddress` عند الوضع المحدد.

`clear()`

تحذف جميع العناوين من `TriAddressList`.

`add(TriAddress addr)`

تضيف `Addr` في نهاية هذه القائمة `TriAddressList`.

TriSignatureIdType 8.2.3.6

TriSignatureIdType تقابل مع السطوح البينية التالية:

```
// TRI IDL TriSignatureIdType
package org.etsi.ttcn.tri;
public interface TriSignatureId {
    public String getSignatureName();
    public void setSignatureName(String sigName);
    public boolean equals(TriSignatureId sig);
}
```

الطرائق 1.8.2.3.6

`getSignatureName()` -

يعيد معرف التوقيع على النحو المعرف في مواصفات TTCN-3.

`setSignatureName(String sigName)` -

تدمث معرف التوقيع في هذا `TriSignatureId` على `sigName`.

`equals(TriSignatureId sig)` -

يقارن التوقيع على `TriSignatureId` للتأكد من التكافؤ، يعيد `true` إذا، وإذا فقط، كان للتوقيعين نفس معرف التوقيع و`fals` إذا كان غير ذلك.

TriParameterType 9.2.3.6

TriParameterType تقابل مع السطوح البينية التالية:

```
// TRI IDL TriParameterType
package org.etsi.ttcn.tri;
public interface TriParameter {
    public String getParameterName();
    public void setParameterName(String name);
    public int getParameterPassingMode();
    public void setParameterPassingMode(in mode);
    public byte[] getEncodedParameter();
    public void setEncodedParameter(byte[] parameter);
}
```

1.9.2.3.6 الطرائق

- `getParameterName()`
تعيد اسم المعلمة على النحو المعرف في مواصفات TTCN-3.

- `setParameterName(String name)`
تدمث اسم هذه المعلمة `TriSignatureId` على `name`.

- `getParameterPassingMode()`
تعيد أسلوب تمرير المعلمة في هذه المعلمة.

- `setParameterPassingMode(in mode)`
تدمث أسلوب المعلمة في هذه المعلمة `TriParameter` على `mode`.

- `getEncodedParameter()`
تعيد تمثيل المعلمة المشفرة في `TriParameter` أو هدف `null` إذا كانت المعلمة تحتوي على القيمة المميزة `null` (انظر أيضاً 1.4.5.5).

- `setEncodedParameter(byte[] parameter)`
تدمث تمثيل المعلمة المشفرة في هذه `TriParameter` على `parameter`. وإذا كانت القيمة المميزة `null` فسوف تدمث لبيان أن هذه المعلمة لا تحتفظ بأي قيمة، وسوف تمرر جافا `null` كمعلمة `parameter` (انظر أيضاً 1.4.5.5).

TriParameterPassingModeType 10.2.6.6

TriParameterPassingModeType يقابل مع السطوح البينية التالية:

```
// TRI IDL TriParameterPassingModeType
package org.etsi.ttcn.tri;
public interface TriParameterPassingMode {
    public final static int TRI_IN = 0;
    public final static int TRI_INOUT = 1;
    public final static int TRI_OUT = 2;
}
```

1.10.2.3.6 الثوابت

- `TRI_IN`
سوف تستخدم لبيان أن `TriParameter` عبارة عن معلمة `in`.

- `TRI_INOUT`
سوف تستخدم لبيان أن `TriParameter` عبارة عن معلمة `inout`.

- `TRI_OUT`
سوف تستخدم لبيان أن `TriParameter` عبارة عن معلمة `out`.

TriParameterListType 11.2.3.6

TriParameterListType تقابل مع السطوح البينية التالية:

```
// TRI IDL TriParameterListType
package org.etsi.ttcn.tri;
public interface TriParameterList {
    public int size();
    public boolean isEmpty();
    public java.util.Enumeration getParameters();
    public TriParameter get(int index);
    public void clear();
    public void add(TriParameter parameter);
}
```

1.11.2.3.6 الطرائق

- `size()`
تعيد عدد المعلمات في هذه القائمة.

- `isEmpty()`
تعيد `true` إذا كانت القائمة لا تحتوي على معلمات.

getParameters()
تعيد Enumeration على المعلمات في القائمة. ويقدم الترقيم المعلمات بنفس الترتيب الذي تظهر به في القائمة.

get(int index)
تعيد TriParameter عند الوضع المحدد.

clear()
تحذف جميع المعلمات من هذه القائمة TriParameterList.

add(TriParameter parameter)
تضاف parameter في نهاية هذه القائمة TriParameterList.

TriExceptionType 12.2.3.6

TriExceptionType تقابل مع السطوح البينية التالية:

```
// TRI IDL TriExceptionType
package org.etsi.ttcn.tri;
public interface TriException {
    public byte[] getEncodedException();
    public void setEncodedException(byte[] message);
    public boolean equals(TriException exc);
}
```

الطرائق 1.12.2.3.6

getEncodedException() -
تعيد الاستثناء المشفر وفقاً لقواعد التشفير المحددة في مواصفات TTCN-3.

setEncodedMessage(byte[] exc) -
تدمث تمثيل الاستثناء المشفر من هذا TriException على exc.

equals(TriException exc) -
تقارن exc بهذا الكيان TriException للتأكد من التكافؤ وتعيد true إذا، وإذا فقط، كان لكلا الاستثناءين نفس التمثيل المشفر وfals إذا كانا غير ذلك.

TriTimerIdType 13.2.3.6

TriTimerIdType تقابل مع السطح بيني التالي:

```
// TRI IDL TriTimerIdType
package org.etsi.ttcn.tri;
public interface TriTimerId {
    public String getTimerName();
    public boolean equals(TriTimerId timer);
}
```

الطرائق 1.13.2.3.6

getTimerName() -
يقيد اسم معرف المؤقت هذا على النحو المعرف في مواصفات TTCN-3. وفي حالة المؤقتات الضمنية، تعتمد النتيجة على التنفيذ (انظر 2.1.4).

equals(TriTimerId timer) -
يقارن المؤقت timer بهذا المعرف TriTimerId للتأكد من التكافؤ – ويعيد true إذا، وإذا فقط، كان كلا معرفي المؤقتين يمثلان نفس المؤقت، وfals إذا كان الأمر غير ذلك.

TriTimerDurationType 14.2.3.6

TriTimerDurationType تتقابل مع السطح بيني الآتي:

```
// TRI IDL TriTimerDurationType
package org.etsi.ttcn.tri;
public interface TriTimerDuration {
    public double getDuration();
    public void setDuration(double duration);
    public boolean equals(TriTimerDuration duration);
}
```

1.14.2.3.6 الطرائق

getDuration() -
يعيد مدة المؤقت في شكل مزدوج.

setDuration(double duration) -
يدمث مدة هذا المعرف TriTimerDuration على duration.

equals(TriTimerDuration duration) -
يقارن duration بهذا المعرف TriTimerDuration للتأكد من التكافؤ ويعيد true إذا، وإذا فقط، كان للاثنين نفس المدة وfals إذا كان الأمر غير ذلك.

15.2.3.6 TriFunctionIdType

TriFunctionIdType تتقابل مع السطح بيني الآتي:

```
// TRI IDL TriFunctionIdType
package org.etsi.ttcn.tri;
public interface TriFunctionId {
    public String toString();
    public String getFunctionName();
    public boolean equals(TriFunctionId fun);
}
```

1.15.2.3.6 الطرائق

toString() -
يعيد تمثيل السلسلة للوظيفة على النحو المعرف في مواصفات TTCN-3.

getFunctionName() -
يعيد معرف الوظيفة على النحو المعرف في مواصفات TTCN-3.

equals(TriFunctionId fun) -
يقارن fun بالمعرف TriFunctionId للتأكد من التكافؤ ويعيد true إذا، وإذا فقط، كان لكلتا الوظيفتين نفس معرف الوظائف، وfals إذا كان الأمر غير ذلك.

16.2.3.6 riTestCaseIdType

TriTestCaseIdType تتقابل مع السطح بيني التالي:

```
// TRI IDL TriTestCaseIdType
package org.etsi.ttcn.tri;
public interface TriTestCaseId {
    public String toString();
    public String getTestCaseName();
    public boolean equals(TriTestCaseId tc);
}
```

1.16.2.3.6 الطرائق

toString() -
يعيد تمثيل السلسلة لحالة الاختبار على النحو المعرف في مواصفات TTCN-3.

getTestCaseName()-
يعيد معرف حالة الاختبار على النحو المعرف في مواصفات TTCN-3.

equals(TriTestCaseId tc) -
يقارن tc بهذا المعرف TriTestCaseId للتأكد من التكافؤ. ويعيد true إذا، وإذا فقط، كان لكلتا حالتي الاختبار نفس المعرف وfals إذا كان الأمر غير ذلك.

17.2.3.6 TriActionTemplateType

تقادمت.

TriStatusType 18.2.3.6

TriStatusType تتقابل مع السطح بيني الآتي:

```
// TriStatusType
package org.etsi.ttcn.tri;
public interface TriStatus {
    public final static int TRI_OK = 0;
    public final static int TRI_ERROR = -1;
    public String toString();
    public int getStatus();
    public void setStatus(int status);
    public boolean equals(TriStatus status);
}
```

1.18.2.3.6 الطرائق

toString() -
تعيد تمثيل السلسلة للحالة.

getStatus() -
تعيد حالة هذا المعرف TriStatus.

setStatus(int status) -
تدمث حالة هذا المعرف TriStatus.

equals(TriStatus status) -
تقارن status مع هذا المعرف TriStatus للتأكد من التكافؤ وتقيد true إذا، وإذا فقط، كان لهما نفس الحالة وfals إذا كان الأمر غير ذلك.

4.6 الثوابت

جرى في لغة جافا تحديد تقابل أنواع الثوابت. وتعرف جميع هذه الأنواع في public final static ويمكن النفاذ إليها من كل غرض في كل حزمة. وأنواع الثوابت المعرفة بالبند IDL. وبدلاً من ذلك، فإنها تنشأ من مواصفات أنماط TRI IDL المشار إليها بالمحلية.

ويمكن استحداث أنماط الثوابت التالية لتحديد أسلوب المرور لمعلمات TTCN-3 (انظر أيضاً 10.2.3.6).

- org.etsi.ttcn.tri.TriParameterPassingMode.TRI_IN؛
- org.etsi.ttcn.tri.TriParameterPassingMode.TRI_INOUT؛
- etsi.ttcn.tri.TriParameterPassingMode.TRI_OUT.

وسوف تعكس قيم حالات أنماط الثبات هذه أسلوب مرور المعلمات المعرف في توقيعات إجراء TTCN-3

وبالنسبة لقيمة المعلمة المميزة null، تدمث قيمة المعلمة المشفرة على null في جافا.

وسوف تستخدم أنماط الثبات التالية لبيان النفاذ المحلي للطريقة (انظر أيضاً 18.2.3.6).

- org.etsi.ttcn.tri.TriStatus.TRI_OK؛
- org.etsi.ttcn.tri.TriStatus.TRI_ERROR.

5.6 تقابل السطوح البينية

يعرف TRI IDL السطحين البيانين **triCommunication** و**triPlatform** ونظراً لأن العمليات معرفة لاتجاهات مختلفة في هذا السطح بيني أي أنه لا يمكن طلب بعض العمليات إلاّ بواسطة الجوانب القابلة للتنفيذ في TTCN-3 على مكيف النظام (SA) في حين لا يمكن طلب البعض الآخر إلاّ بواسطة (SA) على (TE). ويظهر ذلك من تقسيم السطوح البينية في TRI IDL إلى سطحين بينين فرعيين كلاهما يلحق بالكيان المنادى.

الجدول Z.144/4 – السطوح البينية الفرعية

الكيان طالب النداء والكيان لمنادى	TE	معرف النظام SA	معرف المنتدى PA
TE	-	TriCommunicationSA	triPlatformPA
SA	TriCommunicationTE	-	-
PA	TriPlatformTE	-	-

تعامل كل الطرائق المعرفة في هذه السطوح البينية كما عُرفت في الفقرة 5.

1.5.6 أسلوب مرور معلمات خارج وداخل خارج

تستخدم أنماط IDL التالية في أسلوب مرور معلمات out أو inout:

- معلمات TriParameter.
- قائمة معلمات TriParameterList.
- بولاين TriBoolean.
- مدة مؤقت TriTimerDuration.

وفي حالة استخدامها في أسلوب مرور معلمات out أو inout ستمر أهداف كل فئة منها مع نداء الطريقة، ويمكن للكيان المنادي أن ينفذ إلى الطرائق لتدميث قيم العودة.

2.5.6 triCommunication – السطح بيني

يتقسم السطح بيني **triCommunication** إلى سطحي، بينيين فرعيين هي السطح بيني triCommunicationSA الذي يعرف النداءات من TE إلى SA، والسطح بيني triCommunicationTE الذي يعرف النداءات من SA إلى TE

1.2.5.6 triCommunicationSA

يتقابل السطح بيني triCommunicationSA مع السطح بيني التالي:

```
// TriCommunication
// TE -> SA
package org.etsi.ttcn.tri;
public interface TriCommunicationSA {
    // Reset Operation
    // Ref: TRI-Definition 5.5.1
    TriStatus triSAReset();

    // Connection handling operations
    // Ref: TRI-Definition 5.5.2.1
    public TriStatus triExecuteTestCase(TriTestCaseId
        testCaseId,TriPortIdList tsiPorts);
    // Ref: TRI-Definition 5.5.2.2
    public TriStatus triMap(TriPortId compPortId, TriPortId tsiPortId);
    // Ref: TRI-Definition 5.5.2.3
    public TriStatus triUnmap(TriPortId compPortId, TriPortId tsiPortId);

    // Message based communication operations
    // Ref: TRI-Definition 5.5.3.1
    public TriStatus triSend(TriComponentId componentId, TriPortId tsiPortId,
            TriAddress sutAddress, TriMessage sendMessage);
    // Ref: TRI-Definition 5.5.3.2
    public TriStatus triSendBC(TriComponentId componentId, TriPortId tsiPortId,
            TriMessage sendMessage);
    // Ref: TRI-Definition 5.5.3.3
    public TriStatus triSendMC(TriComponentId componentId, TriPortId tsiPortId,
            TriAddressList addresses, TriMessage sendMessage);

    // Procedure based communication operations
    // Ref: TRI-Definition 5.5.4.1
    public TriStatus triCall(TriComponentId componentId,
            TriPortId tsiPortId, TriAddress sutAddress,
            TriSignatureId signatureId, TriParameterList parameterList);
    // Ref: TRI-Definition 5.5.4.2
```

```
    public TriStatus triCallBC(TriComponentId componentId,
            TriPortId tsiPortId,
            TriSignatureId signatureId, TriParameterList parameterList);
    // Ref: TRI-Definition 5.5.4.3
    public TriStatus triCallMC(TriComponentId componentId,
            TriPortId tsiPortId, TriAddressList sutAddresses,
            TriSignatureId signatureId, TriParameterList parameterList);

    // Ref: TRI-Definition 5.5.4.4
    public TriStatus triReply(TriComponentId componentId,
            TriPortId tsiPortId, TriAddress sutAddress,
            TriSignatureId signatureId, TriParameterList parameterList,
            TriParameter returnValue);
    // Ref: TRI-Definition 5.5.4.5
    public TriStatus triReplyBC(TriComponentId componentId,
            TriPortId tsiPortId,
            TriSignatureId signatureId, TriParameterList parameterList,
            TriParameter returnValue);
    // Ref: TRI-Definition 5.5.4.6
    public TriStatus triReplyMC(TriComponentId componentId,
            TriPortId tsiPortId, TriAddressList sutAddresses,
            TriSignatureId signatureId, TriParameterList parameterList,
            TriParameter returnValue);

    // Ref: TRI-Definition 5.5.4.7
    public TriStatus triRaise(TriComponentId componentId, TriPortId tsiPortId,
            TriAddress sutAddress,
            TriSignatureId signatureId,
            TriException exc);
    // Ref: TRI-Definition 5.5.4.8
    public TriStatus triRaiseBC(TriComponentId componentId, TriPortId tsiPortId,
            TriSignatureId signatureId,
            TriException exc);
    // Ref: TRI-Definition 5.5.4.9
    public TriStatus triRaiseMC(TriComponentId componentId,TriPortId tsiPortId,
            TriAddresses sutAddresses,
            TriSignatureId signatureId,
            TriException exc);

    // Miscellaneous operations
    // Ref: TRI-Definition 5.5.5.1
    public TriStatus triSutActionInformal(String description);

}
```

triCommunicationTE 2.2.5.6

يتقابل السطح بيني `triCommunicationTE` مع السطح بيني التالي:

```
// TriCommunication
// SA -> TE
package org.etsi.ttcn.tri;
public interface TriCommunicationTE {
    // Message based communication operations
    // Ref: TRI-Definition 5.5.3.4
    public void triEnqueueMsg(TriPortId tsiPortId,
            TriAddress sutAddress, TriComponentId componentId,
            TriMessage receivedMessage);

    // Procedure based communication operations
    // Ref: TRI-Definition 5.5.4.10
    public void triEnqueueCall(TriPortId tsiPortId,
            TriAddress SUTaddress, TriComponentId componentId,
            TriSignatureId signatureId, TriParameterList parameterList );

    // Ref: TRI-Definition 5.5.4.11
    public void triEnqueueReply(TriPortId tsiPortId, TriAddress sutAddress,
            TriComponentId componentId, TriSignatureId signatureId,
            TriParameterList parameterList, TriParameter returnValue);

    // Ref: TRI-Definition 5.5.4.12
    public void triEnqueueException(TriPortId tsiPortId,
            TriAddress sutAddress, TriComponentId componentId,
        TriSignatureId signatureId, TriException exc);
}
```

3.5.6 منتدى triPlatform – السطح بيني

ينقسم السطح بيني triPlatform إلى سطحي، بينيين فرعيين، السطح بيني triPlatformPA الذي يعرف النداءات من TE إلى PA، والسطح بيني triPlatformTE الذي يعرف النداءات من PA إلى TE

1.3.5.6 triPlatformPA

يتقابل السطح بيني triPlatformPA مع السطح بيني التالي:

```
// TriPlatform
// TE -> PA
package org.etsi.ttcn.tri;
public interface TriPlatformPA {
    // Ref: TRI-Definition 5.6.1
    public TriStatus triPAReset();

    // Timer handling operations
    // Ref: TRI-Definition 5.6.2.1
    public TriStatus triStartTimer(TriTimerId timerId,
        TriTimerDuration timerDuration);

    // Ref: TRI-Definition 5.6.2.2
    public TriStatus triStopTimer(TriTimerId timerId);

    // Ref: TRI-Definition 5.6.2.3
    public TriStatus triReadTimer(TriTimerId timerId,
        TriTimerDuration elapsedTime);

    // Ref: TRI-Definition 5.6.2.4
    public TriStatus triTimerRunning(TriTimerId timerId,
        TriBoolean running);

    // Miscellaneous operations

    // Ref: TRI-Definition 5.6.3.1
    public TriStatus triExternalFunction(TriFunctionId functionId,
     TriParameterList parameterList, TriParameter returnValue);
}
```

2.3.5.6 triPlatformTE

يتقابل السطح بيني triPlatformTE مع السطح بيني التالي لجافا:

```
// TriPlatform
// PA -> TE
package org.etsi.ttcn.tri;
public interface TriPlatformTE {
    // Ref: TRI-Definition 5.6.2.5
    public void triTimeout(TriTimerId timerId);
}
```

6.6 معلمات اختيارية

يعرف البند 4.5 بأن القيمة المحتجزة سوف تستخدم في بيان غياب معلمة اختيارية. وبالنسبة لتقابل لغة جافا، سوف يستخدم null في جافا لبيان غياب قيمة اختيارية. فعلى سبيل المثال، إذا حذفت معلمة العنوان من عملية triSend، فإن تنفيذ العملية سوف يكون triSend (componentId, tsiPortId, null, sendMessage).

7.6 تدميث السطح بيني لوقت تسيير TTCN-3

جميع الطرائق غير ثابتة أي أنه لا يمكن طلب العمليات إلاّ على أهداف. ونظراً لأن هذه التوصية لاتعرف استراتيجيات التنفيذ في TE وSA وPA، فإن الآلية التي يمكن أن تعرف بها TE وSA وPA، كيفية مناولة الأهداف المعنية تقع خارج نطاق هذه التوصية.

وسوف يوفر موردو الأدوات الطرائق لواضعي SA وPA لتسجيل وSA وPA على شريك التنفيذ المعني بكل منها.

8.6 مناولة الأخطاء

باستثناء مناولة الأخطاء الواردة في 2.5، لا يتوافر تعريف لأية رسائل إضافية لمناولة الأخطاء في تقابل لغة جافا المشار إليه. وعلى وجه الخصوص، لا يرد أي تعريف لآليات مناولة الاستثناء.

7 تقابل لغة ANSI-C

1.7 مقدمة

يعرف هذا البند تقابل لغة TRI ANSI-C لأنماط البيانات المجردة المحددة في 3.5. وبالنسبة لأنماط IDL الأساسية، يتواءم التقابل مع توصيات فريق إدارة الأهداف.

2.7 الأسماء والنطاقات

تبدأ معرفات معلمات C بحروف صغيرة وتبدأ الأجزاء التالية التي تكوّن معرف المعلمات بحروف كبيرة. فهي على سبيل المثال نمط `sutAddress` يتقابل مع `sutAddress` في C.

وتحذف معرفات نمط البيانات المجردة في C التخالف `Type` المستخدم في تعريف IDL. فعلى سبيل المثال، فإن النمط IDL `TriPortIdType` يتقابل مع `TriPortId` في C.

وكانت مواصفات C القديمة قد قصرت الطابع الفريد للمعرف على أهم 8 واسمات. ومع ذلك فإن مواصفات ANSI-C الحديثة نقلت هذا التحديد إلى أهم 31 واسمة. وبغض النظر عن هذه المسألة، لم يحدد أي تعريف للتضارب في التسميات أو النطاق في هذا التقابل.

1.2.7 تقابل النمط المجرد

TRI ADT	تمثيل ANSI-C	ملاحظات وتعليقات
`TriAddress`	`BinaryString`	

`TriAddressList`	`typedef struct TriAddressList`
	`{`
	` TriAddress** addrList;`
	` long int length;`
	`} TriAddressList;`
		ملاحظة: لاتوجد قيم خاصة تشير إلى نهاية `addrList[]`. وسوف يستخدم طول المجال لعبور هذا الصفيف بصورة سليمة.

`TriComponentId`	`typedef struct TriComponentId`
	`{`
	` BinaryString compInst;`
	` String compName;`
	` QualifiedName compType;`
	`} TriComponentId;`
		ملاحظة: `compInst` لحالة المكونات.

`TriComponentIdList`	`typedef struct TriComponentIdList`
	`{`
	` TriComponentId** compIdList;`
	` long int length;`
	`} TriComponentIdList;`
		ملاحظة: لاتوجد قيم خاصة تشير إلى نهاية `compIdList[]`. وسوف يستخدم طول المجال لعبور هذا الصفيف بصورة سليمة.

`TriException`	`BinaryString`	
`TriFunctionId`	`QualifiedName`	
`TriMessage`	`BinaryString`	

TRI ADT	تمثيل ANSI-C	ملاحظات وتعليقات
TriParameterList	typedef struct TriParameterList { TriParameter** parList; long int length; } TriParameterList;	**ملاحظة:** لاتوجد قيم خاصة تشير إلى نهاية parList. وسوف يستخدم طول المجال لعبور هذا الصفيف بصورة سليمة.
TriParameter	typedef struct TriParameter { BinaryString par; TriParameterPassingMode mode; } TriParameter;	
TriParameterPassingMode	typedef enum { TRI_IN = 0, TRI_INOUT = 1, TRI_OUT = 2 } TriParameterPassingMode;	ملاحظة: قيم حالات هذا النمط سوف تعكس أسلوب مرور المعلمات المعرف في توقيتات إجراءات TTCN-3 المقابلة.
TriPortIdList	typedef struct TriPortIdList { TriPortId** portIdList; long int length; } TriPortIdList;	الملاحظة: لاتوجد قيم خاصة تشير إلى نهاية portIdList[]. وسوف يستخدم طول المجال لعبور هذا الصفيف بصورة سليمة.
TriPortId	typedef struct TriPortId { TriComponentId compInst; char* portName; long int portIndex; QualifiedName portType; void* aux; } TriPortId;	**الملاحظة 1:** compInst– لحالة المكونات. **الملاحظة 2:** بالنسبة للإعلان المفرد (غير الصفيف)، ينبغي أن تكون قيمة portIndex يعادل 1 **الملاحظة 3:** مجال aux لتوسيع وظائف TRI في المستقبل.
TriSignatureId	QualifiedName	
TriStatus	long int #define TRI_ERROR -1 #define TRI_OK 0	**ملاحظة:** تحتجز جميع القيم السالبة لتوسيع وظائف TRI في المستقبل
TriTestCaseId	QualifiedName	
TriTimerDuration	Double	
TriTimerId	BinaryString	**ملاحظة:** في انتظار بيان بشأن المؤقت واللقطات، قد تؤثر الدلالات اللغوية على التمثيل في المستقبل.

2.2.7 تعاريف نمط ANSI-C

C ADT	تعريف النمط	ملاحظات وتعليقات
BinaryString	typedef struct BinaryString { unsigned char* data; long int bits; void* aux; } BinaryString;	**الملاحظة 1:** data عبارة عن سلسلة منتهية غير صفرية. **الملاحظة 2:** bits هو عدد البتات المستخدمة في بيانات قيمة البتات 1 تستخدم لبيان قيمة محذوفة. **الملاحظة 3:** المجال aux لتوسيع وظائف TRI في المستقبل.
QualifiedName	typedef struct QualifiedName { char* moduleName; char* objectName; void* aux; } QualifiedName;	**الملاحظة 1:** moduleName وobjectName هما معرفان للنسخة TTCN-3 حرفياً. **الملاحظة 2:** مجال aux لتوسيع وظائف TRI في المستقبل

3.2.7 تقابل نمط IDL

نمط IDL	تمثيل ANSI-C	ملاحظات وتعليقات
Boolean	unsigned char	تقابل من OMG IDL إلى C++ mapping
String	char*	تقابل من OMG IDL إلى C++ mapping

4.2.7 تقابل عملية TRI

تمثيل IDL	تمثيل ANSI-C
TriStatusType triSAReset()	TriStatus triSAReset()
TriStatusType triExecuteTestCase (in TriTestCaseIdType testCaseId, in TriPortIdListType tsiPortList)	TriStatus triExecuteTestCase (const TriTestCaseId* testCaseId, const TriPortIdList* tsiPortList)
TriStatusType triMap (in TriPortIdType compPortId, in TriPortIdType tsiPortId)	TriStatus triMap (const TriPortId* compPortId, const TriPortId* tsiPortId)
TriStatusType triUnmap (in TriPortIdType compPortId, in TriPortIdType tsiPortId)	TriStatus triUnmap (const TriPortId* compPortId, const TriPortId* tsiPortId)
TriStatusType triSend (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriMessageType sendMessage)	TriStatus triSend (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriAddress* sutAddress, const TriMessage* sendMessage)
TriStatusType triSendBC (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriMessageType sendMessage)	TriStatus triSendBC (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriMessage* sendMessage)

تمثيل ANSI-C	تمثيل IDL
TriStatus triSendMC (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriAddressList* sutAddresses, const TriMessage* sendMessage)	TriStatusType triSendMC (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressListType SUTaddresses, in TriMessageType sendMessage)
void triEnqueueMsg (const TriPortId* tsiPortId, const TriAddress* sutAddress, const TriComponentId* componentId, const TriMessage* receivedMessage)	void triEnqueueMsg (in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriComponentIdType componentId, in TriMessageType receivedMessage)
TriStatus triCall (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriAddress* sutAddress, const TriSignatureId* signatureId, const TriParameterList* parameterList)	TriStatusType triCall (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriSignatureIdType signatureId, in TriParameterListType parameterList)
TriStatus triCallBC (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriSignatureId* signatureId, const TriParameterList* parameterList)	TriStatusType triCallBC (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriSignatureIdType signatureId, in TriParameterListType parameterList)
TriStatus triCallMC (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriAddressList* sutAddresses, const TriSignatureId* signatureId, const TriParameterList* parameterList)	TriStatusType triCallMC (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressListType SUTaddresses, in TriSignatureIdType signatureId, in TriParameterListType parameterList)
TriStatus triReply (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriAddress* sutAddress, const TriSignatureId* signatureId, const TriParameterList* parameterList, const TriParameter* returnValue)	TriStatusType triReply (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriSignatureIdType signatureId, in TriParameterListType parameterList, in TriParameterType returnValue)
TriStatus triReplyBC (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriSignatureId* signatureId, const TriParameterList* parameterList, const TriParameter* returnValue)	TriStatusType triReplyBC (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriSignatureIdType signatureId, in TriParameterListType parameterList, in TriParameterType returnValue)

تمثيل ANSI-C	تمثيل IDL
TriStatus triReplyMC (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriAddressList* sutAddresses, const TriSignatureId* signatureId, const TriParameterList* parameterList, const TriParameter* returnValue)	TriStatusType triReplyMC (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressListType SUTaddresses, in TriSignatureIdType signatureId, in TriParameterListType parameterList, in TriParameterType returnValue)
TriStatus triRaise (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriAddress* sutAddress, const TriSignatureId* signatureId, const TriException* exception)	TriStatusType triRaise (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriSignatureIdType signatureId, in TriExceptionType exc)
TriStatus triRaiseBC (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriSignatureId* signatureId, const TriException* exception)	TriStatusType triRaiseBC (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriSignatureIdType signatureId, in TriExceptionType exc)
TriStatus triRaiseMC (const TriComponentId* componentId, const TriPortId* tsiPortId, const TriAddressList* sutAddresses, const TriSignatureId* signatureId, const TriException* exception)	TriStatusType triRaiseMC (in TriComponentIdType componentId, in TriPortIdType tsiPortId, in TriAddressListType SUTaddresses, in TriSignatureIdType signatureId, in TriExceptionType exc)
void triEnqueueCall (const TriPortId* tsiPortId, const TriAddress* sutAddress, const TriComponentId* componentId, const TriSignatureId* signatureId, const TriParameterList* parameterList)	void triEnqueueCall (in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriComponentId componentId, in TriSignatureIdType signatureId, in TriParameterListType parameterList)
void triEnqueueReply (const TriPortId* tsiPortId, const TriAddress* sutAddress, const TriComponentId* componentId, const TriSignatureId* signatureId, const TriParameterList* parameterList, const TriParameter* returnValue)	void triEnqueueReply (in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriComponentIdType componentId, in TriSignatureIdType signatureId, in TriParameterListType parameterList, in TriParameterType returnValue)

تمثيل IDL	تمثيل ANSI-C
void triEnqueueException (in TriPortIdType tsiPortId, in TriAddressType SUTaddress, in TriComponentIdType componentId, in TriSignatureIdType signatureId, in TriExceptionType exc)	void triEnqueueException (const TriPortId* tsiPortId, const TriAddress* sutAddress, const TriComponentId* componentId, const TriSignatureId* signatureId, const TriException* exception)
TriStatusType triSUTActionInformal (in string description)	TriStatus triSUTActionInformal (const char* description)
TriStatusType triPAReset()	TriStatus triPAReset()
TriStatusType triStartTimer (in TriTimerIdType timerId, in TriTimerDurationType timerDuration)	TriStatus triStartTimer (const TriTimerId* timerId, TriTimerDuration timerDuration)
TriStatusType triStopTimer (in TriTimerIdType timerId)	TriStatus triStopTimer (const TriTimerId* timerId)
TriStatusType triReadTimer (in TriTimerIdType timerId, out TriTimerDurationType elapsedTime)	TriStatus triReadTimer (const TriTimerId* timerId, TriTimerDuration* elapsedTime)
TriStatusType triTimerRunning (in TriTimerIdType timerId, out boolean running)	TriStatus triTimerRunning (const TriTimerId* timerId, unsigned char* running)
void triTimeout (in TriTimerIdType timerId)	void triTimeout (const TriTimerId* timerId)
TriStatusType triExternalFunction (in TriFunctionIdType functionId, inout TriParameterListType parameterList, out TriParameterType returnValue)	TriStatus triExternalFunction (const TriFunctionId* functionId, TriParameterList* parameterList, TriParameter* returnValue)

3.7 إدارة الذاكرة

تقادمت.

4.7 مناولة الأخطاء

لم تعرف أية مناولة للأخطاء لهذا التقابل.

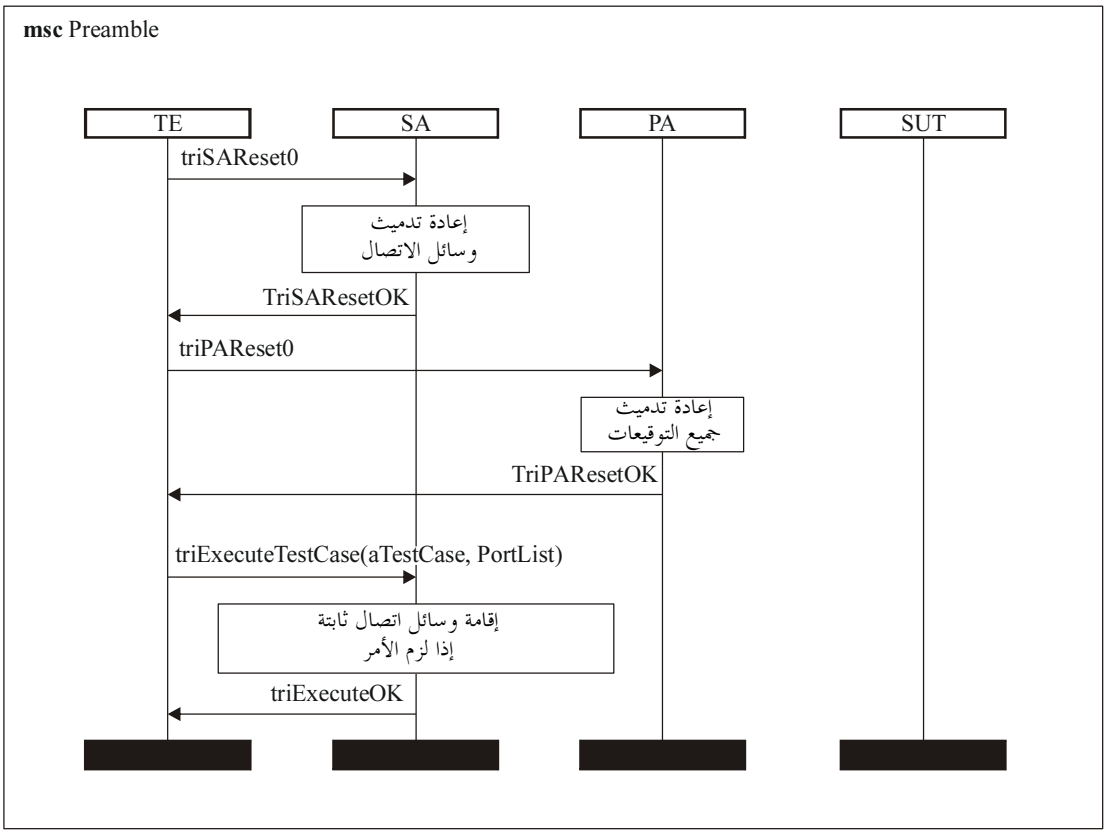
8 سيناريوهات الاستخدام

يحتوي هذا البند على سيناريوهات الاستخدام التي ينبغي أن تساعد مستعملي TRI وموردي الأدوات بشرط إدراك TRI للتركيبات اللغوية للعمليات المحددة في هذه التوصية.

ويجرى تعريف ثلاثة سيناريوهات من حيث مخططات تتابع الرسائل (MSC). ويتألف السيناريو ألف من الجزء من شفرة TTCN-3 التي تستخدم وظائف اتصالات TTCN-3 إلى النظام قيد الاختبار SUT فضلاً عن وظائف مناولة المؤقت. وتبين SUT التفاعلات بين الكيانات TE وSA وPA، وبينها وبين SUT.

ويرجى الملاحظة أن أجزاء TTCN-3 ليست كاملة حيث أن الهدف الرئيسي للأجزاء هو استخدام السلوك الديناميي. وجميع السيناريوهات المقدمة تستخدم تتابع استهلالي مشترك لعمليات TRI وترد في الشكل 2.

ويلاحظ إن MSCs المقدمة في هذا البند تستخدم أزواجاً من الرسائل لتشكيل كل عملية TRI وتبين رسالة MSC وهي triMap التي تعقبها triMapOK، مثلاً، أن triMap في عملية TRI قد نفذها TE وعادت بنجاح من SA. وتبين نداءات عملية TRI باستخدام أنماط وقيم مجردة، ويتعرض أن تستخدم لأغراض التوضيح فقط. ويعرف التمثيل العملي لهذه المعلمات في لغة مستهدفة معينة في التقابلات اللغوية الخاصة بكل منها.



الشكل Z.144/2 – مستهل MSC المشترك

1.8 السيناريو الأول

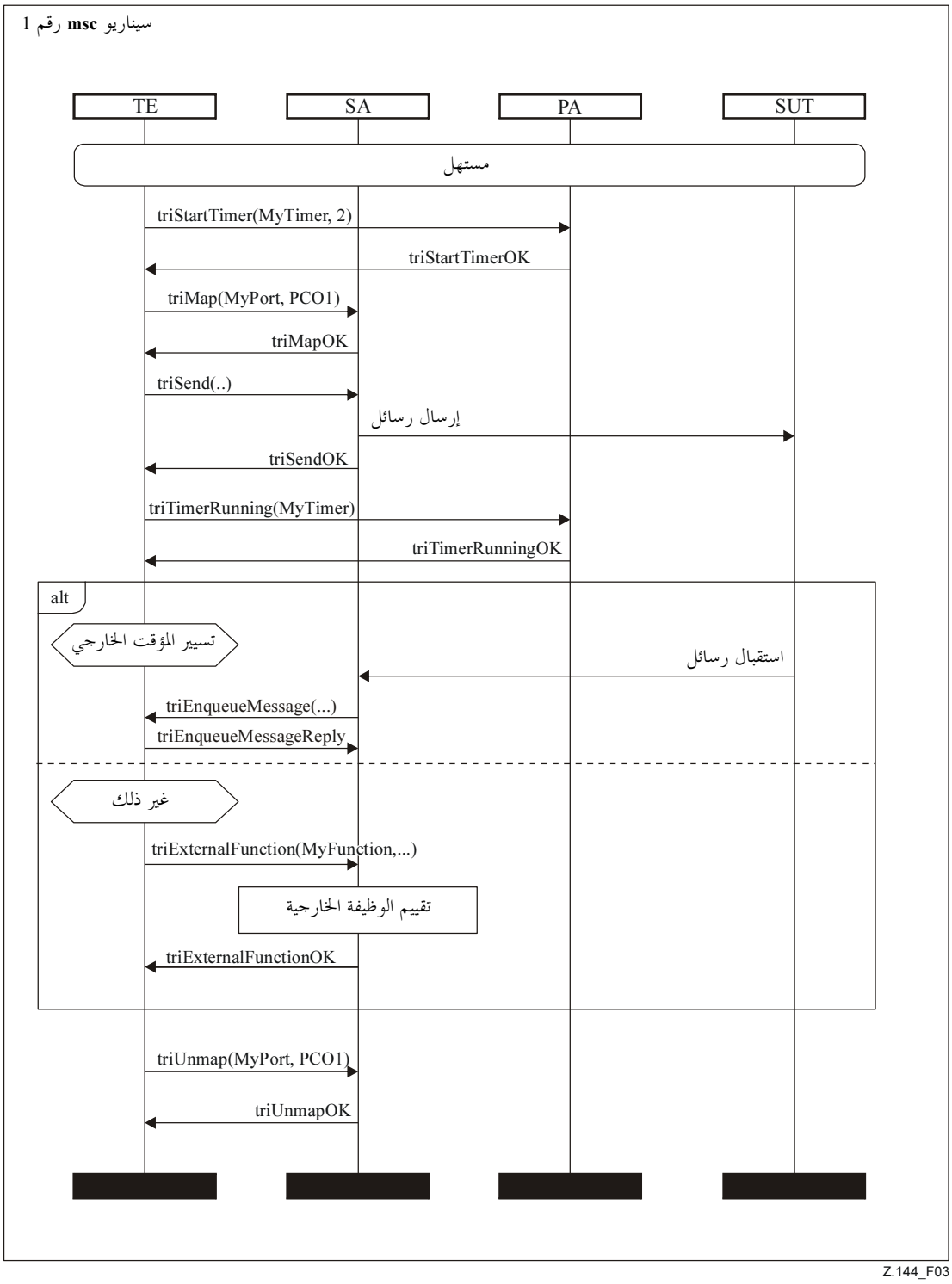
يبين السيناريو الأول بعض العمليات الداخلية في TTCN-3، أي بدء وتوقيت التسيير وعمليات الاتصالات المعتمدة على الرسائل أي الإرسال والاستقبال فضلاً عن عمليات مناولة الربط أي التقابل وعدم التقابل.

1.1.8 جزء من TTCN-3

```
module triScenario1
{
 external function MyFunction();

 type port PortTypeMsg message { inout integer }

 type component MyComponent {
    port PortTypeMsg MyPort;
    timer MyTimer
 }

 type component MyTSI {
    port PortTypeMsg PCO1;
 }

 testcase scenario1() runs on MyComponent system MyTSI
```

```
 {
    MyPort.clear;
    MyPort.start;
    MyTimer.start(2);

    map(MyComponent: MyPort, system: PCO1);
    MyPort.send (integer : 5);
    if (MyTimer.running)
    {
     MyPort.receive(integer:7);
    }
    else
    {
     MyFunction();
    }
    unmap(MyComponent: MyPort, system:PCO1);
    MyPort.stop;
 }


 control {
    execute( scenario1() );
 }

}
```

2.1.8 مخطط تتابع الرسائل



الشكل Z.144/3 – سيناريو الاستخدام 1

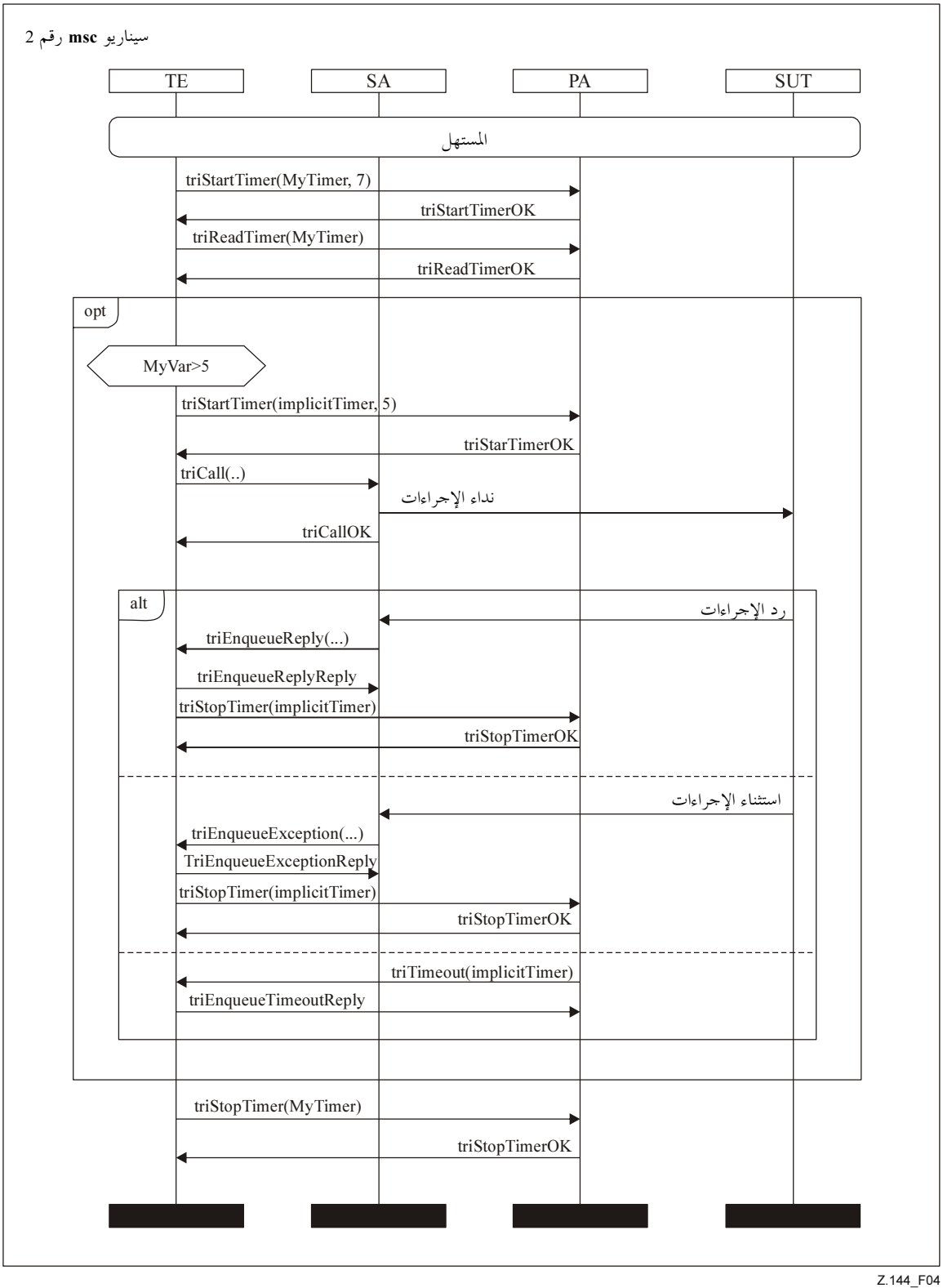
2.8 السيناريو الثاني

يبين المثال الثاني سيناريو مماثلاً يستخدم أيضاً عمليات الاتصالات المعتمدة على الإجراءات المؤقتة التي يتم تدميثها بواسطة مكون الاختبار MyComponent. وفي هذا المثال يفترض أن MyComponent تسير مثل MTC.

1.2.8 جزء من TTCN-3

```
module triScenario2
{

 signature MyProc ( in float par1, inout float par2)
    exception(MyExceptionType);

 type record MyExceptionType { FieldType1 par1, FieldType2 par2 }

 type port PortTypeProc procedure { out MyProc }

 type component MyComponent {
    port PortTypeProc MyPort;
    timer MyTimer = 7
 }

 testcase scenario2() runs on MyComponent
 {
    var float MyVar;

    MyPort.clear;
    MyPort.start;
    MyTimer.start;

 MyVar := MyTimer.read;

 if (MyVar>5.0) {
     MyPort.call (MyProc:{MyVar, 5.7}, 5);
     alt {
     [] MyPort.getreply(MyProc:{-,MyVar*5}) {}
     [] MyPort.catch (MyProc, MyExceptionType:* ) {}
     [] MyPort.catch (timeout) {}
     }
    }
MyTimer.stop;
    MyPort.stop;
 }

 control {
    execute( scenario2() );
 }

}
```

2.2.8 مخطط تتابع الرسائل



الشكل Z.144/4 – سيناريو الاستخدام 2

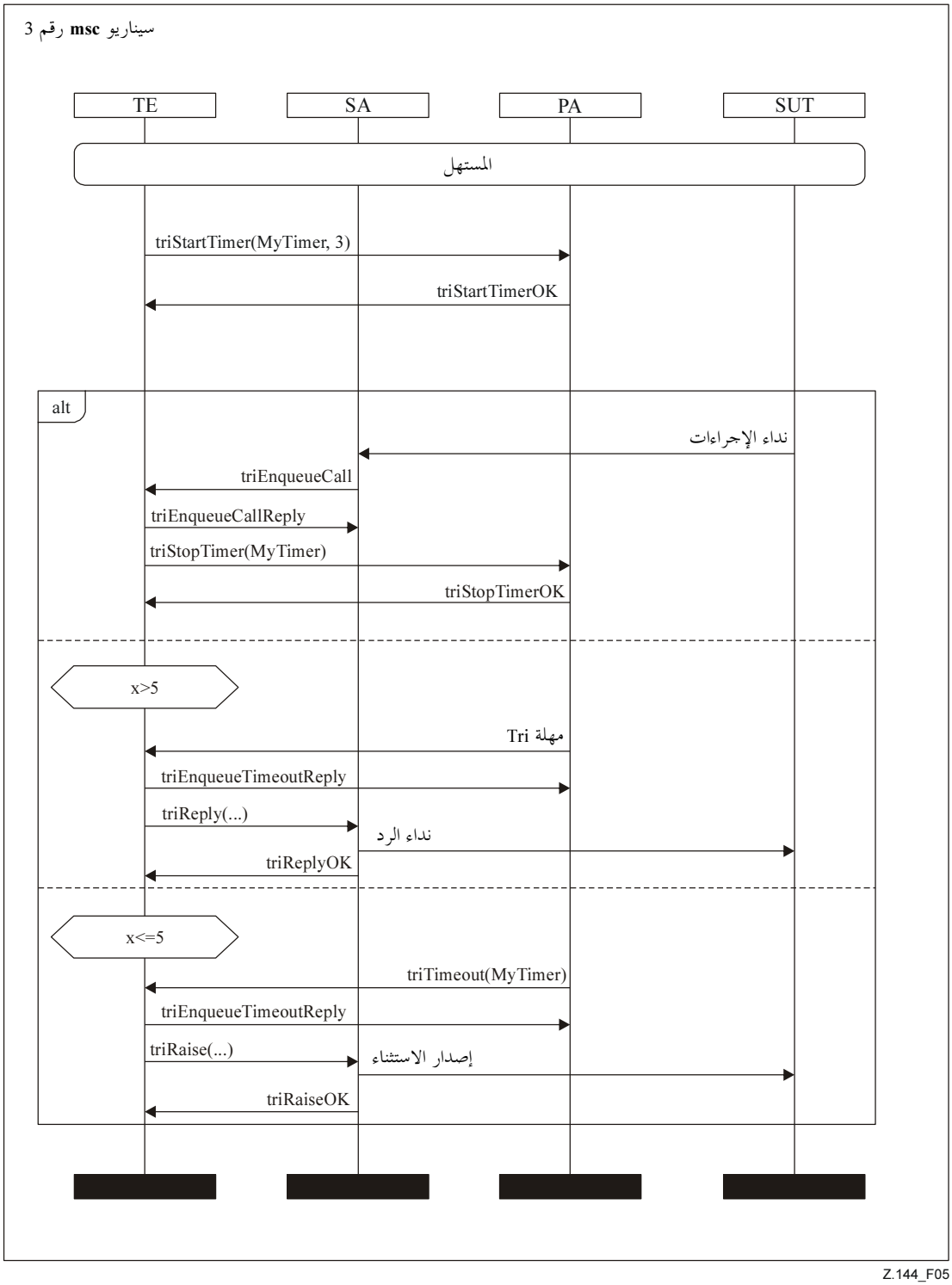
3.8 السيناريو الثالث

يبين سيناريو الاستخدام رقم 3 استقبال نداء الإجراءات فضلاً عن الرد وإصدار الاستثناء المعد على هذا النداء المستقبل. ومرة أخرى يفترض أن MyComponent تسير مثل MTC. ويفترض أن FieldType1، FieldType2، p1 وp2 سوف تعرّف في مكان آخر.

1.3.8 جزء من TTCN-3

```
module triScenario3
{
 signature MyProc ( in float par1, inout float par2)
    exception(MyExceptionType);

 type record MyExceptionType { FieldType1 par1, FieldType2 par2 }

 type port PortTypeProc procedure { in MyProc }

 type component MyComponent {
    port PortTypeProc MyPort;
    timer MyTimer = 3
 }

 testcase scenario3(integer x) runs on MyComponent
 {
    MyPort.start;
    MyTimer.start;
    alt
    {
     [] MyPort.getcall(MyProc:{5.0, 6.0})
     {
     MyTimer.stop;
     }
     [x>5] MyTimer.timeout
     {
     MyPort.reply(MyProc:{-, 30.0});
     }
     [x<=5] MyTimer.timeout
     {
     MyPort.raise(MyProc, MyExceptionType:{p1, p2} );
     }
    }
    MyPort.stop;
 }

 control {
    execute( scenario3(4) );
 }
}
```

2.3.8 مخطط تتابع الرسائل



الشكل Z.144/5 – سيناريو الاستخدام رقم 3

الملحق ألف (معياري)

موجز لغة تعريف السطوح البينية (IDL)

يوجز هذا المرفق تعريف IDL لعمليات TRI المعرفة في البند 5.

```
// ***************************************************************************
// Interface definition for the TTCN-3 Runtime Interface

// ***************************************************************************

module triInterface
{

 //
 // ***************************************************************************
 // Types
 // ***************************************************************************
 //

 // Connection
 native TriPortIdType;
 typedef sequence<TriPortIdType> TriPortIdListType;
 native TriComponentIdType;
 typedef sequence<TriComponentIdType> TriComponentIdListType;

 // Communication
 native TriMessageType;
 native TriAddressType;
 typedef sequence<TriAddressType> TriAddressListType;
 native TriSignatureIdType;
 native TriParameterType;
 typedef sequence<TriParameterType> TriParameterListType;
 native TriExceptionType;

 // Timing
 native TriTimerIdType;
 native TriTimerDurationType;

 // Miscellaneous
 native TriFunctionIdType;
 native TriTestCaseIdType;
 native TriStatusType;

 //
 // ***************************************************************************
 // Interfaces
 // ***************************************************************************
 //

 //
 // ***************************************************************************
 // The communication interface (Ref: TRI-Definition: 5.5)
 // ***************************************************************************
 //
 interface triCommunication
 {

    // Reset operation

    // Ref: TRI-Definition 5.5.1
    TriStatusType triSAReset();


    // Connection handling operations

    // Ref: TRI-Definition 5.5.2.1
    TriStatusType triExecuteTestCase(in TriTestCaseIdType testCaseId,
    in TriPortIdListType tsiPortList);

    // Ref: TRI-Definition 5.5.2.2
    TriStatusType triMap(in TriPortIdType compPortId, in TriPortIdType tsiPortId);
```

```
// Ref: TRI-Definition 5.5.2.3
TriStatusType triUnmap(in TriPortIdType compPortId, in TriPortIdType tsiPortId);

// Message based communication operations

// Ref: TRI-Definition 5.5.3.1
TriStatusType triSend(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriAddressType SUTaddress, in TriMessageType sendMessage);
// Ref: TRI-Definition 5.5.3.2
TriStatusType triSendBC(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriMessageType sendMessage);
// Ref: TRI-Definition 5.5.3.3
TriStatusType triSendMC(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriAddressListType SUTaddresses, in TriMessageType sendMessage);

// Ref: TRI-Definition 5.5.3.4
void triEnqueueMsg(in TriPortIdType tsiPortId , in TriAddressType SUTaddress,
 in TriComponentIdType componentId, in TriMessageType receivedMessage);


// Procedure based communication operations

// Ref: TRI-Definition 5.5.4.1
TriStatusType triCall(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriAddressType SUTaddress, in TriSignatureIdType signatureId,
in TriParameterListType parameterList);
// Ref: TRI-Definition 5.5.4.2
TriStatusType triCallBC(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriSignatureIdType signatureId,
in TriParameterListType parameterList);
// Ref: TRI-Definition 5.5.4.3
TriStatusType triCallMC(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriAddressListType SUTaddresses, in TriSignatureIdType signatureId,
in TriParameterListType parameterList);

// Ref: TRI-Definition 5.5.4.4
TriStatusType triReply(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriAddressType SUTaddress, in TriSignatureIdType signatureId,
in TriParameterListType parameterList, in TriParameterType returnValue );
// Ref: TRI-Definition 5.5.4.5
TriStatusType triReplyBC(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriSignatureIdType signatureId,
in TriParameterListType parameterList, in TriParameterType returnValue );
// Ref: TRI-Definition 5.5.4.6
TriStatusType triReplyMC(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriAddressListType SUTaddresses, in TriSignatureIdType signatureId,
in TriParameterListType parameterList, in TriParameterType returnValue );

// Ref: TRI-Definition 5.5.4.7
TriStatusType triRaise(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriAddressType SUTaddress, in TriSignatureIdType signatureId,
in TriExceptionType exc);
// Ref: TRI-Definition 5.5.4.8
TriStatusType triRaiseBC(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriSignatureIdType signatureId,
in TriExceptionType exc);
// Ref: TRI-Definition 5.5.4.9
TriStatusType triRaiseMC(in TriComponentIdType componentId, in TriPortIdType tsiPortId,
in TriAddressListType SUTaddresses, in TriSignatureIdType signatureId,
in TriExceptionType exc);

// Ref: TRI-Definition 5.5.4.10
void triEnqueueCall(in TriPortIdType tsiPortId, in TriAddressType SUTaddress,
in TriComponentIdType componentId, in TriSignatureIdType signatureId,
in TriParameterListType parameterList );

// Ref: TRI-Definition 5.5.4.11
void triEnqueueReply(in TriPortIdType tsiPortId, in TriAddressType SUTaddress,
in TriComponentIdType componentId, in TriSignatureIdType signatureId,
in TriParameterListType parameterList, in TriParameterType returnValue );

// Ref: TRI-Definition 5.5.4.12
void triEnqueueException(in TriPortIdType tsiPortId, in TriAddressType SUTaddress,
in TriComponentIdType componentId, in TriSignatureIdType signatureId,
in TriExceptionType exc);


// Miscellaneous operations
```

```
    // Ref: TRI-Definition 5.5.5.1
    TriStatusType triSUTactionInformal(in string description);

};
 //
 // *****************************************************************************
 // The platform interface (Ref: TRI-Definition: 5.6)
 // *****************************************************************************
 //
 interface triPlatform
 {

    // Reset Operation

    // Ref: TRI-Definition 5.6.1
    TriStatusType triPAReset();


    // Timer handling operations

    // Ref: TRI-Definition 5.6.2.1
    TriStatusType triStartTimer(in TriTimerIdType timerId,
    in TriTimerDurationType timerDuration);

    // Ref: TRI-Definition 5.6.2.2
    TriStatusType triStopTimer(in TriTimerIdType timerId);

    // Ref: TRI-Definition 5.6.2.3
    TriStatusType triReadTimer(in TriTimerIdType timerId,
    out TriTimerDurationType elapsedTime);

    // Ref: TRI-Definition 5.6.2.4
    TriStatusType triTimerRunning(in TriTimerIdType timerId, out boolean running);

    // Ref: TRI-Definition 5.6.2.5
    void triTimeout(in TriTimerIdType timerId);


    // Miscellaneous operations

    // Ref: TRI-Definition 5.6.3.1
    TriStatusType triExternalFunction(in TriFunctionIdType functionId,
     inout TriParameterListType parameterList,
     out TriParameterType returnValue);

 };
};
```

البيبليوغرافيا

– OMG CORBA (V2.2): *The Common Object Request Broker: Architecture and Specification*, Section 3, February 1998.

– INTOOL CGI/NPL038 (V2.2): *Generic Compiler/Interpreter interface*; GCI Interface Specification, Infrastructural Tools, December 1996.

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 البيانات على الشبكة الهاتفية
السلسلة X	شبكات البيانات 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، 2006