

Международный союз электросвязи

МСЭ-Т

СЕКТОР
СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

Z.145

(03/2006)

СЕРИЯ Z: ЯЗЫКИ И ОБЩИЕ АСПЕКТЫ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СИСТЕМ
ЭЛЕКТРОСВЯЗИ

Методы формального описания (FDT) – Нотация
тестирования и управления тестом (TTCN)

**Нотация тестирования и управления тестом
версии 3 (TTCN-3): Интерфейс управления
(TCI)**

Рекомендация МСЭ-Т Z.145

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ Z
ЯЗЫКИ И ОБЩИЕ АСПЕКТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ СИСТЕМ ЭЛЕКТРОСВЯЗИ

| | |
|--|--------------------|
| МЕТОДЫ ФОРМАЛЬНОГО ОПИСАНИЯ (FDT) | |
| Язык спецификации и описания (SDL) | Z.100–Z.109 |
| Применение методов формального описания | Z.110–Z.119 |
| Диаграмма последовательности сообщений (MSC) | Z.120–Z.129 |
| Расширенный язык описания объектов (eODL) | Z.130–Z.139 |
| Нотация тестирования и управления тестированием (TTCN) | Z.140–Z.149 |
| Нотация требований пользователя (URN) | Z.150–Z.159 |
| ЯЗЫКИ ПРОГРАММИРОВАНИЯ | |
| CHILL: язык высокого уровня МСЭ-Т | Z.200–Z.209 |
| ЯЗЫК "ЧЕЛОВЕК–МАШИНА" | |
| Общие принципы | Z.300–Z.309 |
| Базисный синтаксис и диалоговые процедуры | Z.310–Z.319 |
| Расширенный язык MML для видеотерминалов | Z.320–Z.329 |
| Спецификация интерфейса "человек–машина" | Z.330–Z.349 |
| Информационно-ориентированные интерфейсы "человек–машина" | Z.350–Z.359 |
| Интерфейсы "человек–машина" для управления сетями электросвязи | Z.360–Z.379 |
| КАЧЕСТВО | |
| Качество программного обеспечения электросвязи | Z.400–Z.409 |
| Аспекты качества рекомендаций, относящихся к протоколам | Z.450–Z.459 |
| МЕТОДЫ | |
| Методы проверки и тестирования | Z.500–Z.519 |
| ПРОМЕЖУТОЧНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ | |
| Среда распределенной обработки | Z.600–Z.609 |

Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.

**Нотация тестирования и управления тестом версии 3 (TTCN-3):
Интерфейс управления (TCI)**

Резюме

В настоящей Рекомендации приводится описание интерфейсов управления для реализаций тестовой системы согласно TTCN-3 (*Нотация тестирования и управления тестом версии 3*). Интерфейс управления согласно TTCN-3 обеспечивает рекомендуемую адаптацию администрирования, обработки тестового компонента и кодирования/декодирования тестовой системы к отдельной тестовой платформе. В настоящей Рекомендации интерфейсы определяются как набор операций, независимых от целевого языка.

Определения интерфейсов совместимы с положениями Рекомендации МСЭ-Т Z.140. Для полного определения интерфейса TCI в настоящей Рекомендации используется язык описания интерфейса (IDL) обобщенной архитектуры посредника объектных запросов (CORBA). В разделах 8 и 9 приводится описание отображений языка абстрактной спецификации на целевые языки Java и ANSI-C. В Приложении А содержится резюме, касающееся спецификации интерфейса на базе языка IDL.

Источник

Рекомендация МСЭ-Т Z.145 утверждена 16 марта 2006 года 17-й Исследовательской комиссией МСЭ-Т (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т А.8.

ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

На Всемирной ассамблее по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяются темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соблюдение положений данной Рекомендации носит добровольный характер. Однако в Рекомендации могут содержаться определенные обязательные положения (например, для обеспечения возможности взаимодействия или применимости), и соблюдение положений данной Рекомендации достигается в случае выполнения всех этих обязательных положений. Для выражения необходимости выполнения требований используется синтаксис долженствования и соответствующие слова (такие, как "должен" и т. п.), а также их отрицательные эквиваленты. Использование этих слов не предполагает, что соблюдение положений данной Рекомендации является обязательным для какой-либо из сторон.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или реализация этой Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, обоснованности или применимости заявленных прав интеллектуальной собственности, независимо от того, отстаиваются ли они членами МСЭ или другими сторонами вне процесса подготовки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ не получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: <http://www.itu.int/ITU-T/ipr/>.

© ITU 2008

Все права сохранены. Никакая часть данной публикации не может быть воспроизведена с помощью каких-либо средств без предварительного письменного разрешения МСЭ.

СОДЕРЖАНИЕ

| | <i>Стр.</i> |
|--|-------------|
| 1 Сфера применения | 1 |
| 2 Справочные документы | 1 |
| 3 Определения и сокращения | 1 |
| 3.1 Определения | 1 |
| 3.2 Сокращения | 3 |
| 4 Введение | 3 |
| 5 Соответствие | 3 |
| 6 Общая структура тестовой системы TTCN-3 | 4 |
| 6.1 Объекты в тестовой системе TTCN-3 | 4 |
| 6.2 Требования к выполнению для тестовой системы TTCN-3 | 7 |
| 7 Интерфейс управления и операции TTCN-3 | 7 |
| 7.1 Обзор интерфейса TCI | 7 |
| 7.2 Данные интерфейса TCI | 9 |
| 7.3 Операции интерфейса TCI | 18 |
| 8 Отображение данных языка Java | 83 |
| 8.1 Введение | 83 |
| 8.2 Имена и контексты | 83 |
| 8.3 Константы | 97 |
| 8.4 Отображение интерфейсов | 98 |
| 8.5 Факультативные параметры | 106 |
| 8.6 Инициализация TCI | 106 |
| 8.7 Обработка ошибок | 106 |
| 9 Отображение данных языка ANSI-C | 106 |
| 9.1 Введение | 106 |
| 9.2 Интерфейсы значений | 106 |
| 9.3 Интерфейс регистрации | 110 |
| 9.4 Интерфейсы операций | 110 |
| 9.5 Данные | 123 |
| 9.6 Разное | 125 |
| 10 Отображение языка XML консорциума W3C | 125 |
| 10.1 Введение | 125 |
| 10.2 Контексты | 125 |
| 10.3 Отображение типа | 126 |
| 10.4 Отображение операций в интерфейсе регистрации | 144 |
| 11 Сценарии использования | 166 |
| 11.1 Инициализация, сбор информации, регистрация | 166 |
| 11.2 Выполнение тестовых примеров и управления | 169 |
| 11.3 Обработка компонента | 172 |
| 11.4 Завершение тестовых примеров и управления | 179 |
| 11.5 Связь | 184 |
| Приложение А – Спецификация TCI на базе языка IDL | 189 |
| Приложение В – Отображение на языке XML для интерфейса TCI-TL provided | 204 |
| В.1 Схема для простых типов на языке XML интерфейса TCI-TL | 204 |
| В.2 Схема для типов на языке XML интерфейса TCI-TL | 204 |
| В.3 Схема для значений на языке XML интерфейса TCI-TL | 206 |
| В.4 Схема для шаблонов на языке XML интерфейса TCI-TL | 211 |
| В.5 Схема для событий на языке XML интерфейса TCI-TL | 218 |
| В.6 Схема для регистрации на языке XML интерфейса TCI-TL | 235 |
| ЛИТЕРАТУРА | 238 |

Нотация тестирования и управления тестом версии 3 (TTCN-3): Интерфейс управления (TCI)

1 Сфера применения

В настоящей Рекомендации приводится описание интерфейсов управления для реализаций тестовой системы согласно TTCN-3. Интерфейс управления TTCN-3 обеспечивает стандартизованную адаптацию администрирования, обработки тестового компонента и кодирования/декодирования тестовой системы к отдельной тестовой платформе. В настоящей Рекомендации интерфейсы определяются как набор операций, независимых от целевого языка.

Определения интерфейсов совместимы с положениями стандарта TTCN-3 (см. справочную литературу ниже). Для полного определения интерфейса TCI используется язык описания интерфейса (IDL) архитектуры CORBA. В разделах 8 и 9 приводится описание отображений языка абстрактной спецификации на целевые языки Java и ANSI-C. В Приложении А содержится резюме, касающееся спецификации интерфейса на базе языка IDL.

2 Справочные документы

Указанные ниже Рекомендации МСЭ-Т и другие источники содержат положения, которые путем ссылки на них в данном тексте составляют положения настоящей Рекомендации. На момент публикации указанные издания были действующими. Все Рекомендации и другие источники могут подвергаться пересмотру; поэтому всем пользователям данной Рекомендации предлагается изучить возможность применения последнего издания Рекомендаций и других источников, перечисленных ниже. Список действующих в настоящее время Рекомендаций МСЭ-Т, регулярно публикуется. Ссылка на документ в данной Рекомендации не придает ему как отдельному документу статус Рекомендации.

- [1] Рекомендация МСЭ-Т ITU-T Z.144 (2006 г.), *Нотация тестирования и управления тестом версии 3 (TTCN-3): Интерфейс времени выполнения (TRI)*.
- [2] Рекомендация МСЭ-Т ITU-T Z.140 (2006 г.), *Нотация тестирования и управления тестом версии 3 (TTCN-3): Базовый язык*.
- [3] Рекомендация МСЭ-Т Z.143 (2006 г.), *Нотация тестирования и управления тестом версии 3 (TTCN-3): Операционная семантика*.
- [4] ITU-T Recommendation МСЭ-Т X.290 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts*.
ISO/IEC 9646-1:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts*. [5] W3C Recommendation (2004), *XML Schema Part 0: Primer*.
NOTE – See at <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>.
- [6] W3C Recommendation (2004), *XML Schema Part 1: Structures*.
NOTE – See at <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>.
- [7] W3C Recommendation (2004), *XML Schema Part 2: Datatypes*.
NOTE – See at <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>.

3 Определения и сокращения

3.1 Определения

В данной Рекомендации применяются термины и определения, приведенные в Рекомендации МСЭ-Т X.290 [4], а также следующие определения:

3.1.1 абстрактная тестовая последовательность (Abstract Test Suite) (ATS): Тестовая последовательность, составленная из абстрактных тестовых примеров.

3.1.2 кодек (codec): Объект кодера/декодера, используемый для кодирования и декодирования данных, подлежащих соответственно передаче и приему.

3.1.3 кодирование/декодирование (Coding/Decoding) (CD): Объект, управляющий обработкой значения и типа, включая кодирование и декодирование в тестовой системе TTCN-3.

3.1.4 обработка компонентов (Component Handling) (CH): Объект, управляющий обработкой тестовых компонентов в тестовой системе TTCN-3.

3.1.5 порт связи (communication port): Абстрактный механизм, облегчающий связь между тестовыми компонентами.

ПРИМЕЧАНИЕ. – Порт связи моделируется в виде очереди FIFO в направлении приема. Порты могут базироваться на сообщениях, процедурах или на том и другом.

3.1.6 управляющий компонент (control component): Компонент, выполняющий программу поведения управляющей части модуля TTCN-3.

3.1.7 выполняемая тестовая последовательность (Executable Test Suite) (ETS): См. Рекомендацию МСЭ-Т X.290 [4].

3.1.8 дополнительная информация о реализации для тестирования (ДИРТ) (Implementation eXtra Information for Testing) (IXIT): См. Рекомендацию МСЭ-Т X.290 [4].

3.1.9 адаптер платформы (Platform Adaptor) (PA): Объект, который адаптирует выполняемый модуль TTCN-3 к отдельной платформе выполнения.

ПРИМЕЧАНИЕ. – Адаптер платформы создает для тестовой системы TTCN-3 единственное понятие времени и реализует явные и неявные таймеры, а также внешние функции.

3.1.10 реальный интерфейс тестовой системы (real test system interface): См. Рекомендацию МСЭ-Т X.290 [4].

3.1.11 тестируемая система (System Under Test) (SUT): См. Рекомендацию МСЭ-Т X.290 [4].

3.1.12 адаптер системы SUT (SUT Adaptor) (SA): Объект, который адаптирует операции связи TTCN-3 с системой SUT, базирующейся на абстрактном интерфейсе тестовой системы. Он осуществляет реальный интерфейс тестовой системы.

3.1.13 нотация тестирования и управления тестом (Testing and Test Control Notation) (TTCN-3): См. Рекомендацию МСЭ-Т X.290 [4].

3.1.14 тестовый пример (test case): См. Рекомендацию МСЭ-Т X.290 [4].

3.1.15 тестовое событие (test event): Переданные или принятые тестовые данные (сообщение или вызов процедуры) на порте связи, являющемся частью интерфейса тестовой системы, а также события тайм-аута таймеров.

3.1.16 администрирование тестом (Test Management) (TM): Объект, предоставляющий интерфейс пользователя, а также управляющий тестовой системой TTCN-3.

3.1.17 тестовая регистрация (Test Logging) (TL): Объект, предоставляющий регистрационную информацию о выполнении теста (включая также информацию, предоставляемую log-оператором TTCN-3).

3.1.18 администрирование и управление тестом (Test Management and Control) (TMC): Множество объектов, обеспечивающих администрирование и управление тестом; в это множество входят администрирование тестом (TM), обработка компонента (CH), регистрация тестов (TL) и кодирование/декодирование (CD).

ПРИМЕЧАНИЕ. – TMC является реализацией интерфейса TCI.

3.1.19 тестовая система (test system): См. Рекомендацию МСЭ-Т X.290 [4].

3.1.20 интерфейс тестовой системы (Test system interface) (TSI): Тестовый компонент, обеспечивающий отображение портов, доступных в (абстрактной) тестовой системе TTCN-3, на порты, предоставляемые реальной тестовой системой.

3.1.21 выполняемый модуль TTCN-3 (TTCN-3 Executable) (TE): Часть тестовой системы, которая отвечает за интерпретацию или выполнение тестовой последовательности ETS TTCN-3.

3.1.22 интерфейсы управления TTCN-3 (TTCN-3 Control Interfaces) (TCI): Три интерфейса, которые определяют взаимодействие выполняемого модуля TTCN-3 с администрированием теста, кодированием, декодированием и обработкой тестового компонента в тестовой системе.

3.1.23 интерфейс времени выполнения TTCN-3 (TTCN-3 Runtime Interface) (TRI): Интерфейс, определяющий взаимодействие выполняемого модуля TTCN-3 с системой SUT и адаптерами платформы в тестовой системе.

3.2 Сокращения

В настоящей Рекомендации применяются следующие сокращения:

| | | |
|--------|--|--|
| ATS | Abstract Test Suite | Абстрактная тестовая последовательность |
| CD | Coding/Decoding | Кодирование/декодирование |
| CH | Component Handling | Обработка компонентов |
| ETS | Executable Test Suite | Выполняемая тестовая последовательность |
| IDL | Interface Definition Language | Язык описания интерфейса |
| IXIT | Implementation eXtra Information for Testing | Дополнительная информация о реализации для тестирования (ДИРТ) |
| MSC | Message Sequence Chart | Схема последовательности сообщений |
| MTC | Main Test Component | Главный тестовый компонент |
| OMG | Object Management Group | Рабочая группа по управлению объектами |
| PA | Platform Adaptor | Адаптер платформы |
| PTC | Parallel Test Component | Параллельный тестовый компонент |
| SA | SUT Adaptor | Адаптер системы SUT |
| SUT | System Under Test | Тестируемая система |
| TC | Test Control | Управление тестом |
| TCI | TTCN-3 Control Interfaces | Интерфейсы управления TTCN-3 |
| TE | TTCN-3 Executable | Выполняемый модуль TTCN-3 |
| TL | Test Logging | Тестовая регистрация |
| TM | Test Management | Администрирование тестом |
| TMC | Test Management and Control | Администрирование и управление тестом |
| TRI | TTCN-3 Runtime Interface | Интерфейс времени выполнения TTCN-3 |
| TSI | Test System Interface | Интерфейс тестовой системы |
| TTCN-3 | Testing and Test Control Notation Version 3 | Нотация тестирования и управления тестом версии 3 |

4 Введение

Настоящая Рекомендация состоит из двух отдельных частей. В первой части приводится описание структуры реализации тестовой системы TTCN-3, а во второй части представлена спецификация интерфейсов управления TTCN-3.

В первой части в результате декомпозиции тестовой системы TTCN-3 выделяются четыре основных объекта:

- Администрирование и управление тестом (ТМС);
- Выполняемый модуль TTCN-3 (ТЕ);
- Адаптер системы SUT (SA); и
- Адаптер платформы (РА).

Сам объект ТМС состоит из трех объектов: администрирование тестом (ТМ), кодер/декодер (CD) и обработчик тестового компонента (СН). Кроме того, определяется взаимодействие между этими объектами, то есть соответствующие интерфейсы.

Во второй части настоящей Рекомендации дается подробное описание интерфейсов управления TTCN-3 (ТСИ). Эти интерфейсы рассматриваются в контексте операций, реализуемых как часть одного объекта и вызываемых другими объектами тестовой системы. Для каждой операции в спецификации интерфейса определяются соответствующие структуры данных, ожидаемое влияние на тестовую систему и все ограничения на использование операции. Следует заметить, что в этих спецификациях интерфейсов анализируются взаимодействия только между ТЕ и ТМ, ТЕ и CD, а также ТЕ и СН. Что касается взаимодействий между ТЕ и SA, а также ТЕ и РА, то следует обратиться к спецификации интерфейса времени выполнения TTCN-3 (Рекомендация МСЭ-Т Z.144 [1]).

5 Соответствие

Минимальным требованием в отношении соответствия тестовой системы TTCN-3 интерфейсу TCI является соблюдение спецификации интерфейса, представленной в настоящей Рекомендации. Семантика TTCN-3 в тестовой системе должна соответствовать операционной семантике, определенной в Рекомендации МСЭ-Т Z.143 [3]. Кроме того, должно поддерживаться отображение одного языка. Например, если поставщик поддерживает язык Java, то вызовы операций и реализации интерфейса TCI, являющиеся частью выполняемого модуля TTCN-3, должны соответствовать отображению языка IDL на язык Java, заданному в настоящей Рекомендации. Что касается интерфейса регистрации, то вместо отображения языка Java или языка C может быть использовано отображение языка XML.

6 Общая структура тестовой системы TTCN-3

Тестовую систему TTCN-3 можно представить концептуально в виде множества взаимодействующих объектов. Каждый объект осуществляет конкретные функции тестовой системы. Эти объекты:

- управляют выполнением теста;
- интерпретируют или выполняют скомпилированную программу TTCN-3;
- осуществляют надлежащую связь с системой SUT;
- управляют типами, значениями и тестовыми компонентами;
- реализуют внешние функции; и
- обрабатывают операции таймеров.

6.1 Объекты в тестовой системе TTCN-3

Структура реализации тестовой системы TTCN-3 представлена на Рисунке 1.

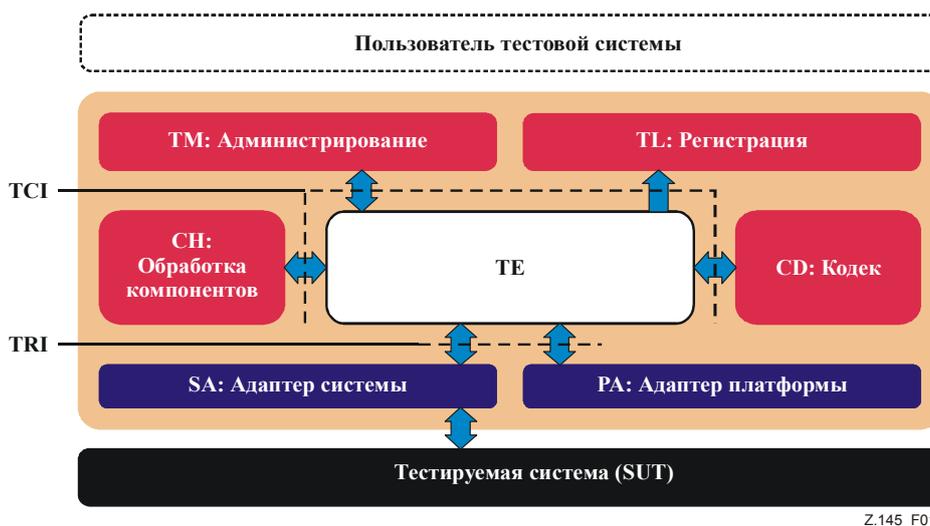


Рисунок 1/Z.145 – Общая структура тестовой системы TTCN-3

Как показано на Рисунке 1, выполняемый модуль TTCN-3 (TE), который также называют выполняемой тестовой последовательностью (ETS), интерпретирует и выполняет модули TTCN-3. Могут быть идентифицированы различные структурные элементы модуля TE: управление, поведение, компоненты, типы, значения и очереди. Структурные элементы в модуле TE обеспечивают функциональные возможности, которые определяются в модуле TTCN-3 или самим стандартом TTCN-3 (Рекомендация МСЭ-Т Z.140 [2]). Например, структурный элемент "Control" ("Управление") представляет управляющую часть внутри модуля TTCN-3, в то время как структурный элемент "Queues" ("Очереди") представляет требование для выполняемого модуля TTCN-3 о том, чтобы каждый порт тестового компонента поддерживал свою собственную очередь порта. В то время как первый элемент точно определяется в модуле TTCN-3, второй элемент требуется для спецификации TTCN-3.

Как показано на Рисунке 1, уточнения относительно блока TE предоставляются в качестве вспомогательного средства при определении интерфейсов управления TTCN-3. Модуль TE в реализации тестовой системы обычно соответствует выполняемой программе, порожденной либо компилятором TTCN-3, либо интерпретатором TTCN-3.

Модуль TE может быть выполнен централизованным или распределенным способом; то есть на одиночном тестовом устройстве или с помощью нескольких тестовых устройств, соответственно.

И хотя структурные объекты модуля ТЕ реализуют полный модуль TTCN-3, отдельные структурные элементы могут быть распределены по нескольким тестовым устройствам.

ТЕ реализует модуль TTCN-3 на абстрактном уровне. Другие объекты тестовой системы TTCN-3 конкретизируют эти абстрактные понятия. Например, абстрактное понятие передачи сообщения или приема данных о тайм-ауте не может быть реализовано в ТЕ. Остальная часть тестовой системы соответственно реализует кодирование сообщения и его передачу по конкретным физическим устройствам или измерение времени и определение момента, когда истекло установленное на таймере время.

Адаптеры SA и PA и их взаимодействие с ТЕ описываются в Рекомендации МСЭ-Т Z.144 [1]. В спецификации интерфейса TCI определяется взаимодействие между ТЕ и ТМС.

Интерфейс регистрации предоставляет возможности регистрации всем элементам архитектуры тестовой системы, то есть ТЕ, ТМ, СН, CD, SA и PA могут регистрировать информацию о выполнении теста через TL. Более детализированный вид TL представлен на Рисунке 2.

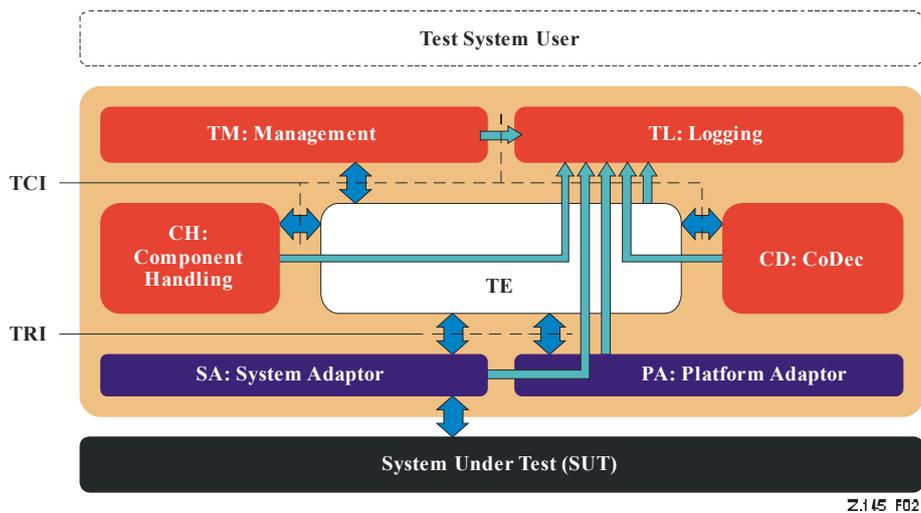


Рисунок 2/Z.145 – Детализированный вид TL

6.1.1 Администрирование и управление тестом (ТМС)

Объект ТМС включает функции, связанные с администрированием:

- выполнения теста;
- компонентов; и
- кодирования и декодирования.

6.1.1.1 Администрирование теста (ТМ)

Объект ТМ отвечает за все администрирование тестовой системы. После инициализации тестовой системы выполнение теста начинается в объекте ТМ. Этот объект отвечает за надлежащее иницирование модулей TTCN-3, то есть за передачу параметров модулей, таких как информация ДИРТ, на объект ТЕ, если это необходимо. Обычно этот объект также реализует пользовательский интерфейс тестовой системы.

6.1.1.2 Кодирование и декодирование (CD)

Объект CD отвечает за кодирование и декодирование значений TTCN-3 в битовые строки, пригодные для передачи тестируемой системе. Модуль ТЕ определяет, какие кодеки должны быть использованы. Для получения закодированных данных он пересылает данные TTCN-3 соответствующему кодирующему устройству. Полученные данные декодируются в объекте CD путем использования соответствующего декодирующего устройства, где они преобразуются в значения TTCN-3.

6.1.1.3 Обработка компонентов (СН)

ТЕ может быть распределен между несколькими тестовыми устройствами. Объект СН осуществляет связь между распределенными объектами тестовой системы. Он предоставляет средства для синхронизации объектов тестовой системы, которые могли бы быть распределены по нескольким узлам.

ПРИМЕЧАНИЕ 1. – Узлы и тестовые устройства используются здесь как синонимы.

На Рисунке 3 изображена общая структура тестовой системы, распределенной среди нескольких узлов.

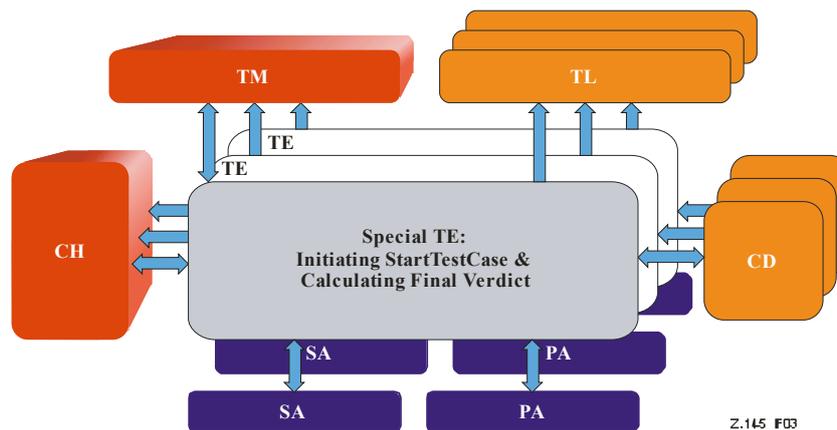


Рисунок 3/Z.145 – Общая структура распределенной тестовой системы TTCN-3

Каждый узел в тестовой системе содержит объекты TE, SA, PA, CD и TL. Объекты CH и TM являются промежуточными звеньями между объектами TE в каждом узле, осуществляя администрирование тестов и обработку тестовых компонентов. Объект TE, который запускает тестовый пример, является специальным объектом TE. Он вычисляет конечный вердикт тестового примера. Кроме того, все объекты TE обрабатываются одним и тем же способом.

ПРИМЕЧАНИЕ 2. – Тестовая система выполняет не более одного тестового примера в данный момент времени. То есть модуль TTCN-3 не может выполнять несколько тестовых примеров в одно и то же время.

Создание МТС, РТС и управляющего компонента в объектах TE контролируется объектом CH. Следует отметить особую роль системного компонента, который существует только концептуально, а не в качестве выполняемого тестового компонента в TE. Системные порты, то есть порты системного компонента, могут быть распределены среди нескольких узлов. Кроме того, тестовые компоненты в различных узлах могут иметь доступ к одному и тому же физическому порту системы SUT, то есть они могут отображаться на один и тот же порт интерфейса тестовой системы.

ПРИМЕР: Доступ к удаленным реальным портам системы SUT может быть выполнен объектами TE через локальные модули-посредники.

Связь между компонентами TTCN-3 базируется либо на сообщениях, либо на процедурах. Поэтому объект CH приспособливает базирующуюся на сообщениях и процедурах связь компонентов TTCN-3 к отдельной платформе выполнения тестовой системы. Он осведомлен о соединениях между портами связи тестового компонента TTCN-3 и передает операции запроса передачи от одного компонента TTCN-3 к другому компоненту TTCN-3. Принимающий компонент может находиться в другом экземпляре одного и того же объекта TE, расположенного в другом узле. Затем он уведомляет объект TE о всех полученных тестовых событиях путем постановки их на очередь в портах этого объекта.

Операции связи на базе процедур между компонентами TTCN-3 также являются видимыми в объекте CH. Объект CH отличает друг от друга различные виды связи на базе процедур, то есть вызов, ответ и особое состояние, и пересылает их надлежащим образом на объект TE, где находится целевой компонент. Семантика связи TTCN-3 на базе процедур, то есть влияние такой операции на выполнение тестового компонента TTCN-3, должна обрабатываться в объекте TE.

Для реализации распределения тестовых компонентов по нескольким узлам необходимо наличие дополнительной связи. Связь для управления компонентом включает в себя индикацию о создании тестовых компонентов, о запуске выполнения тестового компонента, о распределении вердиктов, а также о завершении компонента. Объект CH не реализует поведение компонента TTCN-3. Скорее, он осуществляет связь между несколькими компонентами, реализованными объектом TE.

6.1.1.4 Тестовая регистрация (TL)

Объект TL выполняет регистрацию тестовых событий и их представление пользователю тестовой системы. Он обеспечивает регистрацию информации о выполнении теста, например, какие тестовые компоненты созданы, запущены и завершены, какие данные передаются системе SUT, принимаются от системы SUT и сопоставляются с шаблонами TTCN-3, какие таймеры были запущены, остановлены или находились в состоянии тайм-аута и т. д.

6.1.2 Выполняемый модуль TTCN-3 (TE)

Объект TE выполняет или интерпретирует модуль TTCN-3. Концептуально объект TE можно разбить на шесть взаимодействующих объектов: Control (Управление), Behaviour (Поведение), Component (Компонент), Type (Тип), Value (Значение) и Queue (Очередь). Эта структурная декомпозиция объекта TE определяется в Рекомендации МСЭ-Т Z.144 [1]. В настоящей Рекомендации используется терминология для TE, определяемая в Рекомендации МСЭ-Т Z.144 [1].

6.1.3 Адаптер системы SUT (SA)

Адаптер SA – это реализация адаптера тестируемой системы (SA), как определено в [1]. В настоящей Рекомендации используется терминология для SA, определяемая в Рекомендации МСЭ-Т Z.144 [1].

6.1.4 Адаптер платформы (PA)

Адаптер PA – это реализация адаптера платформы (PA), как определено в Рекомендации МСЭ-Т Z.144 [1]. В настоящей Рекомендации используется терминология для PA, определяемая в Рекомендации МСЭ-Т Z.144 [1].

6.2 Требования к выполнению для тестовой системы TTCN-3

Каждая операция интерфейса TCI должна рассматриваться как элементарная операция в вызывающем объекте. Вызываемый объект, выполняющий операцию интерфейса TCI, должен возвращать управление вызывающему объекту, как только будет получен ожидаемый им результат или если эта операция не может быть завершена успешно. Вызываемый объект не должен блокировать осуществление связи на базе процедур.

Как сказано выше, не делается никаких предположений относительно того, реализуются ли тестовая система TTCN-3 или индивидуальные объекты в одном выполняемом модуле или процессе или же они распределяются между разными процессами или даже тестовыми устройствами.

При реализации интерфейса TCI должны выполняться упомянутые выше требования.

7 Интерфейс управления и операции TTCN-3

В данном разделе определяется множество абстрактных типов данных, используемых для представления данных обмена между TE и TMC. Кроме того, приводятся характеристики операций интерфейса TCI в контексте их сигнатур, а также данные о том, когда они должны использоваться и каково их воздействие на тестовую систему TTCN-3.

В это определение также включено более подробное описание входных параметров, необходимых для каждого вызова операции интерфейса TCI и ее возвращаемого значения.

7.1 Обзор интерфейса TCI

Интерфейс TCI определяет взаимодействие между объектами выполняемого модуля TTCN-3 (TE), обработки компонентов (CH), администрирования тестом (TM), кодирования/декодирования (CD), тестовой регистрации (TL) в тестовой системе TTCN-3. Он предоставляет средства для объекта TE в целях:

- управления процессом выполнения теста;
- распределения процесса выполнения тестовых компонентов среди разных тестовых устройств;
- кодирования и декодирования тестовых данных; и
- регистрации информации о выполнении теста.

В состав интерфейса TCI входят четыре субинтерфейса:

- **Интерфейс администрирования тестом в TCI (TCI-TM):** Данный интерфейс включает все операции, необходимые для управления выполнением теста, для предоставления параметров модуля и внешних констант и для обеспечения регистрации тестовых событий.
- **Интерфейс обработки компонентов в TCI (TCI-CH):** Данный интерфейс состоит из операций, необходимых для реализации администрирования тестовыми компонентами TTCN-3 и связи между ними в централизованной или распределенной тестовой системе. Он включает операции по созданию, запуску и останову тестовых компонентов, установлению соединений между компонентами TTCN-3, управлению тестовыми компонентами и их вердиктами и обработке связи между компонентами TTCN-3, базирующейся на сообщениях и процедурах.
- **Интерфейс кодирования/декодирования в TCI (TCI-CD):** Данный интерфейс включает все операции, необходимые для поиска кодеков и доступа к ним, то есть кодеров и декодеров для кодирования подлежащих передаче данных, определяемых путем использования атрибута кодирования TTCN-3, и для декодирования принятых данных.

- **Интерфейс тестовой регистрации в TCI (TCI-TL):** Данный интерфейс включает все операции, необходимые для вывода информации о выполнении теста и для управления уровнем детализации этой информации.

Все интерфейсы являются двунаправленными, так что вызывающая и вызываемая стороны находятся в объектах TE и в TMC тестовой системы. Предоставляемые интерфейсы (те операции, которые интерфейс предлагает объекту TE) и требуемые операции (те операции, которые интерфейсу необходимы для использования от объекта TE) объединяются в соответствующий предоставляемый (provided) и требуемый (required) субинтерфейс для каждого интерфейса; то есть TCI-TM Provided/TCI-TM Required, TCI-CH Provided/TCI-CH Required, TCI-CD Provided/TCI-CD Required и TCI-TL Provided/TCI-TL Required.

7.1.1 Корреляция между инициированиями операций TTCN-3 и TCI

Для ряда инициирований операций TTCN-3 существует положительная корреляция с инициированием операций TCI, что показано в таблице 1. Некоторые операции TTCN-3 коррелируют с парой, состоящей из запроса операции TCI и операции TCI, для осуществления передачи операций TTCN-3 через тестовую систему. Для других инициирований операций TCI не существует положительной корреляции – они нужны для реализации в виде TTCN-3 семантики базовых понятий.

Корреляция, показанная для операций связи TTCN-3 (то есть для , и), сохраняется, если только эти операции иницируются в порте тестового компонента, соединенном с другим портом тестового компонента. В Рекомендации МСЭ-Т Z.144 [1] определяется корреляция для операций связи, которые иницируются в портах тестового компонента, отображаемых на порты интерфейса тестовой системы.

Таблица 1/Z.145 – Корреляция между инициированиями операций TTCN-3 и TCI

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

Таблица 1/Z.145 – Корреляция между иницированиями операций TTCN-3 и TCI

| Имя операции TTCN-3 | Имя операции TCI | Имя интерфейса TCI |
|---|---------------------------|--------------------|
| | tciTestComponentDone | TCI-CH Required |
| killed | tciTestComponentKilledReq | TCI-CH Provided |
| | tciTestComponentKilled | TCI-CH Required |
| mtc | tciGetMTCReq | TCI-CH Provided |
| | tciGetMTC | TCI-CH Required |
| execute | tciTestCaseExecuteReq | TCI-CH Provided |
| | tciTestCaseExecute | TCI-CH Required |
| log | tliLog | TCI-TL Provided |
| ПРИМЕЧАНИЕ 1. – Для одноадресной связи. ПРИМЕЧАНИЕ 2. – Для широковещательной связи. ПРИМЕЧАНИЕ 3. – Для многоадресной связи. | | |

7.2 Данные интерфейса TCI

В спецификации TCI определяется множество абстрактных типов данных. Они описывают на очень высоком уровне, какой вид данных должен пересылаться от вызывающего объекта к вызываемому объекту. Абстрактные типы данных используются, чтобы определить:

- каким образом данные TTCN-3 передаются от ТЕ к кодирующему устройству для кодирования представлений значений TTCN-3 в битовую строку и обратно;
- каким образом данные, передаваемые от декодирующего устройства к ТЕ, должны декодироваться из битовой строки в их представление значений TTCN-3.

Для этих абстрактных типов данных определяется множество операций по обработке данных кодером/декодером.

Конкретное представление этих абстрактных типов данных, а также определение базовых типов данных наподобие и приведены в соответствующих отображениях языка в разделах 8 и 9.

Следует заметить, что значения для любого типа данных идентификатора должны быть уникальными в реализации тестовой системы, где уникальность определяется как глобально различимое в любой момент времени. Это гарантирует, что различные объекты, например, два таймера, идентифицируются различными идентификаторами, и идентификаторы повторно не используются.

7.2.1 Основные абстрактные типы данных

Для определения операций TCI описываются и используются следующие абстрактные типы данных:

7.2.1.1 Администрирование

| | |
|----------------------------|---|
| TciModuleIdType | Значение типа TciModuleIdType является именем модуля TTCN-3, как указывается в последовательности ATS TTCN-3. Этот абстрактный тип используется для обработки параметров модуля. |
| TciModuleParameterIdType | Значение типа TciModuleParameterIdType является уточненным именем параметра модуля TTCN-3, как указывается в последовательности ATS TTCN-3. Этот абстрактный тип используется для обработки параметров модуля. |
| TciTestCaseIdType | Значение типа TciTestCaseIdType является уточненным именем тестового примера TTCN-3, как указывается в последовательности ATS TTCN-3. Этот абстрактный тип используется для обработки тестового примера. |
| TciModuleIdListType | Значение типа TciModuleIdListType является списком TciModuleIdType. Этот абстрактный тип используется при выводе списка модулей, которые импортируются модулем TTCN-3. |
| TciModuleParameterType | Значение типа TciModuleParameterType является структурой из TciModuleParameterIdType и Value. Этот абстрактный тип используется для представления имени параметра и значения по умолчанию параметра модуля. |
| TciModuleParameterListType | Значение типа TciModuleParameterListType является списком TciModuleParameterType. Этот абстрактный тип используется при выводе параметров модуля TTCN-3. |
| TciParameterType | Значение типа TciParameterType включает значение Value TTCN-3 и значение TciParameterPassingModeType в целях представления режима пересылки параметра, заданного для параметра в последовательности ATS TTCN-3. |

| | |
|--|--|
| <code>TciParameterPassingModeType</code> | Значение типа <code>TciParameterPassingModeType</code> представляет собой либо <code>IN</code> , <code>INOUT</code> , либо <code>OUT</code> . Этот абстрактный тип используется при запуске тестового примера или когда имеется индикация о завершении тестового примера. |
| <code>TciParameterListType</code> | Значение типа <code>TciParameterListType</code> является списком <code>TciParameterType</code> . Этот абстрактный тип используется при запуске тестового примера или когда имеется индикация о завершении тестового примера. |
| <code>TciParameterTypeType</code> | Значение типа <code>TciParameterTypeType</code> является структурой из <code>Type</code> и <code>TciParameterPassingModeType</code> . Этот абстрактный тип используется для представления типа и режима пересылки параметра тестового примера. |
| <code>TciParameterTypeListType</code> | Значение типа <code>TciParameterTypeListType</code> является списком <code>TciParameterTypeType</code> . Этот абстрактный тип используется для представления списка параметров тестового примера. |
| <code>TciTestComponentKindType</code> | Значение типа <code>TciTestComponentKindType</code> является литералом множества видов тестовых компонентов TTCN-3, то есть <code>MTC</code> , <code>PTC</code> , <code>PTC_ALIVE</code> и <code>CONTROL</code> . Этот абстрактный тип используется для обработки компонентов. |
| <code>TciTypeClassType</code> | Значение типа <code>TciTypeClassType</code> является литералом множества классов типа в TTCN-3, таких как <code>boolean</code> , <code>float</code> , <code>record</code> и т. д. Этот абстрактный тип используется для обработки значений. |

7.2.1.2 Связь

`TciBehaviourIdType` Значение типа `TciBehaviourIdType` определяет функции поведения TTCN-3.

Дополнительные абстрактные типы данных с префиксом `Tri` взяты из Рекомендации МСЭ-Т Z.144 [1]: `TriPortIdType`, `TriPortIdListType`, `TriComponentIdType`, `TriComponentIdListType`, `TriAddressType`, `TriAddressListType` и `TriMessageType`.

7.2.2 Абстрактные типы данных и значения в TTCN-3

В данном разделе определяется множество абстрактных типов данных, которые образуют представление типа и значения в TTCN-3. Функциональные возможности каждого типа данных определяются с помощью сопровождающего множества операций. Операции в отношении данного типа абстрактных данных или операции, использующие этот тип, возвращают либо значение этого абстрактного типа, либо базовый тип наподобие .

Все операции были определены путем использования языка описания интерфейса (IDL). Конкретные отображения языка для операций над типами абстрактных данных представлены в разделах 8 и 9. В определенных языках применение операции над абстрактными типами данных представляется путем пересылки (либо по значению, либо по ссылке, в зависимости от отображения) конкретного значения в качестве параметра для операции. В других языках может быть выбран иной метод ссылки на конкретное значение, например, путем рассмотрения значения в качестве объекта, на который инициируется метод, соответствующий данной операции. Для указания на невозможность выполнения определенной задачи или на отсутствие факультативного параметра в дальнейшем используется особое значение . Его можно рассматривать как зарезервированное значение, указывающее на специальное значение. Отображения языка будут определять конкретное представление этого особого значения .

В состав представления абстрактного типа и значения TTCN-3 входят две части:

- абстрактный тип данных , представляющий все типы TTCN-3 в модуле TTCN-3;
- различные абстрактные типы данных, представляющие значения TTCN-3, то есть значения TTCN-3 данного типа TTCN-3. Это могут быть либо значения предопределенных типов TTCN-3, либо значения определяемых пользователем типов TTCN-3.

Для доступа, оценки и кодирования данных TTCN-3 в тестовой системе используется абстрактный тип данных и различные абстрактные типы данных значения. Следовательно эти абстрактные типы данных определяют уровень абстракции между выполняемым модулем TTCN-3 (TE) и остальной тестовой системой, использующей интерфейс TCI.

7.2.2.1 Абстрактные типы данных TTCN-3

Согласно настоящей Рекомендации типы TTCN-3, предопределенные или определяемые пользователем, будут представлены в интерфейсах TCI, использующих абстрактный тип данных .

Для абстрактного типа данных определяется набор операций, чтобы:

- ссылаться на предопределенные и определяемые пользователем типы данных TTCN-3; и
- создавать и поддерживать значения TTCN-3.

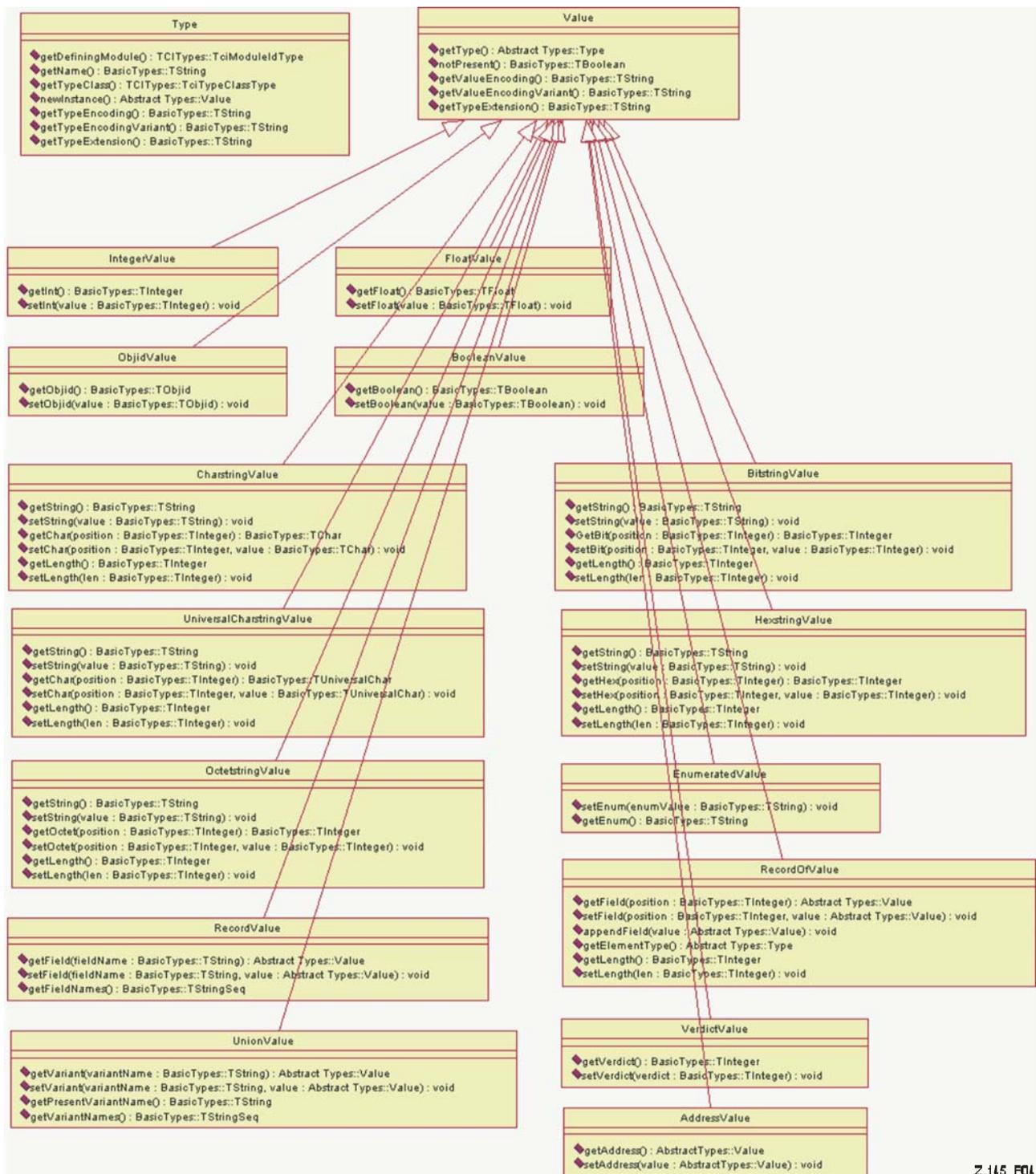
Для абстрактного типа данных определяются следующие операции:

| | |
|--|--|
| <code>TciModuleIdType getDefiningModule()</code> | Возвращает идентификатор модуля для модуля, в котором определяется тип. Возвращает особое значение <code>null</code> , если тип является базовым типом TTCN-3, например, <code>boolean</code> , <code>integer</code> и т. д. |
| <code>TString getName()</code> | Возвращает имя типа, как определено в модуле TTCN-3. |
| <code>TciTypeClassType getTypeClass()</code> | Возвращает класс типа для соответствующего типа. Значение типа <code>TciTypeClassType</code> может иметь одну из следующих констант: <code>ADDRESS</code> , <code>ANYTYPE</code> , <code>BITSTRING</code> , <code>BOOLEAN</code> , <code>CHARSTRING</code> , <code>COMPONENT</code> , <code>ENUMERATED</code> , <code>FLOAT</code> , <code>HEXSTRING</code> , <code>INTEGER</code> , <code>OBJID</code> , <code>OCTETSTRING</code> , <code>RECORD</code> , <code>RECORD_OF</code> , <code>SET</code> , <code>SET_OF</code> , <code>UNION</code> , <code>UNIVERSAL_CHARSTRING</code> , <code>VERDICT</code> . |
| <code>Value newInstance()</code> | Возвращает вновь созданное значение данного типа. Это начальное значение для созданного значения является неопределенным. |
| <code>TString getTypeEncoding()</code> | Возвращает атрибут кодирования типа, как определено в модуле TTCN-3. |
| <code>TString getTypeEncodingVariant()</code> | Эта операция возвращает атрибут варианта кодирования значения, как определено в TTCN-3, если он существует. Если атрибут варианта кодирования не определен, то возвращается особое значение <code>null</code> . |
| <code>TStringseq getTypeExtension()</code> | Возвращает атрибут расширения типа, как определено в модуле TTCN-3. |

7.2.2.2 Абстрактные значения TTCN-3

В соответствии с настоящей Рекомендацией значения TTCN-3 в интерфейсах TCI представляются через многочисленные абстрактные типы данных.

На Рисунке 4 представлена иерархия абстрактных типов данных для значений TTCN-3 (кратко: абстрактных значений).



Z.145_F04

Рисунок 4/Z.145 – Иерархия абстрактных значений

Как показано на Рисунке 4, во всех абстрактных значениях TTCN-3 используется один и тот же базовый абстрактный тип данных. Все операции, определяемые в отношении этого общего базового типа данных, неявно определяются также для абстрактных типов значений, полученных из базового типа.

7.2.2.2.1 Абстрактный тип данных

В отношении базового абстрактного типа данных Value определяются следующие операции. Конкретные представления этих операций определяются в соответствующих разделах отображения языка:

Type getType ()

Возвращает тип заданного значения.

TBoolean notPresent ()

Возвращает true, если заданным значением является omit, в противном случае возвращает false.

| | |
|--|--|
| <code>TString getValueEncoding()</code> | Возвращает атрибут кодирования значения, как определено в TTCN-3, если он существует. Если атрибут кодирования не определен, то возвращается особое значение <code>null</code> . |
| <code>TString getValueEncodingVariant()</code> | Возвращает атрибут варианта кодирования значения, как определено в TTCN-3, если он существует. Если атрибут варианта кодирования не определен, то возвращается особое значение <code>null</code> . |

7.2.2.2.2 Абстрактный тип данных

Абстрактный тип данных `IntegerValue` базируется на абстрактном типе данных `Integer`. Он представляет значения `integer` TTCN-3.

В отношении абстрактного типа данных определяются следующие операции:

| | |
|---|---|
| <code>TInteger getInt()</code> | Возвращает целочисленное значение данного типа <code>integer</code> TTCN-3. |
| <code>void setInt(in TInteger value)</code> | Устанавливает этот тип <code>IntegerValue</code> на <code>value</code> . |

7.2.2.2.3 Абстрактный тип данных

Абстрактный тип данных `FloatValue` базируется на абстрактном типе данных `Float`. Он представляет значения `float` TTCN-3.

В отношении абстрактного типа данных определяются следующие операции:

| | |
|---|---|
| <code>TFloat getFloat()</code> | Возвращает значение с плавающей запятой данного типа <code>float</code> TTCN-3. |
| <code>void setFloat(in TFloat value)</code> | Устанавливает данный тип <code>FloatValue</code> на <code>value</code> . |

7.2.2.2.4 Абстрактный тип данных

Абстрактный тип данных `BooleanValue` базируется на абстрактном типе данных `Boolean`. Он представляет значения `boolean` TTCN-3.

В отношении абстрактного типа данных определяются следующие операции:

| | |
|---|--|
| <code>TBoolean getBoolean()</code> | Возвращает булево значение типа <code>boolean</code> TTCN-3. |
| <code>void setBoolean(in TBoolean value)</code> | Устанавливает это булево значение на <code>value</code> . |

7.2.2.2.5 Абстрактный тип данных

Абстрактный тип данных `ObjidValue` базируется на абстрактном типе данных `Objid`. Он представляет значения `objid` TTCN-3.

В отношении абстрактного типа данных определяются следующие операции:

| | |
|---|--|
| <code>TObjid getObjid()</code> | Возвращает значение идентификатора объекта типа <code>objid</code> TTCN-3. |
| <code>void setObjid(in TObjid value)</code> | Устанавливает данный тип <code>ObjidValue</code> на <code>value</code> . |

7.2.2.2.6 Абстрактный тип данных

Абстрактный тип данных `CharstringValue` базируется на абстрактном типе данных `Charstring`. Он представляет значения типа `charstring` TTCN-3. – это символ в значении типа символьной строки.

В отношении абстрактного типа данных `CharstringValue` определяются следующие операции:

| | |
|---|---|
| <code>TString getString()</code> | Возвращает значение строки типа <code>charstring</code> TTCN-3. Текстовое представление пустого типа <code>charstring</code> TTCN-3 – это <code>''</code> , в то время как его длина равна нулю. |
| <code>void setString(in TString value)</code> | Устанавливает этот тип <code>CharstringValue</code> на <code>value</code> . |
| <code>TChar getChar(in TInteger position)</code> | Возвращает значение символа символьной строки <code>charstring</code> TTCN-3 на <code>position</code> . Позиция 0 обозначает первый символ типа <code>charstring</code> TTCN-3. Диапазон допустимых значений для позиции составляет от 0 до <code>length - 1</code> . |
| <code>void setChar(in TInteger position, in TChar value)</code> | Устанавливает символ в позиции на значение. Диапазон допустимых значений для позиции – от 0 до <code>length - 1</code> . |
| <code>TInteger getLength()</code> | Возвращает длину этого типа <code>CharstringValue</code> в символах; возвращается нуль, если значение этого типа <code>CharstringValue</code> пропущено. |
| <code>void setLength(in TInteger len)</code> | <code>setLength</code> сначала переустанавливает этот тип <code>CharstringValue</code> на его начальное значение, а затем устанавливает длину этого типа <code>CharstringValue</code> в символах на <code>len</code> . |

7.2.2.2.7 Абстрактный тип данных UniversalCharstringValue

Абстрактный тип данных UniversalCharstringValue базируется на абстрактном типе данных Value. Он представляет значения типа universal charstring TTCN-3. TUniversalChar – это символ в значении типа универсальной строки символов.

В отношении абстрактного типа данных UniversalCharstringValue определяются следующие операции:

| | |
|---|---|
| TString getString() | Возвращает текстовое представление данного типа UniversalCharstringValue, как определено в TTCN-3. |
| void setString(in TString value) | Устанавливает значение данного типа UniversalCharstringValue согласно текстовому представлению, как определено значением value. |
| TUniversalChar getChar(in TInteger position) | Возвращает значение универсального символа типа universal charstring TTCN-3 на позицию. Позиция 0 обозначает первый символ TUniversalChar типа universal charstring TTCN-3. Допустимые значения для позиции находятся в диапазоне от 0 до length - 1. |
| void setChar(in TInteger position, in TUniversalChar value) | Устанавливает универсальный символ в позиции на значение. Допустимые значения для позиции находятся в диапазоне от 0 до length - 1. |
| TInteger getLength() | Возвращает длину данного значения типа универсальной строки символов в универсальных символах; возвращает нуль, если значение этого типа пропущено. |
| void setLength(in TInteger len) | setLength сначала переустанавливает этот тип UniversalCharstringValue на его начальное значение, а затем устанавливает длину этого типа UniversalCharstringValue в универсальных символах на len. |

7.2.2.2.8 Абстрактный тип данных BitstringValue

Абстрактный тип данных BitstringValue базируется на абстрактном типе данных Value. Он представляет значения типа bitstring TTCN-3:

В отношении абстрактного типа данных BitstringValue определяются следующие операции.

| | |
|---|--|
| TString getString() | Возвращает текстовое представление данного типа BitstringValue, как определено в TTCN-3. Например, текстовым представлением для 0101 является "0101"В. Текстовым представлением пустого типа bitstring TTCN-3 является ""В, в то время как его длина равна нулю. |
| void setString(in TString value) | Устанавливает значение данного типа BitstringValue согласно текстовому представлению, как определено значением value. Например, значением данного типа BitstringValue является 0101, если текстовым представлением для value является "0101"В. |
| TChar getBit(in TInteger position) | Возвращает в качестве символа значение (0 1) на позицию данного типа битовой строки TTCN-3. Позиция 0 обозначает первый бит битовой строки TTCN-3. Допустимые значения для позиции находятся в диапазоне от 0 до length - 1. |
| void setBit(in TInteger position, TInteger value) | Устанавливает бит в позиции на значение (0 1). Позиция 0 обозначает первый бит в этом типе BitstringValue. Допустимые значения для позиции находятся в диапазоне от 0 до length - 1. |
| TInteger getLength() | Возвращает длину данного типа BitstringValue в битах; возвращает нуль, если значением данного типа BitstringValue является omit. |
| void setLength(in TInteger len) | setLength сначала переустанавливает тип BitstringValue на его начальное значение, а затем устанавливает длину этого типа BitstringValue в битах на len. |

7.2.2.2.9 Абстрактный тип данных OctetstringValue

Абстрактный тип данных OctetstringValue базируется на абстрактном типе данных Value. Он представляет значения типа octetstring TTCN-3.

В отношении абстрактного типа данных `OctetstringValue` определяются следующие операции:

| | |
|---|---|
| <code>TString getString()</code> | Возвращает текстовое представление данного типа <code>OctetstringValue</code> , как определено в TTCN-3. Например, текстовым представлением для <code>0xCAFFEE</code> является "CAFFEE"O. Текстовым представлением пустого типа <code>octetstring</code> TTCN-3 является "", в то время как его длина равна нулю. |
| <code>void setString(in TString value)</code> | Устанавливает значение данного типа <code>OctetstringValue</code> согласно текстовому представлению, как определено значением <code>value</code> . Например, значением данного типа <code>OctetstringValue</code> является <code>0xCAFFEE</code> , если текстовым представлением для <code>value</code> является "CAFFEE"O. |
| <code>TChar getOctet(in TInteger position)</code> | Возвращает значение (0..255) на позицию данной строки октетов TTCN-3. Позиция 0 обозначает первый октет строки октетов TTCN-3. Допустимые значения для позиции находятся в диапазоне от 0 до <code>length - 1</code> . |
| <code>void setOctet(in TInteger position, in TInteger value)</code> | Устанавливает октет в позиции на значение (0..255). Позиция 0 обозначает первый октет в строке октетов. Допустимые значения для позиции находятся в диапазоне от 0 до <code>length - 1</code> . |
| <code>TInteger getLength()</code> | Возвращает длину данного типа <code>OctetstringValue</code> в октетах; возвращает нуль, если значением данного типа <code>OctetstringValue</code> является <code>omit</code> . |
| <code>void setLength(in TInteger len)</code> | <code>setLength</code> сначала переустанавливает этот тип <code>OctetstringValue</code> на его начальное значение, а затем устанавливает длину этого типа <code>OctetstringValue</code> в октетах на <code>len</code> . |

7.2.2.2.10 Абстрактный тип данных `HexstringValue`

Абстрактный тип данных `HexstringValue` базируется на абстрактном типе данных `Value`. Он представляет значения типа `hexstring` TTCN-3.

В отношении абстрактного типа данных `HexstringValue` определяются следующие операции:

| | |
|---|---|
| <code>TString getString()</code> | Возвращает текстовое представление данного типа <code>HexstringValue</code> , как определено в TTCN-3. Например, текстовым представлением для <code>0xAFFEE</code> является "AFFEE"Н. Текстовым представлением пустого типа <code>hexstring</code> TTCN-3 является "", в то время как его длина равна нулю. |
| <code>void setString(in TString value)</code> | Устанавливает значение данного типа <code>HexstringValue</code> в соответствии с текстовым представлением, как определено значением <code>value</code> . Например, значением данного типа <code>HexstringValue</code> является <code>0xAFFEE</code> , если текстовым представлением для <code>value</code> является "AFFEE"Н. |
| <code>TChar getHex(in TInteger position)</code> | Возвращает значение (0..15) на позиции данной шестнадцатеричной строки TTCN-3. Позиция 0 обозначает первые шестнадцатеричные разряды шестнадцатеричной строки TTCN-3. Допустимые значения для позиции находятся в диапазоне от 0 до <code>length - 1</code> . |
| <code>void setHex(in TInteger position, in TInteger value)</code> | Устанавливает шестнадцатеричный разряд в позиции на значение (0..15). Позиция 0 обозначает первый октет в <code>hexstring</code> . Допустимые значения для позиции находятся в диапазоне от 0 до <code>length - 1</code> . |
| <code>TInteger getLength()</code> | Возвращает длину данного типа <code>HexstringValue</code> в октетах; возвращает нуль, если значением данного типа <code>HexstringValue</code> является <code>omit</code> . |
| <code>void setLength(in TInteger len)</code> | <code>setLength</code> сначала переустанавливает данный тип <code>HexstringValue</code> на его начальное значение, а затем устанавливает длину данного типа <code>HexstringValue</code> в шестнадцатеричных разрядах на <code>len</code> . |

7.2.2.2.11 Абстрактный тип данных `RecordValue`

Абстрактный тип данных `RecordValue` базируется на абстрактном типе данных `Value`. Он точно определяет, как получить и установить тип `record` TTCN-3. Тот же абстрактный тип данных применяется к значениям, классом типа которых является `SET`. Различие между `record` и `set` существенно только во время сопоставления.

В отношении абстрактного типа данных `RecordValue` определяются следующие операции:

| | |
|---|---|
| <code>Value getField(in TString fieldName)</code> | Возвращает значение поля, именуемого <code>fieldName</code> . Возвращаемое значение является общим абстрактным базовым типом <code>Value</code> , поскольку поле записи |
|---|---|

может иметь любой тип, который определяется в TTCN-3. Если это поле не может быть получено из записи, то возвращается особое значение null.

`void setField(in TString fieldName, in Value value)`

Устанавливает поле записи, именуемое `fieldName`, на `value`. Никаких предположений не делается относительно того, как поле хранится в записи. При внутренней реализации можно либо хранить ссылку на это значение, либо копировать данное значение. Резонно предположить, что указанное значение копируется. Поэтому следует полагать, что дальнейшие модификации `value` не будут отражаться в записи.

`TStringSeq getFieldNames()`

Возвращает последовательность строки имен поля, пустую последовательность, если запись не имеет полей.

7.2.2.2.12 Абстрактный тип данных RecordOfValue

Абстрактный тип данных `RecordOfValue` базируется на абстрактном типе данных `Value`. Он точно определяет, каким образом можно получить и установить элементы в типах `record of` TTCN-3. Тот же абстрактный тип данных применяется к значению, классом типа которого является `SET_OF`. Различие между `record of` и `set of` существенно только во время сопоставления.

В отношении абстрактного типа данных `RecordOfValue` определяются следующие операции:

`Value getField(in TInteger position)` Возвращает значение элемента `record of` на позицию `position`, если `position` находится между `zero` и `length - 1`; в противном случае возвращает значение `null`. Возвращаемым значением является общий абстрактный базовый тип `Value`, поскольку `record of` может иметь поля любого типа, определенного в TTCN-3.

`void setField(in TInteger position, in Value value)`

Устанавливает поле в позиции `position` на `value`. Если значение `position` больше, чем `(length - 1)`, то элемент `record of` расширяется, чтобы иметь длину `(position + 1)`. Элемент `record of` для элементов между исходной позицией на `length` и позицией `position - 1` устанавливается на `omit`. Никаких предположений не делается относительно того, как поле хранится в `record of`. При внутренней реализации можно либо хранить ссылку на это значение, либо копировать данное значение. Резонно допустить, что указанное значение копируется. Поэтому следует полагать, что дальнейшие модификации `value` не будут отражаться в `record of`.

`void appendField(in Value value)`

Добавляет значение в конце элемента `record of`, то есть на позицию `length`. Никаких предположений не делается относительно того, как поле хранится в `record of`. При внутренней реализации можно либо хранить ссылку на это значение, либо копировать данное значение. Резонно допустить, что указанное значение копируется. Поэтому следует полагать, что дальнейшие модификации `value` не будут отражаться в `record of`.

`Type getElementType()`

Возвращает `Type` элементов для данного элемента `record of`.

`TInteger getLength()`

Возвращает фактическую длину значения `record of`, возвращает нуль, если значением `record of` является `omit`.

`void setLength(in TInteger len)`

Устанавливает длину `record of` на `len`. Если значение `len` больше, чем исходная длина, то вновь созданные элементы имеют значение `omit`. Если значение `len` меньше или равно исходной длине, то эта операция игнорируется.

7.2.2.2.13 Абстрактный тип данных UnionValue

Абстрактный тип данных `UnionValue` базируется на абстрактном типе данных `Value`. Он точно определяет, каким образом можно получить и установить варианты в типе `union` TTCN-3. Элемент `anytype` TTCN-3 представлен элементом `UnionValue`, где классом типа для типа, полученного с помощью `getType()`, является `ANYTYPE`. Подробная информация о классах типов содержится в п. 7.2.2.1.

В отношении абстрактного типа данных `UnionValue` определяются следующие операции:

`Value getVariant(in TString variantName)`

Возвращает значение имени `variantName` объединения TTCN-3, если имя `variantName` равно результату операции `getPresentVariantName`; в

противном случае возвращает особое значение null. Имя `variantName` обозначает имя варианта объединения, как определено в TTCN-3.

`void setVariant(in TString variantName, in Value value)`

Устанавливает имя `variantName` объединения на `Value`. Если имя `variantName` не определяется для данного объединения, то эта операция игнорируется. Если было выбрано другое имя варианта, то взамен выбирается новый вариант.

`TString getPresentVariantName()`

Возвращает `String` (строку), представляющую выбранное в текущий момент имя варианта в данном объединении TTCN-3. Если никакой вариант не выбран, то возвращается особое значение null.

`TStringSeq getVariantNames()`

Возвращает последовательность строки имен вариантов, или возвращается особое значение null, если объединение не имеет полей. Если `UnionValue` представляет `anytype` TTCN-3, то есть классом типа для типа, полученного с помощью `getType()`, является `ANYTYPE`, то возвращаются все предопределенные и определяемые пользователем типы TTCN-3.

7.2.2.2.14 Абстрактный тип данных `EnumeratedValue`

Абстрактный тип данных `EnumeratedValue` базируется на абстрактном типе данных `Value`. Он определяет, каким образом можно получить и установить элемент `enumerated` TTCN-3.

В отношении абстрактного типа данных `EnumeratedValue` определяются следующие операции:

`TString getEnum()`

Возвращает идентификатор строки данного значения `EnumeratedValue`. Этот идентификатор равен идентификатору в спецификации TTCN-3.

`void setEnum(in TString enumValue)`

Устанавливает `enum` (перечисление) на `enumValue`. Если `enumValue` является недопустимым значением для этого перечисления, то данная операция игнорируется.

7.2.2.2.15 Абстрактный тип данных `VerdictValue`

Абстрактный тип данных `VerdictValue` базируется на абстрактном типе данных `Value`. Он определяет, каким образом можно получить и установить `verdict` TTCN-3.

В отношении абстрактного типа данных `VerdictValue` определяются следующие операции:

`TInteger getVerdict()`

Возвращает целочисленное значение для данного значения `VerdictValue`. Это целое число является одной из следующих констант: `ERROR`, `FAIL`, `INCONC`, `NONE`, `PASS`.

`void setVerdict(in TInteger verdict)`

Устанавливает это значение `VerdictValue` на `verdict`. Следует заметить, что значение `VerdictValue` может быть установлено на любой из упомянутых выше вердиктов в любое время. Значение `VerdictValue` не выполняет никаких вычислений вердикта, как определено в TTCN-3. Например, допустимо устанавливать `VerdictValue` сначала на `ERROR`, а затем на `PASS`.

7.2.2.2.16 Абстрактный тип данных `AddressValue`

В отношении базового абстрактного типа данных `AddressValue` определяются следующие операции. Конкретные представления этих операций определяются в соответствующих частях отображения языка:

`Value getAddress()`

Возвращает значение адреса, который больше не будет иметь класс типа `ADDRESS`, но скорее будет иметь фактический тип, используемый для адреса.

`void setAddress(in Value value)`

Устанавливает это значение адреса на `value`.

7.2.3 Абстрактные типы регистрации

7.2.3.1 Абстрактный тип данных `TciValueTemplate`

В отношении абстрактного типа данных `TciValueTemplate` определяются следующие операции. Конкретные представления этих операций определяются в соответствующих частях отображения языка:

`boolean isOmit()`

Возвращает `true`, если шаблон является шаблоном `omit`.

`boolean isAny()`

Возвращает `true`, если шаблон является шаблоном `any`.

`boolean isAnyOrOmit()`

Возвращает `true`, если шаблон является шаблоном `any` или `omit`.

`TString getTemplateDef()`

Возвращает определение этого шаблона.

7.2.3.2 Абстрактный тип данных TciNonValueTemplate

В отношении абстрактного типа данных TciNonValueTemplate определяются следующие операции. Конкретные представления этих операций определяются в соответствующих частях отображения языка:

| | |
|--------------------------|---|
| boolean isAny() | Возвращает true, если шаблон является шаблоном any. |
| boolean isAll() | Возвращает true, если шаблон является шаблоном all. |
| TString getTemplateDef() | Возвращает определение этого шаблона. |

7.2.3.3 Список значений и типы несоответствия

Для регистрации различий между значениями и шаблонами определяются и используются следующие абстрактные типы данных:

| | |
|------------------------|--|
| TciValueList | Значение элемента TciValueList является списком значений. |
| TciValueDifference | Значение элемента TciValueDifference является структурой, содержащей значение, шаблон и описание этого различия. |
| TciValueDifferenceList | Значение элемента TciValueDifferenceList является последовательностью различий в значениях. |

7.3 Операции интерфейса TCI

В данном разделе указываются операции, которые выполняемый модуль TTCN-3 предоставляет тестовой системе (*required operations – требуемые операции*), и функциональные возможности, которые тестовая система должна предоставлять выполняемому модулю TTCN-3 (*provided operations – предоставляемые операции*).

Термины "required" и "provided" отражают тот факт, что в настоящей Рекомендации определяются требования к выполняемому модулю TTCN-3 с точки зрения пользователя. Пользователь "запрашивает" от выполняемого модуля TTCN-3 предоставление определенных функциональных возможностей для построения полной тестовой системы на базе TTCN-3. Для выполнения своей задачи выполняемый модуль TTCN-3 должен информировать пользователя об определенных событиях, когда пользователь должен "предоставлять" эту возможность выполняемому модулю TTCN-3.

Все определения операций в данном разделе рассматриваются с использованием языка описания интерфейса (IDL). Конкретные отображения языка определяются в разделах 8 и 9. В Приложении В представлено альтернативное отображение на язык XML для интерфейса регистрации.

Для каждого вызова операции интерфейса TCI все параметры *in*, *inout* и *out*, перечисленные в отдельном определении операции, являются обязательными. Значение параметра *in* задается вызывающим объектом. Вызывающий объект указывает на направление вызова. Для операций в *требуемом* интерфейсе вызывающий объект является тестовой системой, в то время как вызываемый объект является выполняемым модулем TTCN-3. Для операций в *предоставляемом* интерфейсе вызывающий объект является выполняемым модулем TTCN-3, в то время как тестовая система является вызываемым объектом.

Подобным образом, значение параметра *out* задается вызываемым объектом. В случае параметра *inout* значение сначала задается вызывающим объектом, но может быть заменено вызываемым объектом на новое значение. Следует заметить, что хотя в TTCN-3 также используются параметры *in*, *inout* и *out* для определений сигнатуры, обозначения, используемые в спецификации на языке IDL интерфейса TCI, не связаны с обозначениями в спецификации TTCN-3.

В вызовах операций следует использовать зарезервированное значение для указания на отсутствие параметров. Зарезервированные значения для этих типов определяются в каждом отображении языка, и на них будут в дальнейшем ссылаться как на значение null.

Кроме того, значение null также будет использоваться для указания на невозможность выполнения определенной задачи.

Поскольку в данном разделе определяются только интерфейсы и не предлагаются конкретные реализации в отношении способов выполнения заданных функциональных возможностей, термин "объект" будет использоваться для идентификации части реализации тестовой системы, которая реализует этот интерфейс и выполняет запрашиваемые функции. Например, вызывающим объектом в операции tciSendConnected является объект TE, то есть часть реализации тестовой системы, которая предоставляет функциональные возможности TE.

Все функции в интерфейсе описываются путем использования следующего шаблона. Описания, которые нельзя применить к определенным операциям, устраняются.

| | |
|----------------------|--|
| Сигнатура | Сигнатура IDL |
| Параметры in | Описание данных, передаваемых в качестве параметров для операции от вызывающего объекта к вызываемому объекту. |
| Параметры out | Описание данных, передаваемых в качестве параметров для операции от вызываемого объекта к вызывающему объекту. |

| | |
|------------------------------|---|
| Параметры inout | Описание данных, передаваемых в качестве параметров для операции от вызывающего объекта к вызываемому объекту и от вызываемого объекта к вызывающему объекту. |
| Возвращаемое значение | Описание данных, возвращаемых от операции вызывающему объекту. |
| Ограничение | Описание любых ограничений для случаев, когда может быть вызвана операция. |
| Результат | Требуемое поведение вызываемого объекта до возможного возврата данной операции. |

7.3.1 Интерфейс TCI-TM

Интерфейс администрирования тестом в TCI (TCI-TM) описывает операции, которые требуются для реализации в выполняемом модуле TTCN-3, и операции, которые предоставляются объекту TE при реализации администрирования тестом (Рисунок 5).

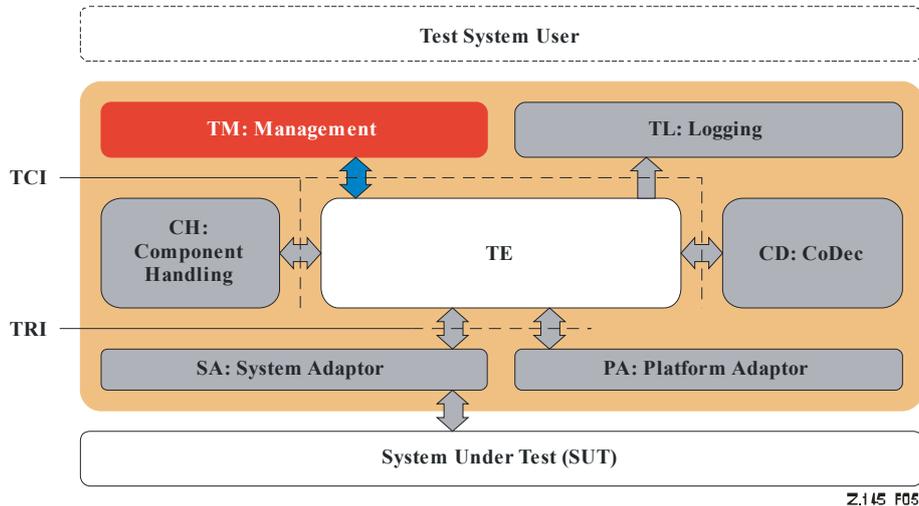


Рисунок 5/Z.145 – Интерфейс TCI-TM

Реализация администрирования тестом обеспечивает полное администрирование тестом для пользователя тестовой системы. При этом от модуля TE требуется наличие операций для запуска и останова выполнения теста модуля TTCN-3 или определенных тестовых примеров в модуле TTCN-3. В свою очередь, он предоставляет модулю TE операции для устранения параметра модуля на этапе выполнения и индикацию завершения выполнения.

В разделе 10 приведено описание, касающееся использования и последовательного упорядочения вызовов операций либо объектом TE, либо администрированием тестом.

7.3.1.1 Интерфейс TCI-TM required

В данном разделе указываются операции, которые объект TM запрашивает от TE. В дополнение к операциям, описанным в этом разделе, для администрирования тестом необходимы операции, требуемые в интерфейсе TCI-CD.

7.3.1.1.1 tciRootModule

| | | |
|------------------------------|--|---|
| Сигнатура | void tciRootModule (in TciModuleIdType moduleName) | |
| Параметры in | moduleName | moduleName обозначает идентификаторы модуля, как определено в TTCN-3. |
| Возвращаемое значение | void | |
| Ограничение | Используется, если только ни управляющая часть, ни тестовый пример не выполняются в текущий момент. | |
| Результат | tciRootModule выбирает указанный модуль для выполнения посредством последующего вызова, используя tciStartTestCase или tciStartControl. tciError будет генерироваться с помощью TE, если такой модуль не существует. | |

7.3.1.1.2 getImportedModules

| | |
|------------------------------|---|
| Сигнатура | TciModuleIdListType getImportedModules () |
| Возвращаемое значение | Список всех импортируемых модулей корневого модуля. Порядок следования этих модулей тот же, что и в модуле TTCN-3. Если нет импортируемых модулей, то возвращается пустой список модулей. |

| | |
|--------------------|---|
| Ограничение | Используется, если только ранее был установлен корневой модуль. |
| Результат | TE предоставляет для администрирования список импортируемых модулей корневого модуля. Если импортируемый модуль не существует, то возвращается пустой список модулей. Если TE не может предоставить список, то возвращается особое значение null. |

7.3.1.1.3 tciGetModuleParameters

| | | |
|------------------------------|--|--|
| Сигнатура | TciModuleParameterListType tciGetModuleParameters(in TciModuleIdType moduleName) | |
| Параметры in | moduleName | moduleName обозначает идентификаторы модуля, для которых должны быть найдены параметры модуля. |
| Возвращаемое значение | Список всех параметров модуля для идентифицируемого модуля. Порядок следования этих параметров тот же, что и в модуле TTCN-3. Если параметров нет, то возвращается пустой список параметров модуля. | |
| Ограничение | Используется, если только ранее был установлен корневой модуль. | |
| Результат | TE предоставляет для администрирования список параметров модуля для идентифицируемого модуля. Если параметров модуля нет, то возвращается пустой список параметров модуля. Если TE не может предоставить список, то возвращается особое значение null. | |

7.3.1.1.4 tciGetTestCases

| | | |
|------------------------------|--|--|
| Сигнатура | TciTestCaseIdListType tciGetTestCases() | |
| Возвращаемое значение | Список всех тестовых примеров, которые либо определяются в корневом модуле, либо импортируются в корневой модуль. | |
| Ограничение | Используется, если только ранее был установлен корневой модуль. | |
| Результат | TE предоставляет для администрирования список тестовых примеров. Если тестовые примеры не существуют, то возвращается пустой список тестовых примеров. Если TE не может предоставить список, то возвращается особое значение null. | |

7.3.1.1.5 tciGetTestCaseParameters

| | | |
|------------------------------|---|--|
| Сигнатура | TciParameterTypeListType tciGetTestCaseParameters(in TciTestCaseIdType testCaseId) | |
| Параметры in | testCaseId | Идентификатор тестового примера, как определено в модуле TTCN-3. |
| Возвращаемое значение | Список всех типов параметров данного тестового примера. Порядок следования этих типов параметров тот же, что и в сигнатуре TTCN-3 тестового примера. Если параметры не существуют, то возвращается пустой список типов параметров. | |
| Ограничение | Используется, если только ранее был установлен корневой модуль. | |
| Результат | TE предоставляет для администрирования список типов параметров данного тестового примера. Если параметры тестового примера не существуют, то возвращается пустой список типов параметров. Если TE не может предоставить список, то возвращается особое значение null. | |

7.3.1.1.6 tciGetTestCaseTSI

| | | |
|------------------------------|---|--|
| Сигнатура | TriPortIdListType tciGetTestCaseTSI(in TciTestCaseIdType testCaseId) | |
| Параметры in | testCaseId | Идентификатор тестового примера, как определено в модуле TTCN-3. |
| Возвращаемое значение | Список всех системных портов данного тестового примера, которые были объявлены в определении системного компонента для тестового примера, то есть порты интерфейса TSI. Если для тестового примера системный компонент не был определен явным образом, то тогда этот список содержит все порты связи тестового компонента MTC. Порядок следования этих портов тот же, что и в соответствующем объявлении типа компонента TTCN-3. Если системные порты не существуют, то возвращается пустой список, то есть список нулевой длины. | |
| Ограничение | Используется, если только ранее был установлен корневой модуль. | |
| Результат | TE предоставляет для администрирования список системных портов данного тестового примера. Если системные порты не существуют, то возвращается пустой список портов. Если TE не может предоставить список, то возвращается особое значение null. | |

7.3.1.1.7 tciStartTestCase

| | | |
|------------------------------|---|---|
| Сигнатура | void tciStartTestCase(in TciTestCaseIdType testCaseId, in TciParameterListType parameterList) | |
| Параметры in | testCaseId | Идентификатор тестового примера, как определено в модуле TTCN-3. |
| | parameterList | Список значений Values, где каждое значение определяет параметр из списка параметров, как указано в определении тестового примера TTCN-3. Порядок следования параметров в parameterList тот же, что и в сигнатуре TTCN-3 тестового примера. Если нет параметров, подлежащих передаче, тогда должны быть переданы либо значение null, либо пустой parameterList, то есть список нулевой длины. |
| Возвращаемое значение | void | |
| Ограничение | Требуется, если только ранее был выбран тот или иной модуль. Должны передаваться только идентификаторы testCaseIds для тестовых примеров, которые объявлены в выбранном в текущий момент модуле TTCN-3. Тестовые примеры, импортируемые в рассматриваемом модуле, не могут быть запущены. Для запуска импортируемых тестовых примеров сначала с помощью операции tciRootModule должен быть выбран модуль (импортируемый), на который ссылаются. | |
| Результат | tciStartTestCase запускает тестовый пример в выбранном в текущий момент времени модуле с данными параметрами. ТЕ будет генерировать tciError, если такого тестового примера не существует. Все параметры in и inout тестового примера в parameterList содержат Value. Все параметры out тестового примера в parameterList должны содержать особое значение null, поскольку они существенны только в случае завершения тестового примера. | |

7.3.1.1.8 tciStopTestCase

| | | |
|------------------------------|---|--|
| Сигнатура | void tciStopTestCase() | |
| Возвращаемое значение | void | |
| Ограничение | Требуется, если только ранее был выбран тот или иной модуль. | |
| Результат | tciStopTestCase останавливает выполняемый в текущий момент тестовый пример. Если ТЕ не выполняет тестовый пример, то данная операция будет игнорироваться. Если выполняется управляющая часть, то tciStopTestCase остановит прогон выполняемого в текущий момент тестового примера, то есть прогон тестового примера, на который только что было указано путем использования provided (предоставляемой) операции tciTestCaseStarted. Управляющая часть может продолжить выполнение, как если бы тестовый пример был остановлен обычным образом и возвращен с вердиктом ERROR. | |

7.3.1.1.9 tciStartControl

| | | |
|------------------------------|---|--|
| Сигнатура | TriComponentId tciStartControl() | |
| Возвращаемое значение | Идентификатор TriComponentId, который представляет тестовый компонент, выполняемый в управляющей части модуля. Если ТЕ не может запустить управляющую часть выбранного модуля, то будет возвращено особое значение null. | |
| Ограничение | Требуется, если только ранее был выбран тот или иной модуль. | |
| Результат | Запускает управляющую часть выбранного модуля. Управляющая часть будет запускать тестовые примеры TTCN-3, как описано в TTCN-3. При выполнении управляющей части ТЕ будет вызывать provided (предоставляемую) операцию tciTestCaseStarted и tciTestCaseTerminated для каждого тестового примера, который был запущен и завершен. После завершения управляющей части ТЕ будет вызывать provided (предоставляемую) операцию tciControlPartTerminated. | |

7.3.1.1.10 tciStopControl

| | | |
|------------------------------|---|--|
| Сигнатура | void tciStopControl() | |
| Возвращаемое значение | void | |
| Ограничение | Требуется, если только ранее был выбран тот или иной модуль. | |
| Результат | tciStopControl останавливает выполнение управляющей части. Если в текущий момент никакая управляющая часть не выполняется, то операция будет игнорироваться. Если непосредственно был запущен тестовый пример, то это остановит выполнение текущего тестового примера, как будто была вызвана операция tciStopTestCase. | |

7.3.1.2 Интерфейс TCI-TM provided

В данном разделе указываются операции, которые объект ТМ должен предоставлять объекту ТЕ.

7.3.1.2.1 tciTestCaseStarted

| | | |
|------------------------------|--|---|
| Сигнатура | void tciTestCaseStarted(in TciTestCaseIdType testCaseId, in TciParameterListType parameterList, in TFloat timer) | |
| Параметры in | testCaseId | Идентификатор тестового примера, как определено в модуле TTCN-3. |
| | parameterList | Список значений, являющихся частью сигнатуры тестового примера. Порядок следования параметров в списке parameterList тот же, что и в объявлении тестового примера TTCN-3. |
| | timer | Значение с плавающей запятой, представляющее период времени таймера тестового примера. |
| Возвращаемое значение | void | |
| Ограничение | Требуется только после того, как либо управляющая часть модуля, либо тестовый пример будут запущены с использованием <i>required</i> (требуемых) операций tciStartControl или tciStartTestCase. | |
| Результат | tciTestCaseStarted указывает объекту ТМ, что был запущен тестовый пример с идентификатором testCaseId. При этом нельзя будет различить, был ли запущен тестовый пример явным образом при использовании <i>required</i> (требуемой) операции tciStartTestCase или же он был запущен неявным образом при выполнении управляющей части. | |

7.3.1.2.2 tciTestCaseTerminated

| | | |
|------------------------------|--|---|
| Сигнатура | void tciTestCaseTerminated(in VerdictValue verdict, in TciParameterListType parameterList) | |
| Параметры in | verdict | Конечный вердикт тестового примера. |
| | parameterList | Список значений, являющихся частью сигнатуры тестового примера. Порядок следования параметров в списке parameterList тот же, что и в объявлении тестового примера TTCN-3. |
| Возвращаемое значение | void | |
| Ограничение | Требуется только после того, как либо управляющая часть модуля, либо тестовый пример будут запущены с использованием <i>required</i> (требуемых) операций tciStartControl или tciStartTestCase. | |
| Результат | Эта операция будет вызываться объектом ТЕ для указания администрированию тестом на то, что тестовый пример, который в данный момент выполнялся в компоненте МТС, завершен и что конечным вердиктом был verdict. При вызове операции tciTestCaseTerminated все параметры <i>out</i> и <i>inout</i> тестового примера содержат значения Values. Все параметры <i>in</i> тестового примера содержат особое значение null, поскольку они существенны только в случае запуска тестового примера, но не при ответе на вызов. | |

7.3.1.2.3 tciControlTerminated

| | | |
|------------------------------|---|--|
| Сигнатура | void tciControlTerminated() | |
| Возвращаемое значение | void | |
| Ограничение | Требуется только тогда, когда выполнение модуля было запущено с использованием операции tciStartControl. | |
| Результат | Эта операция будет вызываться объектом ТЕ для указания администрированию тестом на то, что управляющая часть выбранного модуля только что завершила выполнение. | |

7.3.1.2.4 tciGetModulePar

| | | |
|------------------------------|--|--|
| Сигнатура | Value tciGetModulePar(in TciModuleParameterIdType parameterId) | |
| Параметры in | parameterId | Идентификатор параметра модуля, как определено в TTCN-3. |
| Возвращаемое значение | Значение. | |
| Ограничение | Эта операция вызывается всякий раз, когда объекту ТЕ необходим доступ к значению параметра модуля. Для каждого параметра модуля, к которому осуществлен доступ, будет только один раз проведен анализ между парой операций tciStartTestCase и tciTestCaseTerminated, если тестовый пример был запущен явным образом, и между парой операций tciStartControl и tciControlTerminated, если была запущена управляющая часть модуля. | |
| Результат | При администрировании объекту ТЕ предоставляется Value для указанного идентификатора parameterId. Каждый вызов параметра tciGetModulePar() будет возвращать одно и то же значение во время выполнения запущенного явным образом тестового примера или во время выполнения управляющей части. Если администрирование не может предоставить значение TTCN-3, то тогда возвращается особое значение null. | |

7.3.1.2.5 tciError

| | | |
|------------------------------|---|---|
| Сигнатура | void tciError(in TString message) | |
| Параметры in | message | Значение строки, то есть ошибка сообщения |
| Возвращаемое значение | void | |
| Ограничение | Может потребоваться в любое время объектом ТЕ для указания на ситуацию с неисправимой ошибкой. На эту ситуацию с ошибкой может указывать либо СН или CD, либо эта ситуация может иметь место в ТЕ. | |
| Результат | ТЕ указывает на возникновение ситуации с неисправимой ошибкой. Сообщение message содержит фразу с указанием причины, которая может быть передана пользователю тестовой системы. Администрирование тестом должно завершить выполнение тестовых примеров или управляющих частей, если они функционируют. Администрирование тестом должно принять определенные меры по немедленному завершению выполнения теста. | |

7.3.2 Интерфейс TCI-CD

Интерфейс кодека в TCI (TCI-CD) описывает операции, которые требуется реализовать выполняемому модулю TTCN-3, и операции, которые предоставляются объекту ТЕ при реализации кодека для определенной схемы кодирования (Рисунок 6).

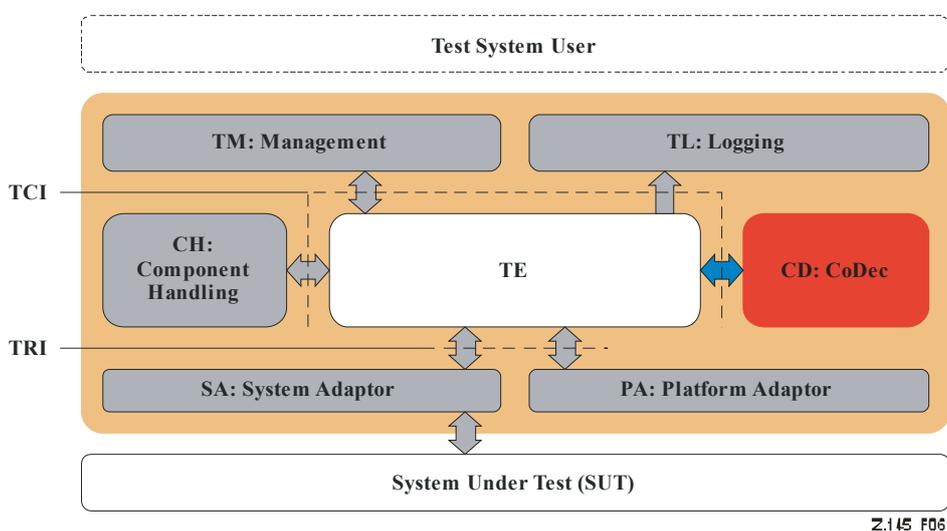


Рисунок 6/Z.145 – Интерфейс TCI-CD

При реализации кодека значения TTCN-3 кодируются согласно атрибуту кодирования в битовую строку, а битовая строка декодируется согласно гипотезе декодирования. Чтобы можно было декодировать битовую строку в значение TTCN-3, объект CD требует от ТЕ выполнения определенных функций. В свою очередь CD предоставляет выполняемому модулю TTCN-3 функциональные возможности кодирования и декодирования.

В разделе 10 приведено описание, касающееся использования и последовательного упорядочения вызовов операций либо объектом ТЕ, либо объектом CD.

7.3.2.1 Интерфейс TCI-CD required

В данном разделе указываются операции, которые объект CD запрашивает от ТЕ.

Все операции, описанные в этом разделе, также требуются в интерфейсах TCI-TM и TCI-CH.

7.3.2.1.1 getTypeForName

| | | |
|------------------------------|--|--|
| Сигнатура | Type getTypeForName(in TString typeName) | |
| Параметры in | typeName | Имя типа TTCN-3, как определено в модуле TTCN-3. Ниже приведены зарезервированные имена типов, которые будут возвращать предопределенный тип: "integer" "float" "bitstring" "hexstring" "octetstring" "charstring" "universal charstring" "boolean" "verdicttype" "objid" Имя typeName должно быть полностью классифицированным именем типа, то есть именем module.typeName |
| Возвращаемое значение | Тип, представляющий запрашиваемый тип TTCN-3. | |
| Ограничение | --- | |
| Результат | Возвращает тип, представляющий тип TTCN-3. Предопределенные типы TTCN-3 могут быть выведены из TE путем использования ключевых слов TTCN-3 для предопределенных типов. В этом случае typeName указывает на базовый тип TTCN-3, подобный "charstring", "bitstring" и т. д. Возвращает особое значение null, если запрашиваемый тип не может быть возвращен. Следует отметить, что anytype и address не могут быть получены модулем, установленным на null. И хотя это –предопределенные типы, они могут быть различными для модулей. Например, address может быть либо немодифицированным предопределенным типом, либо определяемым пользователем типом в модуле. Прочие предопределенные типы не могут быть заново определены. | |

7.3.2.1.2 getInteger

| | |
|------------------------------|---|
| Сигнатура | Type getInteger() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип integer TTCN-3. |
| Результат | Конструирует и возвращает базовый тип integer TTCN-3. |

7.3.2.1.3 getFloat

| | |
|------------------------------|---|
| Сигнатура | Type getFloat() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип float TTCN-3. |
| Результат | Конструирует и возвращает базовый тип float TTCN-3. |

7.3.2.1.4 getBoolean

| | |
|------------------------------|---|
| Сигнатура | Type getBoolean() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип boolean TTCN-3. |
| Результат | Конструирует и возвращает базовый тип boolean TTCN-3. |

7.3.2.1.5 getObjid

| | |
|------------------------------|---|
| Сигнатура | Type getObjid() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип object id TTCN-3. |
| Результат | Конструирует и возвращает базовый тип object id TTCN-3. |

7.3.2.1.6 getCharstring

| | |
|-----------------------|--|
| Сигнатура | Type getCharstring() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип charstring TTCN-3. |
| Результат | Конструирует и возвращает базовый тип charstring TTCN-3. |

7.3.2.1.7 getUniversalCharstring

| | |
|-----------------------|--|
| Сигнатура | Type getUniversalCharstring() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип universal charstring TTCN-3. |
| Результат | Конструирует и возвращает базовый тип universal charstring TTCN-3. |

7.3.2.1.8 getHexstring

| | |
|-----------------------|---|
| Сигнатура | Type getHexstring() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип hexstring TTCN-3. |
| Результат | Конструирует и возвращает базовый тип hexstring TTCN-3. |

7.3.2.1.9 getBitstring

| | |
|-----------------------|--|
| Сигнатура | Type getBitstring() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип bitstring TTCN -3. |
| Результат | Конструирует и возвращает базовый тип bitstring TTCN-3. |

7.3.2.1.10 getOctetstring

| | |
|-----------------------|---|
| Сигнатура | Type getOctetstring() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип octetstring TTCN-3. |
| Результат | Конструирует и возвращает базовый тип octetstring TTCN-3. |

7.3.2.1.11 getVerdict

| | |
|-----------------------|---|
| Сигнатура | Type getVerdict() |
| Возвращаемое значение | Экземпляр типа Type, представляющий тип verdict TTCN-3. |
| Результат | Конструирует и возвращает базовый тип verdict TTCN-3. |

7.3.2.1.12 tciErrorReq

| | |
|-----------------------|--|
| Сигнатура | void tciErrorReq(in TString message) |
| Параметры in | message Значение строки, то есть фраза об ошибке, описывающая проблему. |
| Возвращаемое значение | void |
| Ограничение | Требуется всякий раз, когда возникает ситуация с ошибкой. |
| Результат | TE будет уведомляться о ситуации с неустранимой ошибкой в CD, а индикация об ошибке будет передаваться администрированию тестом. |

7.3.2.2 Интерфейс TCI-CD provided

В этом разделе указываются операции, которые ТМ предоставляет объекту ТЕ.

7.3.2.2.1 decode (декодирование)

| | | |
|------------------------------|--|--|
| Сигнатура | Value decode(in TriMessageType message, in Type decodingHypothesis) | |
| Параметры in | message | Закодированное сообщение, подлежащее декодированию. |
| | decodingHypothesis | Гипотеза, на которой может базироваться декодирование. |
| Возвращаемое значение | Возвращает декодированное значение, если это значение является значением совместимого типа, как decodingHypothesis; иначе возвращает особое значение null. | |
| Ограничение | Эта операция вызывается всякий раз, когда ТЕ должен декодировать закодированное значение. ТЕ мог декодировать закодированное значение сразу же после его приема, либо мог из соображений выполнения отложить это декодирование до фактической выборки закодированного значения. | |
| Результат | Эта операция декодирует сообщение message согласно правилам кодирования и возвращает значение TTCN-3. Гипотеза decodingHypothesis используется для определения, может ли быть декодировано закодированное значение. Если правило кодирования не является самодостаточным, то есть, если закодированное сообщение по существу не содержит свой тип, то будет использована гипотеза decodingHypothesis. Если закодированное значение может быть декодировано без гипотезы декодирования, то будет возвращено особое значение null, если тип, определяемый из закодированного сообщения, несовместим с гипотезой декодирования. | |

7.3.2.2.2 encode (кодирование)

| | | |
|------------------------------|---|-----------------------------------|
| Сигнатура | TriMessageType encode(in Value value) | |
| Параметры in | value | Значение, подлежащее кодированию. |
| | | |
| Возвращаемое значение | Возвращает закодированное сообщение TriMessage для заданного правила кодирования. | |
| Ограничение | Эта операция вызывается всякий раз, когда объект ТЕ должен закодировать значение Value. | |
| Результат | Возвращает закодированное TriMessage согласно правилам кодирования. | |

7.3.3 Интерфейс TCI-CN

Интерфейс обработки компонента в TCI (TCI-CN) описывает операции, которые требуется реализовать выполняемому модулю TTCN-3, и операции, которые предоставляются объекту ТЕ при реализации обработки компонента (Рисунок 7).

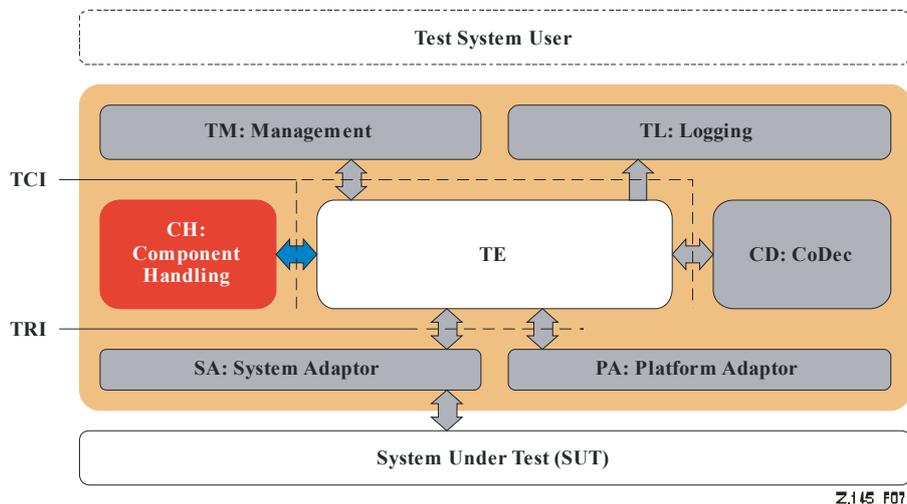


Рисунок 7/Z.145 – Интерфейс TCI-CN

При выполнении обработки компонента операции в рамках конфигурации TTCN-3, такие как create, connect и start, и операции межкомпонентной связи, такие как send распределяются в подсоединенном порте среди одного или нескольких выполняемых модулей TTCN-3, участвующих в том или ином сеансе тестирования. Следует заметить, что хотя в сеансе тестирования могло участвовать множество экземпляров объекта ТЕ, это не является обязательным.

Основной принцип заключается в том, что интерфейс TCI-CN не реализует любой вид функциональных возможностей TTCN-3. Вместо этого он будет получать информацию от ТЕ о том, что, например, создается тестовый компонент. На основе внутренней информации об обработке компонента (CH) на другой (удаленный) объект ТЕ, участвующий в сеансе тестирования, будет передан

запрос на создание тестового компонента. Этот второй (удаленный) участвующий объект ТЕ создаст компонент TTCN-3 и передаст обработку обратно запрашивающему (локальному) ТЕ. Запрашивающий (локальный) ТЕ теперь может выполнять операции над созданным тестовым компонентом через обработку этого компонента.

Термины "локальный ТЕ" и "удаленный ТЕ" используются в определениях операций для обозначения того факта, что реализация тестовой системы могла быть распределена по нескольким тестовым устройствам, в каждом из которых размещается полный ТЕ. Термины "локальный" и "удаленный" всегда указывают на описываемые в текущий момент интерфейсы. Для удобства термин "локальный" всегда указывает на ТЕ, который является либо принимающей стороной операции (для *требуемых* операций), либо вызывающей стороной операции (для *предоставляемых* операций). В то время как ТЕ концептуально считается распределяемым объектом, СН, как полагают, является нераспределяемым объектом. Этого можно достичь, применяя централизованную архитектуру, либо путем использования платформы программного обеспечения промежуточного слоя, которая абстрагируется от аспектов распределения. И хотя ТЕ может быть распределен по различным физическим устройствам, могут быть конфигурации, где только один, нераспределенный ТЕ будет участвовать в сеансе тестирования. В этом случае термины "локальный" и "удаленный" относятся к одному и тому же экземпляру ТЕ.

В разделе 10 приводится описание, касающееся использования и последовательного упорядочения вызовов операций со стороны либо ТЕ, либо СН.

И хотя все выполняемые модули TTCN-3, участвующие в сеансе тестирования, равны, имеется особый объект ТЕ*. Этот ТЕ* является объектом ТЕ, где был обработан явно заданный элемент `tciStartTestCase()` or `tciStartControl()`. Причиной для этого различия является то, что ТЕ* вычисляет глобальный вердикт. ТЕ* будет уведомлять администрирование теста о завершении выполнения теста, а затем предоставляет глобальный вердикт тестового примера.

7.3.3.1 Интерфейс TCI-CH required

В данном разделе описываются операции, которые требуются СН от объекта ТЕ. В дополнение к операциям, приведенным в данном разделе, также необходимы все *требуемые* операции интерфейса TCI-CD.

7.3.3.1.1 tciEnqueueMsgConnected

| | | |
|------------------------------|--|--|
| Сигнатура | void tciEnqueueMsgConnected(in TriPortIdType sender, in TriComponentIdType receiver, in Value rcvdMessage) | |
| Параметры in | sender | Идентификатор порта в компоненте передачи, через который посылается сообщение. |
| | receiver | Идентификатор принимающего компонента. |
| | rcvdMessage | Значение, подлежащее постановке в очередь. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, когда в удаленном ТЕ была вызвана <i>предоставляемая</i> операция <code>tciSendConnected</code> . | |
| Результат | ТЕ заносит полученное значение в очередь локального порта указанного компонента получателя. | |

7.3.3.1.2 tciEnqueueCallConnected

| | | |
|------------------------------|--|---|
| Сигнатура | void tciEnqueueCallConnected(in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList) | |
| Параметры in | sender | Идентификатор порта в передающем компоненте, через который посылается сообщение. |
| | receiver | Идентификатор принимающего компонента. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | parameterList | Список параметров значения, являющихся частью указанной сигнатуры. Порядок следования параметров в списке <code>parameterList</code> тот же, что и в объявлении сигнатуры TTCN-3. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, когда в удаленном ТЕ была вызвана <i>предоставляемая</i> операция <code>tciCallConnected</code> . Все параметры процедур <i>in</i> и <i>inout</i> содержат значения. Все параметры процедур <i>out</i> содержат особое значение <code>null</code> , поскольку они существенны только в ответе на вызов процедуры, но не в самом вызове процедуры. Параметры процедуры являются параметрами, заданными в шаблоне сигнатуры TTCN-3. | |
| Результат | ТЕ помещает вызовы в очередь локального порта указанного компонента получателя. | |

7.3.3.1.3 tciEnqueueReplyConnected

| | | |
|------------------------------|---|--|
| Сигнатура | void tciEnqueueReplyConnected(in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue) | |
| Параметры in | sender | Идентификатор порта, передающего ответ. |
| | receiver | Идентификатор компонента, принимающего ответ. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | parameterList | Список параметров значения, являющихся частью указанной сигнатуры. Порядок следования параметров в списке parameterList тот же, что и в объявлении сигнатуры TTCN-3. |
| | returnValue | (Факультативное) возвращаемое значение вызова процедуры. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается объектом СН в локальном ТЕ, когда в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciReplyConnected. Все параметры процедур <i>out</i> и <i>inout</i> и возвращаемое значение содержат значения. Все параметры процедур <i>in</i> должны содержать особое значение null, поскольку они существенны только для вызова процедуры, но не в случае ответа на этот вызов. Список parameterList содержит параметры вызова процедуры. Это параметры, описанные в шаблоне сигнатуры TTCN-3. Если в последовательности ATS TTCN-3 не был определен возвращаемый тип для сигнатуры процедуры, то для returnValue передается особое значение null. | |
| Результат | ТЕ заносит в очередь ответ локального порта указанного компонента получателя. | |

7.3.3.1.4 tciEnqueueRaiseConnected

| | | |
|------------------------------|--|---|
| Сигнатура | void tciEnqueueRaiseConnected(in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in Value exception) | |
| Параметры in | sender | Идентификатор порта, посылающего ответ. |
| | receiver | Идентификатор компонента, принимающего ответ. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | exception | Особое состояние. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если на удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciRaiseConnected. | |
| Результат | ТЕ заносит в очередь особое состояние локального порта указанного компонента получателя. | |

7.3.3.1.5 tciCreateTestComponent

| | | |
|------------------------------|---|---|
| Сигнатура | TriComponentIdType tciCreateTestComponent(in TciTestComponentKindType kind, in Type componentType), in TString name) | |
| Параметры in | kind | Вид компонента, который создается; это либо MTC, PTC, либо CONTROL. |
| | componentType | Идентификатор компонентного типа TTCN-3, который создается. |
| | name | Имя компонента, который создается. |
| Возвращаемое значение | Значение TriComponentIdType для создаваемого компонента. | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciCreateTestComponentReq. Тип componentType должен быть установлен на особое значение null, если создается тестовый компонент вида control. Элемент name должен быть установлен на особое значение null, если в операторе create TTCN-3 не задано имя. | |
| Результат | ТЕ создает тестовый компонент TTCN-3 типа componentType и передает ссылку TriComponentIdType назад к СН. СН передает эту ссылку обратно к удаленному ТЕ. | |

7.3.3.1.6 tciStartTestComponent

| | | |
|------------------------------|--|--|
| Сигнатура | void tciStartTestComponent(in TriComponentIdType component, in TciBehaviourIdType behaviour, in TciParameterListType parameterList) | |
| Параметры in | component | Идентификатор компонента, подлежащего запуску. Относится к идентификатору, ранее созданному вызовом операции tciCreateTestComponent |
| | behaviour | Идентификатор поведения, подлежащего запуску в компоненте. |
| | parameterList | Список значений, где каждое значение определяет параметр из этого списка параметров, как описано в объявлении функции TTCN-3 для запускаемой функции. Порядок следования параметров в списке parameterList тот же, что и в сигнатуре TTCN-3 тестового примера. Если не нужно пересылать параметры, то передаются либо значение null, либо пустой список parameterList, то есть список нулевой длины. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciStartTestComponentReq. Поскольку для запускаемых функций допустимы только параметры <i>in</i> (Рекомендация МСЭ-T Z.140 [2]), то список parameterList содержит только параметры <i>in</i> . | |
| Результат | ТЕ запускает указанное поведение в указанном компоненте. | |

7.3.3.1.7 tciStopTestComponent

| | | |
|------------------------------|--|---|
| Сигнатура | void tciStopTestComponent(in TriComponentIdType component) | |
| Параметры in | component | Идентификатор компонента, подлежащего останову. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciStopTestComponentReq. | |
| Результат | ТЕ останавливает указанное поведение в указанном компоненте. | |

7.3.3.1.8 tciConnect

| | | |
|------------------------------|---|--|
| Сигнатура | void tciConnect(in TriPortIdType fromPort, in TriPortIdType toPort) | |
| Параметры in | fromPort | Идентификатор порта тестового компонента, от которого должно выполняться соединение. |
| | toPort | Идентификатор порта, с которым должно выполняться соединение. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciConnect. | |
| Результат | ТЕ соединяет указанные порты друг с другом. | |

7.3.3.1.9 tciDisconnect

| | | |
|------------------------------|--|---|
| Сигнатура | void tciDisconnect(in TriPortIdType fromPort, in TriPortIdType toPort) | |
| Параметры in | fromPort | Идентификатор порта тестового компонента, подлежащего разъединению. |
| | toPort | Идентификатор порта тестового компонента, подлежащего разъединению. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciDisconnect. | |
| Результат | ТЕ разъединяет указанные порты. | |

7.3.3.1.10 tciMap

| | | |
|------------------------------|--|---|
| Сигнатура | void tciMap(in TriPortIdType fromPort, in TriPortIdType toPort) | |
| Параметры in | fromPort | Идентификатор порта тестового компонента, подлежащего отображению. |
| | toPort | Идентификатор порта тестового компонента, на который должно быть выполнено отображение. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciMapReq. | |
| Результат | ТЕ отображает указанные порты друг на друга. | |

7.3.3.1.11 tciUnmap

| | | |
|------------------------------|--|---|
| Сигнатура | void tciUnmap(in TriPortIdType fromPort, in TriPortIdType toPort) | |
| Параметры in | fromPort | Идентификатор порта тестового компонента, не подлежащего отображению. |
| | toPort | Идентификатор порта тестового компонента, не подлежащего отображению. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciUnmapReq. | |
| Результат | ТЕ не отображает указанные порты. | |

7.3.3.1.12 tciTestComponentTerminated

| | | |
|------------------------------|--|---|
| Сигнатура | void tciTestComponentTerminated(in TriComponentIdType component, in VerdictValue verdict) | |
| Параметры in | component | Идентификатор компонента, который был завершен. |
| | verdict | Вердикт после завершения компонента. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciTestComponentTerminatedReq. | |
| Результат | Локальный ТЕ получает уведомление о завершении указанного тестового компонента в удаленном ТЕ. Поскольку выполняемая функция в тестовом компоненте может иметь только параметры <i>in</i> (Рекомендация МСЭ-Т Z.140 [2]), то операция tciTestComponentTerminated не имеет параметра parameterList. | |

7.3.3.1.13 tciTestComponentRunning

| | | |
|------------------------------|--|---|
| Сигнатура | TBoolean tciTestComponentRunning(in TriComponentIdType component) | |
| Параметры in | component | Идентификатор компонента, подлежащего проверке на предмет выполнения. |
| Возвращаемое значение | true, если указанный компонент все еще выполняет поведение; в противном случае false. | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciTestComponentRunningReq. | |
| Результат | Локальный ТЕ определяет, выполняется ли указанным компонентом тестовое поведение. Если этот компонент выполняет поведение, то будет возвращено значение true. В любом другом случае, например, когда тестовый компонент закончил выполнение или не был запущен и т. д., будет возвращено значение false. После выполнения операцией возврата СН передаст это значение назад к удаленному ТЕ. | |

7.3.3.1.14 tciTestComponentDone

| | | |
|------------------------------|--|---|
| Сигнатура | TBoolean tciTestComponentDone(in TriComponentIdType comp) | |
| Параметры in | comp | Идентификатор компонента, подлежащего проверке на предмет завершенности выполнения. |
| Возвращаемое значение | true, если указанный компонент завершил выполнение своего поведения; в противном случае false. | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciTestComponentDoneReq. | |
| Результат | Локальный ТЕ определяет, завершил ли указанный компонент выполнение своего тестового поведения. Если этот компонент завершил свое поведение, то будет возвращено значение true. В любом другом случае, например, когда тестовый компонент не запускался или он все еще выполняется, будет возвращено значение false. После выполнения операцией возврата СН передаст это значение назад к удаленному ТЕ. | |

7.3.3.1.15 tciGetMTC

| | | |
|------------------------------|--|--|
| Сигнатура | TriComponentIdType tciGetMTC() | |
| Возвращаемое значение | Значение TriComponentIdType компонента МТС, если МТС выполняется в локальном ТЕ; в противном случае особое значение null. | |
| Ограничение | СН может вызывать эту операцию в соответствующем локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciGetMTCReq. | |
| Результат | Локальный ТЕ определяет, выполняется ли компонент МТС в локальном ТЕ. Если МТС выполняется в локальном ТЕ, то возвращается идентификатор компонента этого МТС. Если МТС не выполняется в локальном ТЕ, то будет возвращено особое значение null. Данная операция не будет влиять на выполнение компонента МТС. После выполнения операцией возврата СН передаст это значение назад к удаленному ТЕ. | |

7.3.3.1.16 tciExecuteTestCase

| | | |
|------------------------------|--|---|
| Сигнатура | void tciExecuteTestCase(in TciTestCaseIdType testCaseId, in TriPortIdListType tsiPortList) | |
| Параметры in | testCaseId | Идентификатор тестового примера, как определено в модуле TTCN-3. |
| | tsiPortList | Содержит все порты, которые были объявлены в определении компонента системы для тестового примера, то есть порты интерфейса TSI. Если компонент системы не был определен явным образом для тестового примера, тогда tsiPortList содержит все порты связи компонента МТС. Порядок следования портов в списке tsiPortList тот же, что и в соответствующем объявлении типа компонента TTCN-3. Если не должны передаваться никакие порты, то пересылается либо значение null, либо пустой список tsiPortList, то есть список нулевой длины. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в соответствующем локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciExecuteTestCaseReq. | |
| Результат | Локальный ТЕ определяет, должны ли быть выполнены статические соединения с системой SUT и инициализация средств связи для портов интерфейса TSI. | |

7.3.3.1.17 tciReset

| | | |
|------------------------------|--|--|
| Сигнатура | void tciReset() | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в соответствующих локальных ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciResetReq. | |
| Результат | ТЕ может решить принять все меры для локальной переустановки тестовой системы. | |

7.3.3.1.18 tciKillTestComponent

| | | |
|------------------------------|--|---|
| Сигнатура | void tciKillTestComponent(in TriComponentIdType comp) | |
| Параметры in | comp | Идентификатор компонента, подлежащего устранению. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciKillTestComponentReq. | |
| Результат | ТЕ останавливает поведение в указанном компоненте, если это необходимо, и переводит его в устранимое состояние. | |

7.3.3.1.19 tciTestComponentAlive

| | | |
|------------------------------|---|---|
| Сигнатура | TBoolean tciTestComponentAlive(in TriComponentIdType comp) | |
| Параметры in | comp | Идентификатор компонента, подлежащего проверке на предмет функционирования. |
| Возвращаемое значение | true, если указанный компонент функционирует; в противном случае false. | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciTestComponentAliveReq. | |
| Результат | Локальный ТЕ определяет, функционирует ли указанный компонент. После выполнения операций возврата СН передаст это значение назад к удаленному ТЕ. | |

7.3.3.1.20 tciTestComponentKilled

| | | |
|------------------------------|--|---|
| Сигнатура | TBoolean tciTestComponentKilled(in TriComponentIdType comp) | |
| Параметры in | comp | Идентификатор компонента, подлежащего проверке на предмет устранения. |
| Возвращаемое значение | true, если указанный компонент был устранен; в противном случае false. | |
| Ограничение | Эта операция вызывается СН в локальном ТЕ, если в удаленном ТЕ была вызвана <i>предоставляемая</i> операция tciTestComponentKilledReq. | |
| Результат | Локальный ТЕ определяет, находится ли указанный компонент в состоянии "устранен". Если находится, то будет возвращено значение true. В любом другом случае будет возвращено значение false. После выполнения операций возврата СН передаст это значение назад к удаленному ТЕ. | |

7.3.3.2 Интерфейс TCI-СН provided

В данном разделе определяются операции, которые СН предоставляет для ТЕ.

7.3.3.2.1 tciSendConnected

| | | |
|------------------------------|--|--|
| Сигнатура | void tciSendConnected(in TriPortIdType sender, in TriComponentIdType receiver, in Value sendMessage) | |
| Параметры in | sender | Идентификатор порта в передающем компоненте, через который передается сообщение. |
| | receiver | Идентификатор принимающего компонента. |
| | sendMessage | Сообщение, подлежащее передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет одноадресную операцию передачи TTCN-3 на порте компонента, который был соединен с другим портом компонента. | |
| Результат | Осуществляет асинхронную передачу только к данному компоненту получателя. СН передает сообщение к удаленному ТЕ, в котором получателем реализуется его функция, и заносит в очередь данные в удаленном ТЕ. | |

7.3.3.2.2 tciSendConnectedBC

| | | |
|------------------------------|---|--|
| Сигнатура | void tciSendConnectedBC(in TriPortIdType sender, in Value sendMessage) | |
| Параметры in | sender | Идентификатор порта в передающем компоненте, через который передается сообщение. |
| | sendMessage | Сообщение, подлежащее передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет широковещательную операцию передачи TTCN-3 на порте компонента, который соединен с другими портами компонента. | |
| Результат | Осуществляет асинхронную передачу ко всем компонентам, соединенным с данным портом. СН передает сообщение ко всем удаленным ТЕ, в которых получателем реализуется его функция, и заносит в очередь данные в удаленных ТЕ. | |

7.3.3.2.3 tciSendConnectedMC

| | | |
|------------------------------|--|--|
| Сигнатура | void tciSendConnectedMC(in TriPortIdType sender, in TriComponentIdListType receivers, in Value sendMessage) | |
| Параметры in | sender | Идентификатор порта в передающем компоненте, через который передается сообщение. |
| | receivers | Идентификаторы принимающих компонентов. |
| | sendMessage | Сообщение, подлежащее передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет многоадресную операцию передачи TTCN-3 на порте компонента, который был соединен с другими портами компонента. | |
| Результат | Осуществляет асинхронную передачу ко всем данным компонентам получателя. СН передает сообщение ко всем удаленным ТЕ, в которых получателями выполняются их функции, и заносит в очередь данные в удаленных ТЕ. | |

7.3.3.2.4 tciCallConnected

| | | |
|------------------------------|---|--|
| Сигнатура | void tciCallConnected(in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList) | |
| Параметры in | sender | Идентификатор порта в передающем компоненте, через который передается сообщение. |
| | receiver | Идентификатор принимающего компонента. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | parameterList | Список параметров значения, являющихся частью указанной сигнатуры. Параметры в списке parameterList следуют в том же порядке, что и в объявлении сигнатуры TTCN-3. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет одноадресную операцию вызова TTCN-3 на порте компонента, который был соединен с другим портом компонента. Все параметры процедуры <i>in</i> и <i>inout</i> содержат значения. Все параметры процедуры <i>out</i> содержат особое значение null, поскольку они существенны только в ответе на вызов процедуры, но не в самом вызове процедуры. Параметрами процедуры являются параметры, заданные в шаблоне сигнатуры TTCN-3. | |
| Результат | При вызове данной операции ТЕ может инициировать вызов процедуры, соответствующий идентификатору сигнатуры signature в вызываемом компоненте получателя. Операция tciCallConnected возвращает значение без ожидания возврата сгенерированного вызова процедуры. Следует заметить, что факультативное значение тайм-аута, которое может быть задано в последовательности ATS TTCN-3 для операции вызова, не включено в сигнатуру операции tciCallConnected. ТЕ отвечает за адресацию этого генерирования путем запуска таймера для операции вызова TTCN-3 в РА с отдельным вызовом операции интерфейса TRI, то есть triStartTimer. СН передает вызов удаленному ТЕ, в котором элементом receiver выполняются его функции, и ставит в очередь вызов в удаленном ТЕ. | |

7.3.3.2.5 tciCallConnectedBC

| | | |
|------------------------------|--|--|
| Сигнатура | void tciCallConnectedBC(in TriPortIdType sender, in TriSignatureIdType signature, in TciParameterListType parameterList) | |
| Параметры in | sender | Идентификатор порта в передающем компоненте, через который передается сообщение. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | parameterList | Список параметров значения, являющихся частью указанной сигнатуры. Порядок следования параметров в списке parameterList тот же, что и в объявлении сигнатуры TTCN-3. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет широкоадресную операцию вызова в TTCN-3 на порте компонента, который был соединен с другими портами компонента. Все параметры процедуры <i>in</i> и <i>inout</i> содержат значения. Все параметры процедуры <i>out</i> будут содержать особое значение null, поскольку они существенны только в ответе на вызов процедуры, но не в самом вызове процедуры. Параметрами процедуры являются параметры, заданные в шаблоне сигнатуры TTCN-3. | |
| Результат | При вызове этой операции ТЕ может инициировать вызов процедуры, соответствующий идентификатору сигнатуры signature у получателя вызываемого компонента. Операция tciCallConnected возвращает значение, не ожидая возврата сгенерированного вызова процедуры. Следует заметить, что факультативное значение тайм-аута, которое может быть задано в последовательности ATS TTCN-3 для операции вызова, не включено в сигнатуру операции tciCallConnected. ТЕ отвечает за адресацию этого генерирования путем запуска таймера для операции вызова TTCN-3 в РА с отдельным вызовом операции интерфейса TRI, то есть triStartTimer. СН передает вызов ко всем удаленным ТЕ, в которых элементом receiver выполняются его функции, и заносит в очередь вызов в удаленных ТЕ. | |

7.3.3.2.6 tciCallConnectedMC

| | | |
|------------------------------|---|--|
| Сигнатура | void tciCallConnectedMC(in TriPortIdType sender, in TriComponentIdListType receivers, in TriSignatureIdType signature, in TciParameterListType parameterList) | |
| Параметры in | sender | Идентификатор порта в передающем компоненте, через который посылается сообщение. |
| | receivers | Идентификатор принимающих компонентов. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | parameterList | Список параметров значения, являющихся частью указанной сигнатуры. Порядок следования параметров в списке parameterList тот же, что и в объявлении сигнатуры TTCN-3. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию многоадресного вызова TTCN-3 на порте компонента, который был соединен с другими портами компонента. Все параметры процедуры <i>in</i> и <i>inout</i> содержат значения. Все параметры процедуры <i>out</i> содержат особое значение null, поскольку они существенны только в ответе на вызов процедуры, но не в самом вызове процедуры. Параметрами процедуры являются параметры, заданные в шаблоне сигнатуры TTCN-3. | |
| Результат | При вызове данной операции ТЕ может инициировать вызов процедуры, соответствующий идентификатору сигнатуры signature у получателя вызываемого компонента. Операция tciCallConnected возвращает значение, не ожидая возврата сгенерированного вызова процедуры. Следует заметить, что факультативное значение тайм-аута, которое может быть задано в последовательности ATS TTCN-3 для операции вызова, не включено в сигнатуру операции tciCallConnected. ТЕ отвечает за адресацию этого генерирования путем запуска таймера для операции вызова TTCN-3 в адаптере РА с отдельным вызовом операции интерфейса TRI, то есть triStartTimer. СН передает вызов ко всем удаленным ТЕ, в которых элементом receiver выполняются его функции, и заносит в очередь вызов в удаленных ТЕ. | |

7.3.3.2.7 tciReplyConnected

| | | |
|------------------------------|--|--|
| Сигнатура | void tciReplyConnected(in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue) | |
| Параметры in | sender | Идентификатор порта, передающего ответ. |
| | receiver | Идентификатор порта, принимающего ответ. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | parameterList | Список закодированных параметров, являющихся частью указанной сигнатуры. Порядок следования параметров в списке parameterList тот же, что и в объявлении сигнатуры TTCN-3. |
| | returnValue | (Факультативное) возвращаемое значение вызова процедуры. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, когда он выполняет операцию одноадресного ответа TTCN-3 на порте компонента, который был соединен с другим портом компонента. Все параметры процедуры <i>out</i> и <i>inout</i> и возвращаемое значение содержат значения. Все параметры процедуры <i>in</i> содержат особое значение null, поскольку они существенны только для вызова процедуры, но не в ответе на вызов. Список parameterList содержит параметры вызова процедуры. Эти параметры являются параметрами, заданными в шаблоне сигнатуры TTCN-3. Если возвращаемый тип не был определен для сигнатуры процедуры в последовательности ATS TTCN-3, то для возвращаемого значения передается особое значение null. | |
| Результат | При вызове данной операции СН может генерировать ответ на вызов процедуры, соответствующий идентификатору сигнатуры signature и идентификатору компонента receiver. СН передает ответ удаленному ТЕ, в котором элементом receiver выполняются его функции, и заносит в очередь ответ в удаленном ТЕ. | |

7.3.3.2.8 tciReplyConnectedBC

| | | |
|------------------------------|---|--|
| Сигнатура | void tciReplyConnectedBC(in TriPortIdType sender, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue) | |
| Параметры in | sender | Идентификатор порта, передающего ответ. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | parameterList | Список закодированных параметров, являющихся частью указанной сигнатуры. Порядок следования параметров в списке parameterList тот же, что и в объявлении сигнатуры TTCN-3. |
| | returnValue | (Факультативное) возвращаемое значение вызова процедуры. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию широковещательного ответа TTCN-3 в порте компонента, который был соединен с другими портами компонента. Все параметры процедуры <i>out</i> и <i>inout</i> и возвращаемое значение содержат значения. Все параметры процедуры <i>in</i> содержат особое значение null, поскольку они существенны только для вызова процедуры, но не в ответе на этот вызов. Список parameterList содержит параметры вызова процедуры. Эти параметры являются параметрами, заданными в шаблоне сигнатуры TTCN-3. Если возвращаемый тип не был определен для сигнатуры процедуры в последовательности ATS TTCN-3, то для возвращаемого значения передается особое значение null. | |
| Результат | При вызове этой операции СН может сгенерировать ответ на вызов процедуры, соответствующий идентификатору сигнатуры signature и всем компонентам, соединенным с элементом sender. СН передает особое состояние ко всем удаленным ТЕ, в которых получателями выполняются их функции, и заносит в очередь особое состояние в удаленных ТЕ. | |

7.3.3.2.9 tciReplyConnectedMC

| | | |
|------------------------------|--|--|
| Сигнатура | void tciReplyConnectedMC(in TriPortIdType sender, in TriComponentIdListType receivers, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue) | |
| Параметры in | sender | Идентификатор порта, передающего ответ. |
| | receivers | Идентификатор компонентов, принимающих ответ. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | parameterList | Список закодированных параметров, являющихся частью указанной сигнатуры. Порядок следования параметров в списке parameterList тот же, что и в объявлении сигнатуры TTCN-3. |
| | returnValue | (Факультативное) возвращаемое значение вызова процедуры. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию многоадресного ответа TTCN-3 на порте компонента, который был соединен с другими портами компонента. Все параметры процедуры <i>out</i> и <i>inout</i> и возвращаемое значение содержат значения. Все параметры процедуры <i>in</i> будут содержать особое значение null, поскольку они существенны только для вызова процедуры, но не в ответе на вызов. Список parameterList содержит параметры вызова процедуры. Эти параметры являются параметрами, заданными в шаблоне сигнатуры TTCN-3. Если возвращаемый тип не был определен для сигнатуры процедуры в последовательности ATS TTCN-3, то для возвращаемого значения передается особое значение null. | |
| Результат | При вызове данной операции СН может сгенерировать ответ на вызов процедуры, соответствующий идентификатору сигнатуры signature и одному из идентификаторов компонента в элементе receivers. СН передает ответ к удаленным ТЕ, в которых элементом receivers выполняются его функции, и заносит в очередь ответ в удаленных ТЕ. | |

7.3.3.2.10 tciRaiseConnected

| | | |
|------------------------------|---|---|
| Сигнатура | void tciRaiseConnected(in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in Value exception) | |
| Параметры in | sender | Идентификатор порта, передающего ответ. |
| | receiver | Идентификатор компонента, принимающего ответ. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | exception | Значение особого состояния. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет одноадресную операцию порождения особого состояния TTCN-3 на порте компонента, который был соединен с другим портом компонента. | |
| Результат | При вызове этой операции СН может породить особое состояние для вызова процедуры, соответствующего идентификатору сигнатуры <i>signature</i> и идентификатору компонента <i>receiver</i> . СН передает особое состояние удаленному ТЕ, в котором элементом <i>receiver</i> выполняются его функции, и заносит в очередь особое состояние в удаленном ТЕ. | |

7.3.3.2.11 tciRaiseConnectedBC

| | | |
|------------------------------|--|---|
| Сигнатура | void tciRaiseConnectedBC(in TriPortIdType sender, in TriSignatureIdType signature, in Value exception) | |
| Параметры in | sender | Идентификатор порта, передающего ответ. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | exception | Значение особого состояния. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет широковещательную операцию порождения особого состояния TTCN-3 на порте компонента, который был соединен с другими портами компонента. | |
| Результат | При вызове этой операции СН может породить особое состояние для вызова процедуры, соответствующего идентификатору сигнатуры <i>signature</i> и всем компонентам, соединенным с элементом <i>sender</i> . СН передает особое состояние всем удаленным ТЕ, в которых получателями выполняются их функции, и заносит в очередь особое состояние в удаленных ТЕ. | |

7.3.3.2.12 tciRaiseConnectedMC

| | | |
|------------------------------|---|--|
| Сигнатура | void tciRaiseConnectedMC(in TriPortIdType sender, in TriComponentIdListType receiver, in TriSignatureIdType signature, in Value exception) | |
| Параметры in | sender | Идентификатор порта, передающего ответ. |
| | receivers | Идентификаторы компонента, принимающего ответ. |
| | signature | Идентификатор сигнатуры вызова процедуры. |
| | exception | Значение особого состояния. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет многоадресную операцию порождения особого состояния TTCN-3 на порте компонента, который был соединен с другим портом компонента. | |
| Результат | При вызове данной операции СН может породить особое состояние для вызова процедуры, соответствующего идентификатору сигнатуры <i>signature</i> и одному из идентификаторов компонента <i>receivers</i> . СН передает это особое состояние всем удаленным ТЕ, в которых элементами <i>receivers</i> выполняются их функции, и заносит в очередь особое состояние в удаленных ТЕ. | |

7.3.3.2.13 tciCreateTestComponentReq

| | | |
|------------------------------|---|---|
| Сигнатура | TriComponentIdType tciCreateTestComponentReq(in TciTestComponentKindType kind, in Type componentType, in TString name) | |
| Параметры in | kind | Вид компонента, который создается; это либо MTC, PTC, либо CONTROL. |
| | componentType | Идентификатор типа компонента TTCN-3, который создается. |
| Возвращаемое значение | Значение TriComponentIdType для создаваемого компонента. | |
| Ограничение | Эта операция вызывается TE, если должен быть создан компонент; либо явным образом, когда вызывается операция создания TTCN-3, либо неявным образом, когда должен быть создан главный тестовый компонент (MTC) или управляющий компонент. Элемент name должен быть установлен на особое значение null, если в операторе создания TTCN-3 имя не задано. | |
| Результат | СН передает удаленному TE запрос на создание компонента и вызывает там операцию tciCreateTestComponent, чтобы получить идентификатор компонента для данного компонента. | |

7.3.3.2.14 tciStartTestComponentReq

| | | |
|------------------------------|--|--|
| Сигнатура | void tciStartTestComponentReq(in TriComponentIdType component, in TciBehaviourIdType behaviour, in TciParameterListType parameterList) | |
| Параметры in | component | Идентификатор компонента, подлежащего запуску. |
| | behaviour | Идентификатор поведения, подлежащего запуску в компоненте. |
| | parameterList | Список элементов Values, где каждое значение определяет параметр из списка параметров, как описано в объявлении функции TTCN-3 для запускаемой функции. Порядок следования этих параметров в списке parameterList тот же, что и в сигнатуре TTCN-3 тестового примера. Если никакие параметры не должны передаваться, то пересылается либо значение null, либо пустой список parameterList, то есть список нулевой длины. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается TE, если он выполняет операцию запуска TTCN-3. Поскольку для запускаемых функций допустимыми являются только параметры in (Рекомендация МСЭ-Т Z.140 [2]), список parameterList содержит только параметры in. | |
| Результат | СН передает удаленному TE запрос на запуск компонента и вызывает там операцию tciStartTestComponent. | |

7.3.3.2.15 tciStopTestComponentReq

| | | |
|------------------------------|--|---|
| Сигнатура | void tciStopTestComponentReq(in TriComponentIdType component) | |
| Параметры in | component | Идентификатор компонента, подлежащего останову. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается TE, если он выполняет операцию останова TTCN-3. | |
| Результат | СН передает удаленному TE запрос на останов компонента и вызывает там операцию tciStopTestComponent. | |

7.3.3.2.16 tciConnectReq

| | | |
|------------------------------|---|--|
| Сигнатура | void tciConnectReq(in TriPortIdType fromPort, in TriPortIdType toPort) | |
| Параметры in | fromPort | Идентификатор порта тестового компонента, от которого должно выполняться соединение. |
| | toPort | Идентификатор порта тестового компонента, с которым должно выполняться соединение. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается TE, если он выполняет операцию соединения TTCN-3. | |
| Результат | СН передает удаленному TE запрос на соединение, где он вызывает операцию tciConnect, чтобы установить логическое соединение между двумя указанными портами. Следует заметить, что оба порта могут быть в удаленных TE. В этом случае операция возвращает значение только после вызова операции tciConnect в обоих удаленных TE. | |

7.3.3.2.17 tciDisconnectReq

| | | |
|------------------------------|--|---|
| Сигнатура | void tciDisconnectReq(in TriPortIdType fromPort, in TriPortIdType toPort) | |
| Параметры in | fromPort | Идентификатор порта тестового компонента, подлежащего разъединению. |
| | toPort | Идентификатор порта тестового компонента, подлежащего разъединению. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию разъединения TTCN-3. | |
| Результат | СН передает удаленному ТЕ запрос на разъединение, где он вызывает операцию tciDisconnect, чтобы разрушить логическое соединение между двумя указанными портами. Следует заметить, что оба порта могут быть в удаленных ТЕ. В этом случае операция возвращает значение только после вызова операции tciDisconnect в обоих удаленных ТЕ. | |

7.3.3.2.18 tciMapReq

| | | |
|------------------------------|---|---|
| Сигнатура | void tciMapReq(in TriPortIdType fromPort, in TriPortIdType toPort) | |
| Параметры in | fromPort | Идентификатор порта тестового компонента, подлежащего отображению. |
| | toPort | Идентификатор порта тестового компонента, отображение на который следует выполнить. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию отображения TTCN-3. | |
| Результат | СН передает удаленному ТЕ запрос на отображение, где он вызывает операцию tciMap, чтобы установить логическую связь между двумя указанными портами. | |

7.3.3.2.19 tciUnmapReq

| | | |
|------------------------------|--|---|
| Сигнатура | void tciUnmapReq(in TriPortIdType fromPort, in TriPortIdType toPort) | |
| Параметры in | fromPort | Идентификатор порта тестового компонента, не подлежащего отображению. |
| | toPort | Идентификатор порта тестового компонента, не подлежащего отображению. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию отсутствия отображения TTCN-3. | |
| Результат | СН передает запрос на отсутствие отображения удаленному ТЕ, где он вызывает операцию tciUnmap, чтобы разрушить логическое соединение между двумя указанными портами. | |

7.3.3.2.20 tciTestComponentTerminatedReq

| | | |
|------------------------------|--|---|
| Сигнатура | void tciTestComponentTerminatedReq(in TriComponentIdType component, in VerdictValue verdict) | |
| Параметры in | component | Идентификатор компонента, который завершен. |
| | verdict | Вердикт после завершения компонента. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если тестовый компонент завершает выполнение либо явным образом с помощью операции останова TTCN-3, либо неявным образом, если он достиг последнего оператора. | |
| Результат | СН получает уведомление о завершении указанного тестового компонента. Поскольку функция, выполняемая над тестовым компонентом, может иметь только параметры <i>in</i> (Рекомендация МСЭ-T Z.140 [2]), то операция tciTestComponentTerminateReq не имеет параметра parameterList. СН сообщает о завершении указанного компонента всем ТЕ, участвующим в сеансе тестирования, и специальному ТЕ*, следящему за всем вердиктом. | |

7.3.3.2.21 tciTestComponentRunningReq

| | | |
|------------------------------|---|---|
| Сигнатура | TBoolean tciTestComponentRunningReq(in TriComponentIdType component) | |
| Параметры in | component | Идентификатор компонента, подлежащего проверке на предмет выполнения. |
| Возвращаемое значение | true, если указанный компонент все еще выполняет поведение, в противном случае false. | |
| Ограничение | Эта операция вызывается ТЕ, если он проводит операцию выполнения TTCN-3. | |
| Результат | СН передает запрос на выполнение удаленному ТЕ, имеющему тестовый компонент, подлежащий проверке, где он вызывает операцию tciTestComponentRunning для проверки состояния выполнения указанного тестового компонента. | |

7.3.3.2.22 tciTestComponentDoneReq

| | | |
|------------------------------|--|---|
| Сигнатура | TBoolean tciTestComponentDoneReq(in TriComponentIdType comp) | |
| Параметры in | comp | Идентификатор компонента, подлежащего проверке на предмет завершенности выполнения. |
| Возвращаемое значение | true, если указанный компонент завершил выполнение своего поведения, в противном случае false. | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию done (завершение выполнения) TTCN-3. | |
| Результат | СН передает запрос на операцию done удаленному ТЕ, имеющему подлежащий проверке компонент, где он вызывает операцию tciTestComponentDone для проверки состояния указанного тестового компонента. | |

7.3.3.2.23 tciGetMTCReq

| | | |
|------------------------------|--|--|
| Сигнатура | TriComponentIdType tciGetMTCReq() | |
| Возвращаемое значение | Значение TriComponentIdType компонента МТС. | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию mtc TTCN-3. | |
| Результат | СН определяет идентификатор компонента для МТС. | |

7.3.3.2.24 tciExecuteTestCaseReq

| | | |
|------------------------------|--|---|
| Сигнатура | void tciExecuteTestCaseReq(in TciTestCaseIdType testCaseId, in TriPortIdListType tsiPortList) | |
| Параметры in | testCaseId | Идентификатор тестового примера, как определено в модуле TTCN-3. |
| | tsiPortList | Список tsiPortList содержит все порты, которые были объявлены в определении компонента системы для тестового примера, то есть порты интерфейса TSI. Если компонент системы не был определен явным образом для тестового примера, тогда список tsiPortList содержит все порты связи компонента МТС. Порядок следования портов в списке tsiPortList тот же, что и в соответствующем объявлении типа компонента TTCN-3. Если никакие порты не должны передаваться, тогда пересылается либо значение null, либо пустой список tsiPortList, то есть список нулевой длины. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция может вызываться ТЕ непосредственно перед тем, как он запускает поведение тестового примера в компоненте МТС (в ходе операции выполнения TTCN-3). | |
| Результат | СН передает запрос на выполнение тестового примера удаленным ТЕ, имеющим порты системы указанного тестового примера. Могут быть установлены статические соединения с системой SUT и инициализация средств связи для портов интерфейса TCI. | |

7.3.3.2.25 tciResetReq

| | | |
|------------------------------|--|--|
| Сигнатура | void tciResetReq() | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция может вызываться ТЕ в любое время для переустановки тестовой системы. | |
| Результат | СН передает запрос на переустановку всем участвующим в сеансе тестирования ТЕ. | |

7.3.3.2.26 tciKillTestComponentReq

| | | |
|------------------------------|---|---|
| Сигнатура | void tciKillTestComponentReq(in TriComponentIdType comp) | |
| Параметры in | comp | Идентификатор компонента, подлежащего устранению. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию kill (устранить) TTCN-3. | |
| Результат | СН передает удаленному ТЕ запрос на устранение компонента и вызывает там операцию tciKillTestComponent. | |

7.3.3.2.27 tciTestComponentAliveReq

| | | |
|------------------------------|---|---|
| Сигнатура | TBoolean tciTestComponentAliveReq(in TriComponentIdType comp) | |
| Параметры in | comp | Идентификатор компонента, подлежащего проверке на предмет функционирования. |
| Возвращаемое значение | true, если указанный компонент функционирует; в противном случае false. | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию alive (функционирование) TTCN-3. | |
| Результат | СН передает запрос удаленному ТЕ, который создал рассматриваемый тестовый компонент, где он вызывает операцию tciTestComponentAlive для проверки состояния указанного тестового компонента. | |

7.3.3.2.28 tciTestComponentKilledReq

| | | |
|------------------------------|--|---|
| Сигнатура | Tboolean tciTestComponentKilledReq(in TriComponentIdType comp) | |
| Параметры in | comp | Идентификатор компонента, подлежащего проверке на предмет устранения. |
| Возвращаемое значение | true, если указанный компонент был устранен; в противном случае false. | |
| Ограничение | Эта операция вызывается ТЕ, если он выполняет операцию killed (устранено) TTCN-3. | |
| Результат | СН передает запрос удаленному ТЕ, который создал рассматриваемый тестовый компонент, где он вызывает операцию tciTestComponentKilled для проверки состояния указанного тестового компонента. | |

7.3.4 Интерфейс TCI-TL

Интерфейс тестовой регистрации в интерфейсе TCI (TCI-TL) описывает операции, которые необходимо реализовать выполняемому модулю TTCN-3, и операции, которые предоставляются для ТЕ при реализации тестовой регистрации (Рисунок 8).

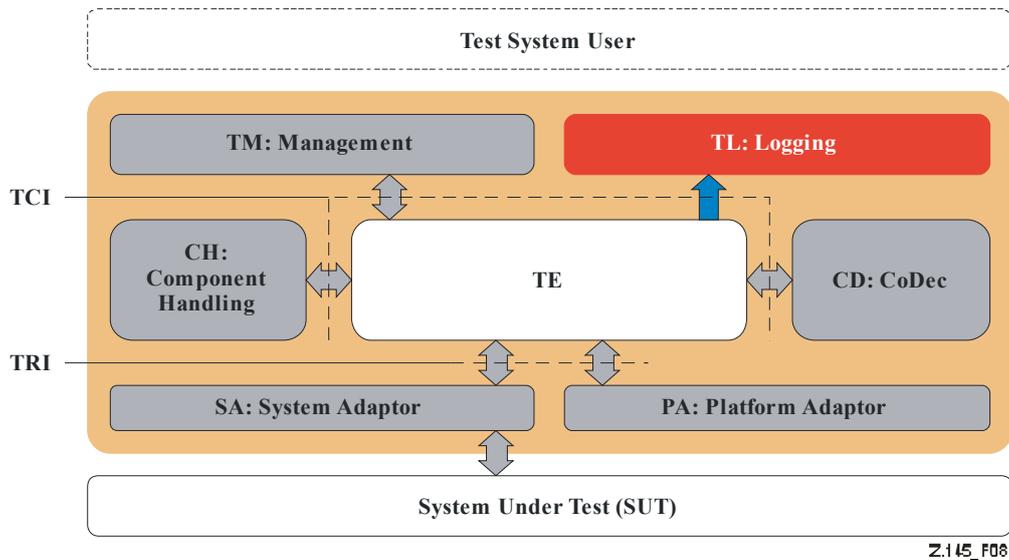


Рисунок 8/Z.145 – Интерфейс TCI-TL

При регистрации всем операциям уровня TTCN-3 предоставляется операция регистрации соответствующего события, которую выполняют для пользователя объекты TE, SA, PA, CH или CD.

7.3.4.1 Интерфейс TCI-TL provided

В данном разделе описываются операции, которые TL предоставляет для ТЕ.

7.3.4.1.1 tliTcExecute

| | | |
|------------------------------|--|---|
| Сигнатура | void tliTcExecute(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in TriTimerDurationType dur) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | tcId | Тестовый пример, подлежащий выполнению. |
| | dur | Продолжительность выполнения. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации запроса на выполнение тестового примера. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.2 tliTcStart

| | | |
|------------------------------|--|---|
| Сигнатура | void tliTcStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in TriTimerDurationType dur) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | tcId | Тестовый пример, подлежащий выполнению. |
| | dur | Продолжительность выполнения. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации запуска тестового примера. Данное событие происходит до запуска тестового примера. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.3 tliTcStop

| | | |
|------------------------------|--|--------------------------------------|
| Сигнатура | void tliTcStop(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации останова тестового примера. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.4 tliTcStarted

| | | |
|------------------------------|--|---|
| Сигнатура | void tliTcStarted(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in TriTimerDurationType dur) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | tcid | Тестовый пример, подлежащий выполнению. |
| | pars | Список параметров, запрашиваемых тестовым примером. |
| | dur | Продолжительность выполнения. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТМ для регистрации запуска тестового примера. Данное событие происходит после запуска тестового примера. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.5 tliTcTerminated

| | | |
|------------------------------|---|---|
| Сигнатура | void tliTcTerminated(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in VerdictValue outcome) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | tcId | Тестовый пример, подлежащий выполнению. |
| | pars | Список параметров, запрашиваемых тестовым примером. |
| | outcome | Вердикт тестового примера. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТМ для регистрации завершения тестового примера. Данное событие происходит после завершения тестового примера. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.6 tliCtrlStart

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliCtrlStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации запуска управляющей части. Данное событие происходит перед запуском управления. Если управление не представлено компонентом интерфейса TRI, то значение с является нулем. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.7 tliCtrlStop

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliCtrlStop(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается TE для регистрации останова управляющей части. Данное событие происходит перед остановом управления. Если управление не представлено компонентом интерфейса TRI, то значение c является нулем. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.8 tliCtrlTerminated

| | | |
|------------------------------|---|---------------------------------------|
| Сигнатура | void tliCtrlTerminated(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается TM для регистрации завершения управляющей части. Данное событие происходит после завершения управления. Если управление не представлено компонентом интерфейса TRI, то значение c является нулем. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.9 tliMSend_m

| | | |
|------------------------------|--|---|
| Сигнатура | void tliMSend_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriAddressType address, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается сообщение. |
| | msgValue | Значение, подлежащее кодированию и передаче. |
| | address | Адрес места назначения в системе SUT. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции одноадресной передачи. Данное событие происходит после передачи и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.10 tliMSend_m_BC

| | | |
|------------------------------|---|---|
| Сигнатура | void tliMSend_m_BC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается сообщение. |
| | msgValue | Значение, подлежащее кодированию и передаче. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | msg | Закодированное сообщение. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции широковещательной передачи. Данное событие происходит после передачи и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.11 tliMSend_m_MC

| | | |
|------------------------------|---|---|
| Сигнатура | void tliMSend_m_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается сообщение. |
| | msgValue | Значение, подлежащее кодированию и передаче. |
| | addresses | Адреса мест назначения в системе SUT. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | msg | Закодированное сообщение. |
| | | transmissionFailure |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции многоадресной передачи. Данное событие происходит после передачи и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.12 tliMSend_c

| | | |
|------------------------------|---|--|
| Сигнатура | void tliMSend_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdType to, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается сообщение. |
| | msgValue | Значение, подлежащее кодированию и передаче. |
| | to | Компонент, который будет получать сообщение. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции одноадресной передачи. Данное событие происходит после передачи и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.13 tliMSend_c_BC

| | | |
|------------------------------|---|--|
| Сигнатура | void tliMSend_c_BC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается сообщение. |
| | msgValue | Значение, подлежащее кодированию и передаче. |
| | to | Компонент, который будет получать сообщение. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции ширококвещательной передачи. Данное событие происходит после передачи и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.14 tliMSend_c_MC

| | | |
|------------------------------|--|--|
| Сигнатура | void tliMSend_c_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается сообщение. |
| | msgValue | Значение, подлежащее кодированию и передаче. |
| | toList | Компонент, который будет получать сообщение. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается CH для регистрации операции многоадресной передачи. Данное событие происходит после передачи и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.15 tliMDetected_m

| | | |
|------------------------------|---|--|
| Сигнатура | void tliMDetected_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriMessageType msg, in TriAddressType address) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается сообщение. |
| | msg | Принимаемое закодированное сообщение. |
| | address | Адрес источника в системе SUT. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации занесения сообщения в очередь. Данное событие происходит после занесения сообщения в очередь и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.16 tliMDetected_c

| | | |
|------------------------------|---|--|
| Сигнатура | void tliMDetected_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdType from) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается сообщение. |
| | msgValue | Принимаемое сообщение. |
| | from | Компонент, посылающий сообщение. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается CH для регистрации занесения сообщения в очередь. Данное событие происходит после занесения сообщения в очередь и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.17 tliMMismatch_m

| | | |
|------------------------------|---|---|
| Сигнатура | void tliMMismatch_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTpl) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается сообщение. |
| | msgValue | Сообщение, проверяемое на предмет совпадения с шаблоном. |
| | msgTpl | Шаблон, используемый для проверки на совпадение с сообщением. |
| | diffs | Различие/несовпадение между сообщением и шаблоном. |
| | address | Адрес источника в системе SUT. |
| addressTpl | Ожидаемый адрес источника в системе SUT. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается TE для регистрации несовпадения с шаблоном. Данное событие происходит после проверки на предмет совпадения с шаблоном и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.18 tliMMismatch_c

| | | |
|------------------------------|---|---|
| Сигнатура | <pre>void tliMMismatch_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTpl)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается сообщение. |
| | msgValue | Сообщение, проверяемое на предмет совпадения с шаблоном. |
| | msgTpl | Шаблон, используемый для проверки на совпадение с сообщением. |
| | diffs | Различие/несовпадение между сообщением и шаблоном. |
| | from | Компонент, посылающий сообщение. |
| | fromTpl | Ожидаемый компонент отправителя. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несовпадения с шаблоном. Данное событие происходит после проверки на предмет совпадения с шаблоном и используется для регистрации межкомпонентной связи. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.19 tliMReceive_m

| | | |
|------------------------------|--|---|
| Сигнатура | <pre>void tliMReceive_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTpl, in TriAddressType address, in TciValueTemplate addressTpl)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается сообщение. |
| | msgValue | Сообщение, проверяемое на предмет совпадения с шаблоном. |
| | msgTpl | Шаблон, используемый для проверки на совпадение с сообщением. |
| | address | Адрес источника в системе SUT. |
| | addressTpl | Ожидаемый адрес источника в системе SUT. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации приема сообщения. Данное событие происходит после проверки на предмет совпадения с шаблоном и используется для регистрации связи с системой SUT. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.20 tliMReceive_c

| | | |
|------------------------------|---|---|
| Сигнатура | <pre>void tliMReceive_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTpl, in TriComponentIdType from, in TciNonValueTemplate fromTpl)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается сообщение. |
| | msgValue | Сообщение, проверяемое на предмет совпадения с шаблоном. |
| | msgTpl | Шаблон, используемый для проверки на совпадение с сообщением. |
| | from | Компонент, посылающий сообщение. |
| fromTpl | Ожидаемый компонент отправителя. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации приема сообщения. Данное событие происходит после проверки на предмет совпадения с шаблоном и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.21 tliPrCall_m

| | | |
|------------------------------|--|---|
| Сигнатура | <pre>void tliPrCall_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriAddressType address, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который инициируется вызов. |
| | signature | Сигнатура вызываемой операции. |
| | parsValue | Параметры вызываемой операции. |
| | address | Адрес места назначения в системе SUT. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | pars | Закодированные параметры. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции одноадресного вызова. Данное событие происходит после выполнения вызова и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.22 tliPrCall_m_BC

| | | |
|------------------------------|---|---|
| Сигнатура | <pre>void tliPrCall_m_BC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который инициируется вызов. |
| | signature | Сигнатура вызываемой операции. |
| | parsValue | Параметры вызываемой операции. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции ширококвещательного вызова. Данное событие происходит после выполнения вызова и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.23 tliPrCall_m_MC

| | | |
|------------------------------|---|--|
| Сигнатура | <pre>void tliPrCall_m_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который инициируется вызов. |
| | signature | Сигнатура вызываемой операции. |
| | parsValue | Параметры вызываемой операции. |
| | addresses | Адреса мест назначения в системе SUT. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции многоадресного вызова. Данное событие происходит после выполнения вызова и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.24 tliPrCall_c

| | | |
|------------------------------|--|--|
| Сигнатура | <pre>void tliPrCall_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdType to, in TriStatusType transmissionFailure)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который инициируется вызов. |
| | signature | Сигнатура вызываемой операции. |
| | parsValue | Параметры вызываемой операции. |
| | to | Компонент, который будет принимать сообщение. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции одноадресного вызова. Данное событие происходит после выполнения вызова и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.25 tliPrCall_c_BC

| | | |
|------------------------------|---|---|
| Сигнатура | <pre>void tliPrCall_c_BC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriStatusType transmissionFailure)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который инициируется вызов. |
| | signature | Сигнатура вызываемой операции. |
| | parsValue | Параметры вызываемой операции. |
| | | transmissionFailure |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции широковещательного вызова. Данное событие происходит после выполнения вызова и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.26 tliPrCall_c_MC

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPrCall_c_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который инициируется вызов. |
| | signature | Сигнатура вызываемой операции. |
| | parsValue | Параметры вызываемой операции. |
| | toList | Компонент, который будет принимать сообщение. |
| transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции многоадресного вызова. Данное событие происходит после выполнения вызова и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.27 tliPrGetCallDetected_m

| | | |
|------------------------------|--|--|
| Сигнатура | void tliPrGetCallDetected_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriParameterListType pars, in TriAddressType address) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается вызов. |
| | signature | Сигнатура обнаруженного вызова. |
| | pars | Закодированные параметры обнаруженного вызова. |
| | address | Адрес места назначения в системе SUT. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции getcall с очередностью вызовов. Данное событие происходит после занесения вызова в очередь и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.28 tliPrGetCallDetected_c

| | | |
|------------------------------|---|--|
| Сигнатура | <pre>void tliPrGetCallDetected_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdType from)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается вызов. |
| | signature | Сигнатура вызываемой операции. |
| | parsValue | Закодированные параметры обнаруженного вызова. |
| | from | Компонент, вызвавший операцию. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции getcall с очередностью вызовов. Данное событие происходит после занесения вызова в очередь и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.29 tliPrGetCallMismatch_m

| | | |
|------------------------------|--|---|
| Сигнатура | <pre>void tliPrGetCallMismatch_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTpl)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается вызов. |
| | signature | Сигнатура обнаруженного вызова. |
| | parsValue | Параметры обнаруженного вызова. |
| | parsTpl | Шаблон, используемый для проверки на совпадение с параметром. |
| | diffs | Различие/несовпадение между вызовом и шаблоном. |
| | address | Адрес источника в системе SUT. |
| addressTpl | Ожидаемый адрес источника в системе SUT. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несовпадения операции getcall. Данное событие происходит после проверки операции getcall на предмет совпадения с шаблоном и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.30 tliPrGetCallMismatch_c

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPrGetCallMismatch_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTpl) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается вызов. |
| | signature | Сигнатура обнаруженного вызова. |
| | parsValue | Параметры обнаруженного вызова. |
| | parsTpl | Шаблон, используемый для проверки на совпадение с параметром. |
| | diffs | Различие/несовпадение между сообщением и шаблоном. |
| | from | Компонент, вызвавший операцию. |
| fromTpl | Ожидаемый вызывающий компонент. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несовпадения операции getcall. Данное событие происходит после проверки операции getcall на предмет совпадения с шаблоном и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.31 tliPrGetCall_m

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPrGetCall_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTpl, in TriAddressType address, in TciValueTemplate addressTpl) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается вызов. |
| | signature | Сигнатура обнаруженного вызова. |
| | parsValue | Параметры обнаруженного вызова. |
| | parsTpl | Шаблон, используемый для проверки на совпадение с параметром. |
| | address | Адрес источника в системе SUT. |
| addressTpl | Ожидаемый адрес источника в системе SUT. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации получения вызова. Данное событие происходит после того, как операция getcall была сопоставлена с шаблоном и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.32 tliPrGetCall_c

| | | |
|------------------------------|--|---|
| Сигнатура | <pre>void tliPrGetCall_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTpl, in TriComponentIdType from, in TciNonValueTemplate fromTpl)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается вызов. |
| | signature | Сигнатура обнаруженного вызова. |
| | parsValue | Параметры обнаруженного вызова. |
| | parsTpl | Шаблон, используемый для проверки на совпадение с параметром. |
| | from | Компонент, вызвавший операцию. |
| fromTpl | Ожидаемый вызывающий компонент. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации получения вызова. Данное событие происходит после того, как операция getcall была сопоставлена с шаблоном и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.33 tliPrReply_m

| | | |
|------------------------------|--|---|
| Сигнатура | <pre>void tliPrReply_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value parsValue, in Value replValue, in TriAddressType address, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)</pre> | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Ответ, подлежащий передаче. |
| | address | Адрес места назначения в системе SUT. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции одноадресного ответа. Данное событие происходит после выполнения ответа и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.34 tliPrReply_m_BC

| | | |
|------------------------------|---|---|
| Сигнатура | void tliPrReply_m_BC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value parsValue, in Value replValue, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Ответ, подлежащий передаче. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | repl | Закодированный ответ. |
| transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции широковещательного ответа. Данное событие происходит после выполнения ответа и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.35 tliPrReply_m_MC

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPrReply_m_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value parsValue, in Value replValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Ответ, подлежащий передаче. |
| | addresses | Адреса мест назначения в системе SUT. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | repl | Закодированный ответ. |
| transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции многоадресного ответа. Данное событие происходит после выполнения ответа и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.36 tliPrReply_c

| | | |
|------------------------------|---|--|
| Сигнатура | void tliPrReply_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value parsValue, in Value replValue, in TriComponentIdType to, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Ответ, подлежащий передаче. |
| | to | Компонент, который будет принимать ответ. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции одноадресного ответа. Данное событие происходит после выполнения ответа и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.37 tliPrReply_c_BC

| | | |
|------------------------------|--|--|
| Сигнатура | void tliPrReply_c_BC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value parsValue, in Value replValue, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Ответ, подлежащий передаче. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции широковещательного ответа. Данное событие происходит после выполнения ответа и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.38 tliPrReply_c_MC

| | | |
|------------------------------|--|--|
| Сигнатура | void tliPrReply_c_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value parsValue, in Value replValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Ответ, подлежащий передаче. |
| | toList | Компоненты, которые будут принимать ответ. |
| transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается CH для регистрации операции многоадресного ответа. Данное событие происходит после выполнения ответа и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.39 tliPrGetReplyDetected_m

| | | |
|------------------------------|---|--|
| Сигнатура | void tliPrGetReplyDetected_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriParameterType repl, in TriAddressType address) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | repl | Полученный закодированный ответ. |
| | address | Адрес источника в системе SUT. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции getreply с очередностью ответов. Данное событие происходит после занесения в очередь ответов операции getreply. Это событие используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.40 tliPrGetReplyDetected_c

| | | |
|------------------------------|--|--|
| Сигнатура | void tliPrGetReplyDetected_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value replValue, in TriComponentIdType from) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | replValue | Полученный ответ. |
| | from | Компонент, пославший ответ. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции getreply с очередностью ответов. Данное событие происходит после занесения в очередь ответов операции getreply. Это событие используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.41 tliPrGetReplyMismatch_m

| | | |
|------------------------------|---|--|
| Сигнатура | void tliPrGetReplyMismatch_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Полученный ответ. |
| | replyTmpl | Шаблон, используемый для проверки на совпадение с ответом. |
| | diffs | Различие/несовпадение между ответом и шаблоном. |
| | address | Адрес источника в системе SUT. |
| | addressTmpl | Ожидаемый адрес источника в системе SUT. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несовпадения с операцией getreply. Данное событие происходит после проверки операции getreply на предмет совпадения с шаблоном и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.42 tliPrGetReplyMismatch_c

| | | |
|------------------------------|--|--|
| Сигнатура | void tliPrGetReplyMismatch_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTpl) | |
| Параметры in | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Полученный ответ. |
| | replyTpl | Шаблон, используемый для проверки на совпадение с ответом. |
| | diffs | Различие/несовпадение между ответом и шаблоном. |
| | from | Компонент, пославший ответ. |
| fromTpl | Ожидаемый отвечающий компонент. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несовпадения с операцией getreply. Данное событие происходит после проверки операции getreply на предмет совпадения с шаблоном и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.43 tliPrGetReply_m

| | | |
|------------------------------|--|--|
| Сигнатура | void tliPrGetReply_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTpl, in TriAddressType address, in TciValueTemplate addressTpl) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Полученный ответ. |
| | replyTpl | Шаблон, используемый для проверки на совпадение с ответом. |
| | address | Адрес источника в системе SUT. |
| | addressTpl | Ожидаемый адрес источника в системе SUT. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации получения ответа. Данное событие происходит после проверки операции getreply на предмет совпадения с шаблоном и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.44 tliPrGetReply_c

| | | |
|------------------------------|--|--|
| Сигнатура | <pre>void tliPrGetReply_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)</pre> | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается ответ. |
| | signature | Сигнатура, относящаяся к ответу. |
| | parsValue | Параметры сигнатуры, относящиеся к ответу. |
| | replValue | Полученный ответ. |
| | replyTmpl | Шаблон, используемый для проверки на совпадение с ответом. |
| | from | Компонент, пославший ответ. |
| | fromTmpl | Ожидаемый отвечающий компонент. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается TE для регистрации получения ответа. Данное событие происходит после проверки операции getreply на предмет совпадения с шаблоном и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.45 tliPrRaise_m

| | | |
|------------------------------|---|---|
| Сигнатура | <pre>void tliPrRaise_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriAddressType address, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure)</pre> | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Особое состояние, подлежащее передаче. |
| | address | Адрес места назначения в системе SUT. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | exc | Закодированное особое состояние. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации одноадресной операции порождения особого состояния. Данное событие происходит после выполнения ответа и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.46 tliPrRaise_m_BC

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPrRaise_m_BC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Особое состояние, подлежащее передаче. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | exc | Закодированное особое состояние. |
| transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации широковещательной операции порождения особого состояния. Данное событие происходит после выполнения ответа и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.47 tliPrRaise_m_MC

| | | |
|------------------------------|---|---|
| Сигнатура | void tliPrRaise_m_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Особое состояние, подлежащее передаче. |
| | addresses | Адреса мест назначения в системе SUT. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | exc | Закодированное особое состояние. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации многоадресной операции порождения особого состояния. Данное событие происходит после выполнения ответа и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.48 tliPrRaise_c

| | | |
|------------------------------|---|---|
| Сигнатура | void tliPrRaise_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriComponentIdType to, in TriStatusType transmissionFailure) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Особое состояние, подлежащее передаче. |
| | to | Компонент, который будет принимать ответ. |
| transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации одноадресной операции порождения особого состояния. Данное событие происходит после выполнения ответа и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.49 tliPrRaise_c_BC

| | | |
|------------------------------|--|--|
| Сигнатура | void tliPrRaise_c_BC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriStatusType transmissionFailure) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Особое состояние, подлежащее передаче. |
| | transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации ширококвещательной операции порождения особого состояния. Данное событие происходит после выполнения ответа и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.50 tliPrRaise_c_MC

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPrRaise_c_MC(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который посылается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Особое состояние, подлежащее передаче. |
| | toList | Компоненты, которые будут принимать ответ. |
| transmissionFailure | Сообщение о неисправности, которое могло появиться при передаче. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации многоадресной операции порождения особого состояния. Данное событие происходит после выполнения ответа и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.51 tliPrCatchDetected_m

| | | |
|------------------------------|---|---|
| Сигнатура | void tliPrCatchDetected_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriExceptionType exc, in TriAddressType address) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | exc | Обнаруженное особое состояние. |
| address | Адрес источника в системе SUT. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции catch с очередностью особых состояний. Данное событие происходит после занесения в очередь особых состояний, обнаруженных операцией catch. Это событие используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.52 tliPrCatchDetected_c

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPrCatchDetected_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value excValue, in TriAddressType address) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | excValue | Обнаруженное особое состояние. |
| | address | Адрес источника в системе SUT. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции catch с очередностью особых состояний. Данное событие происходит после занесения в очередь особых состояний, обнаруженных операцией catch. Это событие используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.53 tliPrCatchMismatch_m

| | | |
|------------------------------|---|--|
| Сигнатура | void tliPrCatchMismatch_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTpl) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Полученное особое состояние. |
| | excTpl | Шаблон, используемый для проверки на совпадение с особым состоянием. |
| | diffs | Различие/несовпадение между особым состоянием и шаблоном. |
| | | address |
| | addressTpl | Ожидаемый адрес источника в системе SUT. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несовпадения с операцией catch. Данное событие происходит после проверки операции catch на предмет совпадения с шаблоном и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.54 tliPrCatchMismatch_c

| | | |
|------------------------------|---|--|
| Сигнатура | <pre>void tliPrCatchMismatch_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTpl)</pre> | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Полученное особое состояние. |
| | excTpl | Шаблон, используемый для проверки на совпадение с особым состоянием. |
| | diffs | Различие/несовпадение между особым состоянием и шаблоном. |
| | from | Компонент, пославший ответ. |
| fromTpl | Ожидаемый отвечающий компонент. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается TE для регистрации несовпадения с операцией catch. Данное событие происходит после проверки операции catch на предмет совпадения с шаблоном и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.55 tliPrCatch_m

| | | |
|------------------------------|---|---|
| Сигнатура | <pre>void tliPrCatch_m(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTpl, in TriAddressType address, in TciValueTemplate addressTpl)</pre> | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Полученное особое состояние. |
| | excTpl | Шаблон, используемый для проверки совпадения с особым состоянием. |
| | address | Адрес источника в системе SUT. |
| | addressTpl | Ожидаемый адрес источника в системе SUT. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации обнаружения особого состояния. Данное событие происходит после проверки операции catch на предмет совпадения с шаблоном и используется для регистрации связи с системой SUT. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.56 tliPrCatch_c

| | | |
|------------------------------|---|--|
| Сигнатура | void tliPrCatch_c(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TciValueTemplate excTpl, in TriComponentIdType from, in TciNonValueTemplate fromTpl) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| | parsValue | Параметры сигнатуры, относящиеся к особому состоянию. |
| | excValue | Полученное особое состояние. |
| | excTpl | Шаблон, используемых для проверки на совпадение с особым состоянием. |
| | from | Компонент, пославший ответ. |
| fromTpl | Ожидаемый отвечающий компонент. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации обнаружения особого состояния. Данное событие происходит после проверки операции catch на предмет совпадения с шаблоном и используется для регистрации межкомпонентной связи. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.57 tliPrCatchTimeoutDetected

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPrCatchTimeoutDetected(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается РА для регистрации обнаружения тайм-аута операции catch. Данное событие происходит после занесения тайм-аута в очередь. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.58 tliPrCatchTimeout

| | | |
|------------------------------|---|---|
| Сигнатура | void tliPrCatchTimeout(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, через который принимается особое состояние. |
| | signature | Сигнатура, относящаяся к особому состоянию. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации обнаружения тайм-аута. Данное событие происходит после выполнения тайм-аута операции catch. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.59 tliCCreate

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliCCreate(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TString name) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | comp | Компонент, который создается. |
| | name | Имя создаваемого компонента. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции создания компонента. Данное событие происходит после создания компонента. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.60 tliCStart

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliCStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TciBehaviourIdType beh, in TciParameterListType pars) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | comp | Компонент, который запускается. |
| | beh | Поведение, запускаемое в компоненте. |
| | pars | Параметры запущенного поведения. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции запуска компонента. Данное событие происходит после запуска компонента. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.61 tliCRunning

| | | |
|------------------------------|--|---|
| Сигнатура | void tliCRunning(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | comp | Компонент, который проверяется на предмет выполнения. |
| | status | Состояние этого компонента. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции выполнения компонента. Данное событие происходит после выполнения компонента. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.62 tliCAlive

| | | |
|------------------------------|--|---|
| Сигнатура | void tliCAlive(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | comp | Компонент, который проверяется на предмет выполнения. |
| | status | Состояние этого компонента. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции наличия компонента. Данное событие происходит после установления наличия компонента. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.63 tliCStop

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliCStop(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | comp | Компонент, который останавливается. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции останова компонента. Данное событие происходит после останова компонента. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.64 tliCKill

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliCKill(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | comp | Компонент, который останавливается. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции устранения компонента. Данное событие происходит после устранения компонента. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.65 tliCDoneMismatch

| | | |
|------------------------------|---|---|
| Сигнатура | void tliCDoneMismatch(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTpl) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | compTpl | Шаблон, используемый для проверки на совпадение с операцией done. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несовпадения с компонентной операцией done. Данное событие происходит после проверки операции done на предмет совпадения с шаблоном. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.66 tliCDone

| | | |
|------------------------------|--|---|
| Сигнатура | void tliCDone(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TciNonValueTemplate compTpl) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | compTpl | Шаблон, используемый для проверки на совпадение с операцией done. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации компонентной операции done. Данное событие происходит после операции done. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.67 tliCKilledMismatch

| | | |
|------------------------------|---|---|
| Сигнатура | void tliCKilledMismatch(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTpl) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | compTpl | Шаблон, используемый для проверки на совпадение с операцией done. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несовпадения с компонентной операцией killed. Данное событие происходит после проверки операции killed на предмет совпадения с шаблоном. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.68 tliCKilled

| | | |
|------------------------------|--|---|
| Сигнатура | void tliCKilled(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate compTpl) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | compTpl | Шаблон, используемый для проверки на совпадение с операцией done. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации компонентной операции killed. Данное событие происходит после операции killed. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.69 tliCTerminated

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliCTerminated(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | verdict | Вердикт компонента. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации завершения компонента. Данное событие происходит после завершения компонента. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.70 tliPConnect

| | | |
|------------------------------|--|--|
| Сигнатура | void tliPConnect(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | c1 | Компонент первого порта, подлежащего соединению. |
| | port1 | Первый порт, подлежащий соединению. |
| | c2 | Компонент второго порта, подлежащего соединению. |
| port2 | Второй порт, подлежащий соединению. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции соединения. Данное событие происходит после операции соединения. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.71 tliPDisconnect

| | | |
|------------------------------|---|--|
| Сигнатура | void tliPDisconnect(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | c1 | Компонент первого порта, подлежащего разъединению. |
| | port1 | Первый порт, подлежащий разъединению. |
| | c2 | Компонент второго порта, подлежащего разъединению. |
| port2 | Второй порт, подлежащий разъединению. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается СН для регистрации операции разъединения. Данное событие происходит после операции разъединения. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.72 tliPMap

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPMap(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | c1 | Компонент первого порта, подлежащего отображению. |
| | port1 | Первый порт, подлежащий отображению. |
| | c2 | Компонент второго порта, подлежащего отображению. |
| port2 | Второй порт, подлежащий отображению. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции отображения. Данное событие происходит после операции отображения. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.73 tliPUnmap

| | | |
|------------------------------|--|---|
| Сигнатура | void tliPUnmap(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | c1 | Компонент первого порта, отображение которого надлежит устранить. |
| | port1 | Первый порт, отображение которого надлежит устранить. |
| | c2 | Компонент второго порта, отображение которого надлежит устранить. |
| port2 | Второй порт, отображение которого надлежит устранить. | |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается SA для регистрации операции устранения отображения. Данное событие происходит после операции устранения отображения. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.74 tliPClear

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliPClear(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, подлежащий очищению. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается TE для регистрации операции очищения порта. Данное событие происходит после операции очищения порта. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.75 tliPStart

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliPStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, подлежащий запуску. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции запуска порта. Данное событие происходит после операции запуска порта. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.76 tliPStop

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliPStop(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, подлежащий останову. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции остановки порта. Данное событие происходит после операции остановки порта. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.77 tliPHalt

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliPHalt(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | port | Порт, подлежащий останову. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции отключения порта. Данное событие происходит после операции отключения порта. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.78 tliEncode

| | | |
|------------------------------|--|---|
| Сигнатура | void tliEncode(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value val, in TriStatusType encoderFailure, in TriMessageType msg, in TString codec) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | value | Значение, подлежащее кодированию. |
| | encoderFailure | Сообщение о неисправности, которое могло появиться при кодировании. |
| | msg | Закодированное значение. |
| | codec | Используемое кодирующее устройство. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается CD для регистрации операции кодирования. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.79 tliDecode

| | | |
|------------------------------|--|---|
| Сигнатура | void tliDecode(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value val, in TriStatusType decoderFailure, in TriMessageType msg, in TString codec) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | msg | Значение, подлежащее декодированию. |
| | decoderFailure | Сообщение о неисправности, которое могло появиться при декодировании. |
| | value | Декодированное значение. |
| | | codec |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается CD для регистрации операции декодирования. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.80 tliTTimeoutDetected

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliTTimeoutDetected(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | | timer |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается PA для регистрации обнаружения тайм-аута. Данное событие происходит после занесения тайм-аута в очередь. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.81 tliTTimeoutMismatch

| | | |
|------------------------------|--|--|
| Сигнатура | void tliTTimeoutMismatch(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTpl) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | timerTpl | Шаблон таймера, с которым отмечено несоответствие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несоответствия с тайм-аутом. Данное событие происходит после несоответствия с тайм-аутом. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.82 tliTTimeoutMismatch

| | | |
|------------------------------|--|---|
| Сигнатура | void tliTTimeoutMismatch(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTpl) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | timerTpl | Шаблон таймера, с которым произошло совпадение. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации совпадения с тайм-аутом. Данное событие происходит после совпадения с тайм-аутом. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.83 tliTStart

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliTStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType dur) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | timer | Таймер, который запускается. |
| | dur | Период времени таймера. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается РА для регистрации запуска таймера. Данное событие происходит после операции запуска таймера. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.84 tliTStop

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliTStop(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | timer | Таймер, который останавливается. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается РА для регистрации останова таймера. Данное событие происходит после операции останова таймера. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.85 tliTRead

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliTRead(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType elapsed) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | timer | Таймер, который запускается. |
| | elapsed | Истекшее время таймера. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается РА для регистрации считывания времени таймера. Данное событие происходит после операции считывания времени таймера. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.86 tliTRunning

| | | |
|------------------------------|--|--|
| Сигнатура | void tliTRunning(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TBoolean status) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | timer | Таймер, который проверяется на предмет выполнения. |
| | status | Состояние этого компонента. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается РА для регистрации операции выполнения таймера. Данное событие происходит после операции выполнения таймера. | |
| Результат | TL представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.87 tliSEnter

| | | |
|------------------------------|---|---------------------------------------|
| Сигнатура | void tliSEnter(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in TciParameterListType parsValue, in TString kind) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | name | Имя контекста. |
| | parsValue | Параметры контекста. |
| | kind | Вид контекста. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации ввода контекста. Данное событие происходит после ввода контекста. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.88 tliSLeave

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliSLeave(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value val, in TString kind) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | name | Имя контекста. |
| | val | Возвращаемое значение контекста. |
| | kind | Вид контекста. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации вывода контекста. Данное событие происходит после вывода контекста. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.89 tliVar

| | | |
|---------------------|--|---------------------------------------|
| Сигнатура | void tliVar(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value varValue) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | name | Имя переменной. |
| | varValue | Новое значение переменной. |
| | Возвращаемое значение | void |
| Ограничение | Эта операция вызывается ТЕ для регистрации модификации значения переменной. Данное событие происходит после изменения значений. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.90 tliModulePar

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliModulePar(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value parValue) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | name | Имя параметра модуля. |
| | parValue | Значение параметра модуля. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации значения параметра модуля. Данное событие происходит после доступа к значению параметра модуля. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.91 tliGetVerdict

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliGetVerdict(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | verdict | Текущее значение локального вердикта. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции getverdict. Данное событие происходит после операции getverdict. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.92 tliSetVerdict

| | | |
|------------------------------|--|--|
| Сигнатура | void tliSetVerdict(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | verdict | Значение, подлежащее установке на локальный вердикт. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации операции setverdict. Данное событие происходит после операции setverdict. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.93 tliLog

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliLog(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciValueList log) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | log | Строка, подлежащая регистрации. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТМ для регистрации регистрационного файла оператора TTCN-3. Данное событие происходит после операции регистрации TTCN-3. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.94 tliAEnter

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliAEnter(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации ввода операции alt. Данное событие происходит после ввода операции alt. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.95 tliALeave

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliALeave(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации вывода операции alt. Данное событие происходит после вывода операции alt. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.96 tliANomatch

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliANomatch(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации несовпадения с операцией alt. Данное событие происходит после несовпадения с операцией alt. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.97 tliARepeat

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliARepeat(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации повторения операции alt. Данное событие происходит при повторении операции alt. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.98 tliADefaults

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliADefaults(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации ввода раздела по умолчанию. Данное событие происходит после ввода раздела по умолчанию. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.99 tliAActivate

| | | |
|------------------------------|--|--|
| Сигнатура | void tliAActivate(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in TciParameterListType pars, in Value ref) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | name | Имя значения по умолчанию. |
| | pars | Параметр значения по умолчанию. |
| | ref | Полученная в результате ссылка по умолчанию. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации активизации значения по умолчанию. Данное событие происходит после активизации значения по умолчанию. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.100 tliADeactivate

| | | |
|---------------------|--|--|
| Сигнатура | void tliADeactivate(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value ref) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| | ref | Полученная в результате ссылка по умолчанию. |
| | Возвращаемое значение | void |
| Ограничение | Эта операция вызывается ТЕ для регистрации деактивизации значения по умолчанию. Данное событие происходит после деактивизации значения по умолчанию. | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

7.3.4.1.101 tliAWait

| | | |
|------------------------------|--|---------------------------------------|
| Сигнатура | void tliAWait(in string am, in long ts, in string src, in long line, in TriComponentId c) | |
| В параметрах | am | Дополнительное сообщение. |
| | ts | Время генерирования события. |
| | src | Исходный файл спецификации теста. |
| | line | Номер строки, где выполняется запрос. |
| | c | Компонент, генерирующий это событие. |
| Возвращаемое значение | void | |
| Ограничение | Эта операция вызывается ТЕ для регистрации того, что компонент ожидает события для нового "снимка". | |
| Результат | ТЛ представляет пользователю всю информацию, содержащуюся в параметрах данной операции; вопрос о том, как это осуществляется, выходит за рамки настоящей Рекомендации. | |

8 Отображение данных языка Java

8.1 Введение

В данном разделе представлено отображение языка Java интерфейса TCI. В целях большей эффективности представлено специальное отображение языка вместо использования отображения языка IDL стандарта OMG на язык Java.

Отображение языка Java для интерфейса управления TTCN-3 описывает, как определения IDL, приведенные в разделе 7, отображаются на язык Java. Отображение языка не зависит от используемой версии Java, поскольку используются только базовые конструкции языка Java.

8.2 Имена и контексты

8.2.1 Имена

К идентификаторам в IDL применяется ряд правил трансляции имен, хотя между идентификаторами, используемыми в определении IDL и языке Java, противоречий нет.

В интерфейсах или идентификаторах класса языка Java опускается конечный тип `Type`, используемый в определении IDL.

ПРИМЕР: Тип `tciTestCaseIdType` IDL отображается на `tciTestCaseId` в языке Java.

Полученное в результате отображение соответствует стандартным соглашениям по кодированию в языке Java.

8.2.2 Контексты

Модуль `tciInterface` IDL отображается на пакет `org.etsi.ttcn3.tci` Java. Все объявления типов IDL в этом модуле отображаются на объявления классов или интерфейсов Java в этом пакете.

Отображение типов

Отображение базовых типов

В Таблице 2 приведен обзор в отношении того, как собственные типы `TBoolean`, `TFloat`, `TInteger`, `TString` и `TStringSeq` отображаются на типы в языке Java.

Таблица 2/Z.1.45 – Отображения базовых типов

| Тип в IDL | Тип в Java |
|-----------|------------|
| | |
| | |
| | |
| | |
| | |

Собственный тип `TObjId` определяется в соответствующей части интерфейса `ObjidValue`.

boolean

Тип `TBoolean` в IDL отображается на базовый тип `boolean` в языке java.

float

Тип `TFloat` в IDL отображается на базовый тип `float` в языке java.

char

Тип `TChar` в IDL отображается на базовый тип `char` в языке java.

int

Тип `TInteger` в IDL отображается на базовый тип `int` в языке java.

String

Тип **TString** в IDL отображается на класс `java.lang.String` без проверки диапазона или границ для символов в строке. Все возможные строки, которые определены в TTCN-3, могут быть преобразованы в `java.lang.String`.

String[]

Тип **TStringSeq** в IDL отображается на массив класса `java.lang.String`.

Universal Char

Тип **TUniversalChar** в IDL отображается на базовый тип `int` в языке `java`. Для целого числа используется каноническая форма, как определено в п. 6.2/X.290.

Отображение структурных типов

В описании IDL интерфейса TCI определяемые пользователем типы рассматриваются как собственные типы. В языке `Java` эти типы отображаются на интерфейсы `Java`. Эти интерфейсы определяют методы и атрибуты, доступные для объектов, реализующих этот интерфейс.

8.2.2.1 TciParameterType

TciParameterType отображается на следующий интерфейс:

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

Методы:

- `getParameterName()` Возвращает имя параметра, как определено в спецификации TTCN-3;
- `setParameterName(String name)` Устанавливает имя этого параметра `TciParameter` на `name`;
- `getParameterPassingMode()` Возвращает режим передачи параметра для этого параметра;
- `setParameterPassingMode(TciParameterPassingMode mode)`
Устанавливает режим параметра для этого параметра `TriParameter` на `mode`;
- `getParameter()` Возвращает параметр `Value` этого `TciParameter` или объект `null`, если этот параметр содержит особое значение `null`;
- `setParameter(Value parameter)` Устанавливает параметр `Value` этого `TciParameter` на параметр. Если устанавливается особое значение `null` для указания на то, что этот параметр не содержит значение, то в качестве параметра передается `null` в языке `Java`.

8.2.2.2 TciParameterPassingModeType

TciParameterPassingModeType отображается на следующий интерфейс:

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

Константы:

- TCI_IN Будет использоваться для указания на то, что TciParameter является параметром in;
- TCI_INOUT Будет использоваться для указания на то, что TciParameter является параметром inout;
- TCI_OUT Будет использоваться для указания на то, что TciParameter является параметром out.

8.2.2.3 TciParameterListType

TciParameterListType отображается на следующий интерфейс:

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

Методы:

- size() Возвращает число параметров в этом списке;
- isEmpty() Возвращает true, если этот список не содержит параметров;
- getParameters() Возвращает Enumeration по всем параметрам в этом списке. В результате такого перечисления порядок следования параметров тот же, что и в списке;
- get(int index) Возвращает TciParameter на заданную позицию;
- clear() Удаляет все параметры из этого списка TciParameterList;
- add(TciParameter parameter) Добавляет parameter в конец этого списка TciParameterList;
- setParameter(TciParameter[] parameters) Заполняет этот список TciParameterList параметрами parameters.

8.2.2.4 TciTypeClassType

TciTypeClassType отображается на следующий интерфейс:

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

8.2.2.5 TciTestComponentKindType

TciTestComponentKindType отображается на следующий интерфейс:

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

8.2.2.6 TciBehaviourIdType

TciBehaviourIdType отображается на следующий интерфейс:

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

8.2.2.7 TciTestCaseIdType

TciTestCaseIdType отображается на следующий интерфейс:

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

8.2.2.8 TciModuleIdType

TciModuleIdType отображается на следующий интерфейс :

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

8.2.2.9 TciModuleParameterIdType

TciModuleParameterIdType отображается на следующий интерфейс:

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

8.2.2.10 TciModuleParameterListType

TciModuleParameterListType отображается на следующий интерфейс :

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

Методы:

- `size()` Возвращает число параметров модуля в этом списке;
- `isEmpty()` Возвращает `true`, если этот список не содержит параметров модуля;
- `getModuleParameters()` Возвращает `Enumeration` по параметрам модуля в списке. В результате этого перечисления порядок следования параметров модуля тот же, что и в списке;
- `get(int index)` Возвращает `TciModuleParameter` на заданную позицию.

8.2.2.11 TciModuleParameterType

TciModuleParameterType отображается на следующий интерфейс:

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

Методы:

- `getModuleParameterName()` Возвращает имя параметра модуля, как определено в спецификации TTCN-3;
- `getDefaultValue()` Возвращает параметр модуля `Value` по умолчанию этого параметра `TciModuleParameter` или объект `null`, если параметр модуля содержит особое значение `null`.

8.2.2.12 TciParameterTypeListType

TciParameterTypeListType отображается на следующий интерфейс:

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

Методы:

- `size()` Возвращает число типов параметра в этом списке;
- `isEmpty()` Возвращает `true`, если этот список не содержит типов параметра;
- `getParameterTypes()` Возвращает `Enumeration` по типам параметра в списке. В результате этого перечисления порядок следования типов параметра тот же, что и в списке;
- `get(int index)` Возвращает `TciParameterType` на заданную позицию.

8.2.2.13 TciModuleIdListType

TciModuleIdListType отображается на следующий интерфейс:

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

Методы:

- `size()` Возвращает число модулей в этом списке;
- `isEmpty()` Возвращает `true`, если этот список не содержит модулей;
- `getImportedModules()` Возвращает `Enumeration` по модулям в списке. В результате этого перечисления порядок следования модулей тот же, что и в списке;
- `get(int index)` Возвращает `TciModuleId` на заданную позицию.

8.2.3 Отображение абстрактных типов

Типы данных TTCN-3 моделируются в языке Java путем использования отображения абстрактных типов, как определено в этом разделе. Интерфейс `Type` описывает только операции, используемые в определяемых типах

TTCN-3 для вывода. Невозможно конструировать типы TTCN-3, используя интерфейс `Type`. Типы моделируются путем использования единого интерфейса `Type`, который предоставляет методы для идентификации типов и вывода значений данного типа.

8.2.3.1 Type

`Type` отображается на следующий интерфейс:

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

Методы:

- `getDefiningModule()` Возвращает идентификатор модуля для модуля, в котором был определен тип. Если тип представляет базовый тип TTCN-3, то будет возвращено особое значение `null`;
- `getName()` Возвращает имя типа, как определено в модуле TTCN-3;
- `getTypeClass()` Возвращает класс типа для соответствующего типа. Значение типа `TciTypeClassType` может иметь одну из следующих констант: `ADDRESS`, `ANYTYPE`, `BITSTRING`, `BOOLEAN`, `CHARSTRING`, `COMPONENT`, `ENUMERATED`, `FLOAT`, `HEXSTRING`, `INTEGER`, `OBJID`, `OCTETSTRING`, `RECORD`, `RECORD_OF`, `SET`, `SET_OF`, `UNION`, `UNIVERSAL_CHARSTRING`, `VERDICT`;
- `newInstance()` Возвращает вновь созданное значение данного типа. Это начальное значение созданного значения является неопределенным;
- `getTypeEncoding()` Возвращает атрибут кодирования типа, как определено в модуле TTCN-3;
- `getTypeEncodingVariant()` Эта операция возвращает атрибут варианта кодирования значения, как определено в TTCN-3, если таковой имеется. Если не был определен атрибут варианта кодирования, то будет возвращено особое значение `null`;
- `getTypeEncoding()` Возвращает атрибут расширения типа, как определено в модуле TTCN-3.

8.2.4 Отображение абстрактных значений

Значения TTCN-3 могут быть выведены из TE и сконструированы путем использования интерфейса `Value`. Интерфейс отображения значений конструируется иерархически с помощью `Value` как базовый интерфейс. Были определены специализированные интерфейсы для различных типов значений.

8.2.4.1 Value

`Value` отображается на следующий интерфейс:

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

Методы:

- `getType()` Возвращает тип заданного значения;
- `notPresent()` Возвращает `true`, если заданным значением является `omit`, в противном случае возвращает `false`;
- `getValueEncoding()` Эта операция возвращает атрибут кодирования значения, как определено в TTCN-3, если таковой имеется. Если атрибут кодирования не был определен, то будет возвращено особое значение `null`;

- `getValueEncodingVariant()` Эта операция возвращает атрибут варианта кодирования значения, как определено в TTCN-3, если таковой имеется. Если атрибут варианта кодирования не был определен, то будет возвращено особое значение `null`;

8.2.4.2 IntegerValue

Тип `IntegerValue` отображается на следующий интерфейс:

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

Методы:

- `setInteger(int value)` Устанавливает `IntegerValue` на целочисленное значение `value`;
- `getInteger()` Возвращает целочисленное значение, представленное этим `IntegerValue`.

8.2.4.3 FloatValue

Тип `FloatValue` отображается на следующий интерфейс:

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

Методы:

- `setFloat(float value)` Устанавливает это `FloatValue` на значение `value` с плавающей запятой;
- `getFloat()` Возвращает значение с плавающей запятой, представленное этим `FloatValue`.

8.2.4.4 BooleanValue

Тип `BooleanValue` отображается на следующий интерфейс:

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

Методы:

- `setBoolean(boolean value)` Устанавливает это `BooleanValue` на булево значение `value`;
- `getBoolean()` Возвращает булево значение, представленное этим `BooleanValue`.

8.2.4.5 ObjidValue

`ObjidValue` отображается на следующий интерфейс:

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

Методы:

- `getObjid()` Возвращает значение идентификатора объекта для `objid` TTCN-3;
- `setObjid(TciObjId value)` Устанавливает это `ObjidValue` на значение `value`.

8.2.4.6 TciObjId

TciObjId отображается на следующий интерфейс. Собственное представление в языке java идентификатора ObjectId TTCN-3 состоит из упорядоченной последовательности элементов TciObjIdElement.

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

Методы:

- size() Возвращает размер этого идентификатора Object Id в элементах TciObjIdElement;
- setObjElement(TciObjIdElement[] objElements) Устанавливает этот идентификатор ObjId в список элементов objElements;
- getObjElement(int index) Возвращает элемент TciObjIdElement на позицию index.

8.2.4.7 TciObjIdElement

Элемент TciObjIdElement представляет одиночный элемент объекта в значении ObjId TTCN-3. Он может быть установлен путем использования различных представлений, подобных представлению ASCII, или как целое число.

TciObjIdElement отображается на следующий интерфейс:

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

Методы:

- setElementAsAscii(String element) Устанавливает внутреннее представление этого элемента ObjIdElement на строковое значение element;
- setElementAsNumber(int element) Устанавливает внутреннее представление этого элемента ObjIdElement на целочисленное значение element;
- getElementAsAscii() Возвращает внутреннее представление этого элемента ObjIdElement в виде строки;
- getElementAsNumber() Возвращает внутреннее представление этого элемента ObjIdElement в виде целого числа;

8.2.4.8 CharstringValue

CharstringValue отображается на следующий интерфейс:

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

Методы:

- getString() Возвращает строку из charstring TTCN-3. Текстовым представлением пустой строки charstring TTCN-3 является '', в то время как ее длина равна нулю;
- setString(String value) Устанавливает это CharstringValue на значение value;

- `getChar(int position)` Возвращает значение символа строки `charstring` TTCN-3 на позицию `position`. Позиция 0 обозначает первый символ строки `charstring` TTCN-3. Диапазон допустимых значений для позиции составляет от 0 до `length - 1`;
- `setChar(int position, char value)` Устанавливает символ в позиции `position` на значение `value`. Диапазон допустимых значений для позиции `position` составляет от 0 до `length - 1`;
- `getLength()` Возвращает длину этого значения `CharstringValue` в символах; нуль, если значением этого `CharstringValue` является `omit`;
- `setLength(int len)` Устанавливает длину этого значения `CharstringValue` в символах на `len`.

8.2.4.9 BitstringValue

BitstringValue отображается на следующий интерфейс:

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

Методы:

- `getString()` Возвращает текстовое представление этого значения `BitstringValue`, как определено в TTCN-3. Например, текстовым представлением для 0101 является "0101"В. Текстовым представлением пустой строки `bitstring` TTCN-3 является ""В, в то время как ее длина равна нулю;
- `setString(String value)` Устанавливает значение этого `BitstringValue` согласно текстовому представлению, как определено значением `value`. Например, значением этого `BitstringValue` будет 0101, если текстовым представлением в значении является "0101"В;
- `getBit(int position)` Возвращает значение (0 | 1) на позицию данной битовой строки TTCN-3. Позиция 0 обозначает первый бит битовой строки TTCN-3. Диапазон допустимых значений для позиции составляет от 0 до `length - 1`;
- `setBit(int position, int value)` Устанавливает бит в позиции на значение (0 | 1). Позиция 0 обозначает первый бит в этом значении `BitstringValue`. Диапазон допустимых значений для позиции составляет от 0 до `length - 1`;
- `getLength()` Возвращает длину этого значения `BitstringValue` в битах; нуль, если значением этого `BitstringValue` является `omit`;
- `setLength(int len)` Устанавливает длину этого значения `BitstringValue` в битах на `len`.

8.2.4.10 OctetstringValue

OctetstringValue отображается на следующий интерфейс:

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

Методы:

- `getString()` Возвращает текстовое представление этого значения `OctetstringValue`, как определено в TTCN-3. Например, текстовым представлением для `0xCAFFEE` является "CAFFEE"O. Текстовым представлением пустой строки `octetstring` TTCN-3 является "O", в то время как ее длина равна нулю;
- `setString(String value)` Устанавливает значение этого `OctetstringValue` согласно текстовому представлению, как определено значением `value`. Например, значением этого `OctetstringValue` будет `0xCAFFEE`, если текстовым представлением в значении является "CAFFEE"O;
- `getOctet(int position)` Возвращает значение (0..255) на позиции `position` этой строки октетов TTCN-3. Позиция 0 обозначает первый октет строки октетов TTCN-3. Диапазон допустимых значений для позиции составляет от 0 до `length - 1`;
- `setOctet(int position, int value)` Устанавливает октет в позиции `position` на значение (0..255). Позиция 0 обозначает первый октет в строке октетов. Диапазон допустимых значений для позиции составляет от 0 до `length - 1`;
- `getLength()` Возвращает длину этого значения `OctetstringValue` в октетах; нуль, если значением для этого `OctetstringValue` является `omit`;
- `setLength(int len)` Устанавливает длину этого значения `OctetstringValue` в октетах на `len`.

8.2.4.11 UniversalCharstringValue

`UniversalCharstringValue` отображается на следующий интерфейс:

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

Методы:

- `getString()` Возвращает текстовое представление этого значения `UniversalCharstringValue`, как определено в TTCN-3;
- `setString(String value)` Устанавливает значение этого `UniversalCharstringValue` согласно текстовому представлению, как определено значением `value`;
- `getChar(int position)` Возвращает значение `UniversalChar` строки `universal charstring` TTCN-3 на позицию. Позиция 0 обозначает первый символ `UniversalChar` строки `universal charstring` TTCN-3. Диапазон допустимых значений для позиции `position` составляет от 0 до `length - 1`;
- `setChar(int position, char value)` Устанавливает символ `UniversalChar` в позиции `position` на значение `value`. Диапазон допустимых значений для `position` составляет от 0 до `length - 1`;
- `getLength()` Возвращает длину этого значения `UniversalCharstringValue` в символах `UniversalChars`; нуль, если значением этого `UniversalCharstringValue` является `omit`;
- `setLength(int len)` Устанавливает длину этого значения `UniversalCharstringValue` в символах `UniversalChars` на `len`.

8.2.4.12 HexstringValue

HexstringValue отображается на следующий интерфейс:

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

Методы:

- `getString()` Возвращает текстовое представление этого значения `HexstringValue`, как определено в TTCN-3. Например, текстовым представлением для `0xAFFEE` является "AFFEE"Н. Текстовым представлением пустой строки `hexstring` TTCN-3 является ""Н, в то время как ее длина равна нулю;
- `setString(String value)` Устанавливает значение этого `HexstringValue` согласно текстовому представлению, как определено значением `value`. Например, значением этого `HexstringValue` будет `0xAFFEE`, если текстовым представлением в значении `value` является "AFFEE"Н;
- `getHex(int position)` Возвращает значение (0..15) на позицию `position` этой шестнадцатеричной строки TTCN-3. Позиция 0 обозначает первые шестнадцатеричные разряды шестнадцатеричной строки TTCN-3. Диапазон допустимых значений для позиции составляет от 0 до `length - 1`;
- `setHex(int position, int value)` Устанавливает шестнадцатеричный разряд в позиции на значение (0..16). Позиция 0 обозначает первый октет в строке `hexstring`. Диапазон допустимых значений для позиции составляет от 0 до `length - 1`;
- `getLength()` Возвращает длину этого значения `HexstringValue` в октетах; нуль, если значением этого `HexstringValue` является `omit`;
- `setLength(int len)` Устанавливает длину этого значения `HexstringValue` в шестнадцатеричных разрядах на `len`.

8.2.4.13 RecordValue

отображается на следующий интерфейс:

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

Методы:

- `getField(String fieldName)` Возвращает значение поля, именуемого `fieldName`. Возвращаемое значение является общим абстрактным базовым типом `Value`, поскольку поле записи может иметь любой тип, определяемый в TTCN-3. Если это поле не может быть получено из записи, то будет возвращено особое значение `null`;
- `setField(String fieldName, Value value)` Устанавливает поле записи, именуемое `fieldName`, на значение `value`. Никаких предположений не делается относительно того, как поле сохраняется в записи. При внутренней реализации может оказаться предпочтительным хранить ссылку на это значение или копировать это значение. Можно с уверенностью предположить, что данное значение будет скопировано. Поэтому следует допустить, что дальнейшие модификации значения `value` не будут отражаться в этой записи;
- `getFieldNames()` Возвращает массив строки имен полей, особое значение `null`, если запись не имеет полей.

8.2.4.14 RecordOfValue

RecordOfValue отображается на следующий интерфейс:

```
// TCI IDL RecordOfValue
package org.etsi.ttcn.tci;
public interface RecordOfValue {
    public Value      getField(String fieldName) ;
    public void       setField(int position, Value value) ;
    public void       appendField(Value value) ;
    public Type       getElementType() ;
    public int        getLength() ;
    public void       setLength(int len) ;
}
```

Методы:

- `getField(String fieldName)` Возвращает значение для `record of` на позиции `position`, если `position` находится между `zero` и `length - 1`; в противном случае возвращается особое значение `null`. Возвращаемое значение является общим абстрактным базовым типом `Value`, поскольку а `record of` может иметь поля любого типа, определяемого в TTCN-3;
- `setField(int position, Value value)` Устанавливает поле в позиции `position` на значение `value`. Если `position` больше, чем `(length - 1)`, то запись будет расширена, чтобы иметь длину `(position + 1)`. Элементы `record of` между начальной позицией на `length` и на `position - 1` будут установлены на ОМТ. Никакого предположения не делается относительно того, как поле сохраняется в элементе `record of`. При внутренней реализации может оказаться предпочтительным хранить ссылку на это значение или копировать это значение. Можно с уверенностью предположить, что данное значение будет скопировано. Поэтому следует допустить, что дальнейшие модификации `value` не будут отражаться в элементе `record of`;
- `appendField(Value value)` Добавляет значение в конце элемента `record of`, то есть на позиции `length`. Никакого предположения не делается относительно того, как поле сохраняется в элементе `record of`. При внутренней реализации может оказаться предпочтительным хранить ссылку на это значение или копировать это значение. Можно с уверенностью предположить, что данное значение будет скопировано. Поэтому следует допустить, что дальнейшие модификации `value` не будут отражаться в элементе `record of`;
- `getElementType()` Эта операция будет возвращать тип `Type` элементов данного элемента `record of`;
- `getLength()` Возвращает фактическую длину значения элемента `record of`; возвращает нуль, если значением элемента `record of` является ОМТ;
- `setLength(int len)` Устанавливает длину элемента `record of` на `len`. Если значение `len` больше, чем исходная длина, то вновь созданные элементы имеют значение ОМТ. Если значение `len` меньше или равно исходной длине, то эта операция будет игнорироваться.

8.2.4.15 UnionValue

UnionValue отображается на следующий интерфейс:

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

Методы:

- `getVariant(String variantName)` Возвращает значение имени `variantName` объединения TTCN-3, если имя `variantName` равно результату операции `getPresentVariantName`; в противном случае равно особому значению `null`. Имя `VariantName` обозначает имя варианта объединения, как определено в TTCN-3;
- `setVariant(String variantName, Value value)` Устанавливает имя `variantName` объединения на значение. Если имя `variantName` не определено для этого объединения, то данная операция будет проигнорирована. Если было выбрано другое имя варианта, то взамен будет выбран новый вариант;
- `getPresentVariantName()` Возвращает имя варианта, которое имеет значение в этом объединении, установленное как `String`. Будет возвращено особое значение `null`, если вариант не выбран;
- `getVariantNames()` Возвращает массив элемента `String` имен варианта, особое значение `null`, если объединение не имеет полей. Если значение `UnionValue` представляет тип `anytype` TTCN-3, то есть, если класс типа, для типа, полученного операцией `getType()`, является `ANYTYPE`, то все предопределенные и определяемые пользователем типы TTCN-3 будут возвращены.

8.2.4.16 EnumeratedValue

EnumeratedValue отображается на следующий интерфейс:

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

Методы:

- `getEnum()` Возвращает идентификатор строки этого значения `EnumeratedValue`. Данный идентификатор равен идентификатору в спецификации TTCN-3;
- `setEnum(String enumValue)` Установить перечисление на `enumValue`. Если `enumValue` не является допустимым значением для этого перечисления, то операция будет проигнорирована.

8.2.4.17 VerdictValue

VerdictValue отображается на следующий интерфейс:

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

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

Методы:

- `getVerdict()` Возвращает целочисленное значение для этого `VerdictValue`. Элемент `integer` является одной из следующих констант: `ERROR`, `FAIL`, `INCONC`, `NONE`, `PASS`;
- `setVerdict(int verdict)` Устанавливает это значение `VerdictValue` на `verdict`. Следует отметить, что значение `VerdictValue` в любой момент времени может быть установлено на любой из упомянутых выше вердиктов. Элемент `VerdictValue` не выполняет никаких вычислений вердикта, как определено в TTCN-3. Например, допустимо устанавливать значение `VerdictValue` сначала на `ERROR`, а затем на `PASS`.

8.2.4.18 AddressValue

AddressValue отображается на следующий интерфейс:

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

Методы:

- `getAddress()` Возвращает значение, представленное этим `AddressValue`;
- `setAddress(Value value)` Устанавливает это `AddressValue` на значение `value`.

8.2.5 Отображение абстрактных типов регистрации

Для облегчения регистрации сопоставлений между значениями и шаблонами определяются дополнительные типы.

8.2.5.1 TciValueTemplate

TciValueTemplate отображается на следующий интерфейс:

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

Методы:

- `isOmit()` Возвращает `true`, если шаблоном является `omit`; в противном случае возвращает `false`;
- `isAny()` Возвращает `true`, если шаблоном является `any`; в противном случае возвращает `false`;
- `isAnyOrOmit()` Возвращает `true`, если шаблоном является `anyoromit`; в противном случае возвращает `false`;
- `getTemplateDef()` Эта операция возвращает определение шаблона.

8.2.5.2 TciNonValueTemplate

TciNonValueTemplate отображается на следующий интерфейс:

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

Методы:

- `isAny()` Возвращает `true`, если шаблоном является `any`; в противном случае возвращает `false`;
- `isAll()` Возвращает `true`, если шаблоном является `all`; в противном случае возвращает `false`;
- `getTemplateDef()` Эта операция возвращает определение шаблона.

8.2.5.3 TciValueList

TciValueList отображается на следующий интерфейс:

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

```
}
```

Методы:

- `size()` Возвращает число значений в этом списке;
- `isEmpty()` Возвращает `true`, если этот список не содержит значений;
- `get(int index)` Возвращает `Value` на заданной позиции.

8.2.5.4 TciValueDifference

TciValueDifference отображается на следующий интерфейс:

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

Методы:

- `getValue()` Возвращает значение этого элемента `TciValueDifference`;
- `getTciValueTemplate()` Возвращает шаблон этого элемента `TciValueDifference`;
- `getDescription()` Возвращает описание несоответствия.

8.2.5.5 TciValueDifferenceList

TciValueDifferenceList отображается на следующий интерфейс:

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

Методы:

- `size()` Возвращает число различий в этом списке;
- `isEmpty()` Возвращает `true`, если этот список не содержит различий;
- `get(int index)` Возвращает `TciValueDifference` на заданной позиции.

8.3 Константы

В языке Java было определено отображение констант. Все константы определяются как `public final static`, и к ним может быть обеспечен доступ из любого объекта любого пакета. Константы, определенные в данном разделе, не определяются в разделе языка IDL. Вместо этого они являются следствием спецификации типов на языке IDL интерфейса TCI, помеченных как собственные.

Следующие константы могут быть использованы для определения режима передачи параметров для параметров TTCN-3 (см. также п. 8.2.2.2).

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

Для особого значения параметра `null` закодированное значение параметра устанавливается на `null` в языке Java.

Для обработки значений используются следующие константы (см. также п. 8.2.2.4).

- `org.etsi.ttcn.tci.TciTypeClass.ADDRESS`;
- `org.etsi.ttcn.tci.TciTypeClass.ANYTYPE`;
- `org.etsi.ttcn.tci.TciTypeClass.BITSTRING`;
- `org.etsi.ttcn.tci.TciTypeClass.BOOLEAN`;

- org.etsi.ttcn.tci.TciTypeClass.CHARSTRING;
- org.etsi.ttcn.tci.TciTypeClass.COMPONENT;
- org.etsi.ttcn.tci.TciTypeClass.ENUMERATED;
- org.etsi.ttcn.tci.TciTypeClass.FLOAT;
- org.etsi.ttcn.tci.TciTypeClass.HEXSTRING;
- org.etsi.ttcn.tci.TciTypeClass.INTEGER;
- org.etsi.ttcn.tci.TciTypeClass.OBJID;
- org.etsi.ttcn.tci.TciTypeClass.OCTETSTRING;
- org.etsi.ttcn.tci.TciTypeClass.RECORD;
- org.etsi.ttcn.tci.TciTypeClass.RECORD_OF;
- org.etsi.ttcn.tci.TciTypeClass.SET;
- org.etsi.ttcn.tci.TciTypeClass.SET_OF;
- org.etsi.ttcn.tci.TciTypeClass.UNION;
- org.etsi.ttcn.tci.TciTypeClass.UNIVERSAL_CHARSTRING;
- org.etsi.ttcn.tci.TciTypeClass.VERDICT.

Для обработки компонентов используются следующие константы (см. также п. 8.2.2.5).

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

8.4 Отображение интерфейсов

В определении на языке IDL интерфейса TCI описываются четыре интерфейса **ТСИ-ТМ**, **ТСИ-СН**, **ТСИ-СД** и **ТСИ-ТЛ**. Эти операции определяются для различных направлений в указанном интерфейсе; то есть некоторые операции могут быть вызваны только выполняемым модулем TTCN-3 (TE), адаптером системы (SA) или адаптером платформы (PA) в администрировании и управлении тестом (TMC), в то время как другие операции могут быть вызваны со стороны TMC в TE. Это показано путем разделения интерфейсов TCI IDL на два субинтерфейса с приписанными Required или Provided в конце каждого субинтерфейса.

Таблица 3/Z.145 – Субинтерфейсы

| Вызывающий/ вызываемый | TE | TM | CD | CH | SA | PA | TL |
|---------------------------|-----------------------------|--------------------|--------------------|--------------------|--------------------------------|--------------------------------|--------------------|
| TE | – | TCI-TM Provided | TCI-CD Provided | TCI-CH Provided | См. Рек. МСЭ-T Z.144 [1] | См. Рек. МСЭ-T Z.144 [1] | TCI-TL Provided |
| TM | TCI-TM Required | – | – | – | – | – | TCI-TL Provided |
| CD | TCI-CD Required | – | – | – | – | – | TCI-TL Provided |
| CH | TCI-CH Required | – | – | – | – | – | TCI-TL Provided |
| SA | См. Рек. МСЭ-T Z.144 [1] | – | – | – | – | – | TCI-TL Provided |
| PA | См. Рек. МСЭ-T Z.144 [1] | – | – | – | – | – | TCI-TL Provided |
| TL | – | – | – | – | – | – | – |

Все методы, которые определены в этом интерфейсе, должны иметь характеристики, описанные в разделе 7.

8.4.1 Интерфейс TCI-TM

Интерфейс **ТСИ-ТМ** подразделяется на два субинтерфейса, интерфейс TCI-TM Provided, определяющий вызовы от TE к TM, и интерфейс TCI-TM Required, определяющий вызовы от TM к TE.

8.4.1.1 TCI-TM provided

Интерфейс TCI-TM Provided отображается на следующий интерфейс:

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

8.4.1.2 TCI-TM required

Интерфейс TCI-TM Required отображается на следующий интерфейс:

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

8.4.2 Интерфейс TCI-CD

Интерфейс TCI-CD подразделяется на два субинтерфейса, интерфейс TCI-CD Provided, определяющий вызовы от TE к CD, и интерфейс TCI-CD Required, определяющий вызовы от CD к TE.

8.4.2.1 TCI-CD provided

Интерфейс TCI-CD Provided отображается на следующий интерфейс:

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

8.4.2.2 TCI-CD required

Интерфейс TCI-CD Required отображается на следующий интерфейс:

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

```

public Type      getHexString();
public Type      getBitstring();
public Type      getOctetstring();
public Type      getVerdict();
public void      tciErrorReq(String message);
}

```

8.4.3 Интерфейс TCI-CH

Интерфейс TCI-CH подразделяется на два субинтерфейса, интерфейс TCI-CH Provided, определяющий вызовы от TE к CH, и интерфейс TCI-CH Required, определяющий вызовы от CH к TE.

8.4.3.1 TCI-CH provided

Интерфейс TCI-CH Provided отображается на следующий интерфейс:

```

// TciCHProvided
// TE -> CH
package org.etsi.ttcn.tci;
public interface TciCHProvided {
    public void tciSendConnected (TriPortId sender, TriComponentId receiver, Value
sendMessage) ;
    public void tciSendConnectedBC (TriPortId sender, Value sendMessage) ;
    public void tciSendConnectedMC (TriPortId sender, TriComponentIdList receivers,
Value sendMessage) ;
    public void tciCallConnected(TriPortId sender,
TriComponentId receiver,
TriSignatureId signature,
TciParameterList parameterList) ;
    public void tciCallConnectedBC(TriPortId sender,
TriSignatureId signature,
TciParameterList parameterList) ;
    public void tciCallConnectedMC(TriPortId sender,
TriComponentIdList receivers,
TriSignatureId signature,
TciParameterList parameterList) ;
    public void tciReplyConnected(TriPortId sender,
TriComponentId receiver,
TriSignatureId signature,
TciParameterList parameterList,
Value returnValue) ;
    public void tciReplyConnectedBC(TriPortId sender,
TriSignatureId signature,
TciParameterList parameterList,
Value returnValue) ;
    public void tciReplyConnectedMC(TriPortId sender,
TriComponentIdList receivers,
TriSignatureId signature,
TciParameterList parameterList,
Value returnValue) ;
    public void tciRaiseConnected(TriPortId sender,
TriComponentId receiver,
TriSignatureId signature,
Value except) ;
    public void tciRaiseConnectedBC(TriPortId sender,
TriSignatureId signature,
Value except) ;
    public void tciRaiseConnectedMC(TriPortId sender,
TriComponentIdList receivers,
TriSignatureId signature,
Value except) ;
    public TriComponentId tciCreateTestComponentReq(int kind, Type componentType, String name);
    public void tciStartTestComponentReq(TriComponentId comp,
TciBehaviourId behavior,
TciParameterList parameterList) ;
    public void tciStopTestComponentReq(TriComponentId comp) ;
}

```

```

public void      tciConnectReq(TriPortId fromPort, TriPortId toPort) ;
public void      tciDisconnectReq(TriPortId fromPort, TriPortId toPort) ;
public void      tciTestComponentTerminatedReq(TriComponentId comp, VerdictValue
verdict) ;
public boolean   tciTestComponentRunningReq(TriComponentId comp) ;
public TriComponentId tciGetMTCReq() ;
public void      tciMapReq(TriPortId fromPort, TriPortId toPort);
public void      tciUnmapReq(TriPortId fromPort, TriPortId toPort);
public void      tciExecuteTestCaseReq(TriComponentId testComponentId,
TriPortIdList tsiPortList);

public void      tciResetReq() ;
public boolean   tciTestComponentDoneReq(TriComponentId testComponentId) ;
public void      tciKillTestComponentReq(TriComponentId component)
public boolean   tciTestComponentAliveReq(TriComponentId component)
public boolean   tciTestComponentKilledReq(TriComponentId component)
}

```

8.4.3.2 TCI-CH required

Интерфейс TCI-CH Required отображается на следующий интерфейс:

```

// TciCHRequired
// CH -> TE
package org.etsi.ttcn.tci;
public interface TciCHRequired extends TciCDRequired {
    public void      tciEnqueueMsgConnected(TriPortId sender,
TriComponentId receiver,
Value receivedMessage) ;
    public void      tciEnqueueCallConnected(TriPortId sender,
TriComponentId receiver,
TriSignatureId signature,
TciParameterList parameterList) ;
    public void      tciEnqueueReplyConnected(TriPortId sender,
TriComponentId receiver,
TriSignatureId signature,
TciParameterList parameterList,
Value returnValue) ;
    public void      tciEnqueueRaiseConnected(TriPortId sender,
TriComponentId receiver,
TriSignatureId signature,
Value except) ;
    public TriComponentId tciCreateTestComponent(int kind, Type componentType, String name);
    public void      tciStartTestComponent(TriComponentId comp,
TciBehaviourId behavior,
TciParameterList parameterList) ;
    public void      tciStopTestComponent(TriComponentId comp) ;
    public void      tciConnect(TriPortId fromPort, TriPortId toPort) ;
    public void      tciDisconnect(TriPortId fromPort, TriPortId toPort) ;
    public void      tciTestComponentTerminated(TriComponentId comp, VerdictValue
verdict) ;
    public boolean   tciTestComponentRunning(TriComponentId comp);
    public boolean   tciTestComponentDone(TriComponentId comp);
    public TriComponentId tciGetMTC() ;
    public void      tciExecuteTestCase(TciTestCaseId TestCaseId, TriPortIdList
tsiPortList);
    public void      tciReset() ;
    public void      tciMap(TriPortId fromPort, TriPortId toPort);
    public void      tciUnmap(TriPortId fromPort, TriPortId toPort);
    public void      tciKillTestComponent(TriComponentId component)
    public boolean   tciTestComponentAlive(TriComponentId component)
    public boolean   tciTestComponentKilled(TriComponentId component)
}

```

8.4.4 Интерфейс TCI-TL

Интерфейс TCI-TL содержит только один субинтерфейс, интерфейс TCI-TL Provided, определяющий вызовы от TE, TM, CH, CD, SA и PA к TL.

8.4.4.1 TCI-TL provided

Интерфейс TCI-TL Provided отображается на следующий интерфейс:

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

```

        TriStatus transmissionFailure);
public void tliPrCall_c_BC(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue,
    TriStatus transmissionFailure);
public void tliPrCall_c_MC(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue,
    TriComponentIdList toList, TriStatus transmissionFailure);
public void tliPrGetCallDetected_m(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TriParameterList pars, TriAddress address);
public void tliPrGetCallDetected_c(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue,
    TriComponentId from);
public void tliPrGetCallMismatch_m(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature,
    TciParameterList parsValue, TciValueTemplate parsTpl, TciValueDifferenceList diffs,
    TriAddress address, TciValueTemplate addressTpl);
public void tliPrGetCallMismatch_c(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature,
    TciParameterList parsValue, TciValueTemplate parsTpl, TciValueDifferenceList diffs,
    TriComponentId from, TciNonValueTemplate fromTpl);
public void tliPrGetCall_m(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature,
    TciParameterList parsValue, TciValueTemplate parsTpl,
    TriAddress address, TciValueTemplate addressTpl);
public void tliPrGetCall_c(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature,
    TciParameterList parsValue, TciValueTemplate parsTpl,
    TriComponentId from, TciNonValueTemplate fromTpl);
public void tliPrReply_m(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
    TriAddress address, TriStatus encoderFailure,
    TriParameter repl, TriStatus transmissionFailure);
public void tliPrReply_m_BC(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
    TriStatus encoderFailure,
    TriParameter repl, TriStatus transmissionFailure);
public void tliPrReply_m_MC(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
    TriAddressList addresses, TriStatus encoderFailure,
    TriParameter repl, TriStatus transmissionFailure);
public void tliPrReply_c(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
    TriComponentId to, TriStatus transmissionFailure);
public void tliPrReply_c_BC(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
    TriStatus transmissionFailure);
public void tliPrReply_c_MC(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value replValue,
    TriComponentIdList toList, TriStatus transmissionFailure);
public void tliPrGetReplyDetected_m(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TriParameter repl, TriAddress address);
public void tliPrGetReplyDetected_c(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, Value replValue, TriComponentId from);
public void tliPrGetReplyMismatch_m(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue,
    Value replValue, TciValueTemplate replyTpl, TciValueDifferenceList diffs,
    TriAddress address, TciValueTemplate addressTpl);
public void tliPrGetReplyMismatch_c(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue,
    Value replValue, TciValueTemplate replyTpl, TciValueDifferenceList diffs,
    TriComponentId from, TciNonValueTemplate fromTpl);
public void tliPrGetReply_m(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue,
    Value replValue, TciValueTemplate replyTpl,
    TriAddress address, TciValueTemplate addressTpl);
public void tliPrGetReply_c(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue,
    Value replValue, TciValueTemplate replyTpl,
    TriComponentId from, TciNonValueTemplate fromTpl);
public void tliPrRaise_m(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,

```

```

        TriAddress address, TriStatus encoderFailure, TriException exc,
        TriStatus transmissionFailure);
public void tliPrRaise_m_BC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriStatus encoderFailure, TriException exc, TriStatus transmissionFailure);
public void tliPrRaise_m_MC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriAddressList addresses, TriStatus encoderFailure, TriException exc,
        TriStatus transmissionFailure);
public void tliPrRaise_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriComponentId to, TriStatus transmissionFailure);
public void tliPrRaise_c_BC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriStatus transmissionFailure);
public void tliPrRaise_c_MC(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue, Value excValue,
        TriComponentIdList toList, TriStatus transmissionFailure);
public void tliPrCatchDetected_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TriException exc, TriAddress address);
public void tliPrCatchDetected_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, Value excValue, TriComponentId from);
public void tliPrCatchMismatch_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value excValue, TciValueTemplate excTmpl, TciValueDifferenceList diffs,
        TriAddress address, TciValueTemplate addressTmpl);
public void tliPrCatchMismatch_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value excValue, TciValueTemplate excTmpl, TciValueDifferenceList diffs,
        TriComponentId from, TciNonValueTemplate fromTmpl);
public void tliPrCatch_m(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value excValue, TciValueTemplate excTmpl,
        TriAddress address, TciValueTemplate addressTmpl);
public void tliPrCatch_c(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue,
        Value excValue, TciValueTemplate excTmpl,
        TriComponentId from, TciNonValueTemplate fromTmpl);
public void tliPrCatchTimeoutDetected(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature);
public void tliPrCatchTimeout(String am, int ts, String src, int line, TriComponentId c,
        TriPortId port, TriSignatureId signature, TciParameterList parsValue);
public void tliCCreate(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp, String name);
public void tliCStart(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp, TciBehaviourId name, TciParameterList parsValue);
public void tliCRunning(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp, TBoolean status);
public void tliCAlive(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp, TBoolean status);
public void tliCStop(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp);
public void tliCKill(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId comp);
public void tliCDoneMismatch(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate compTmpl);
public void tliCDone(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate compTmpl);
public void tliCKilledMismatch(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate compTmpl);
public void tliCKilled(String am, int ts, String src, int line, TriComponentId c,
        TciNonValueTemplate compTmpl);
public void tliCTerminated(String am, int ts, String src, int line, TriComponentId c,
        VerdictValue verdict);
public void tliPConnect(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2);
public void tliPDisconnect(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2);
public void tliPMap(String am, int ts, String src, int line, TriComponentId c,
        TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2);
public void tliPUnmap(String am, int ts, String src, int line, TriComponentId c,

```

```

        TriComponentId c1, TriPortId port1, TriComponentId c2, TriPortId port2);
public void tliPClear(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port);
public void tliPStart(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port);
public void tliPStop(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port);
public void tliPHalt(String am, int ts, String src, int line, TriComponentId c,
    TriPortId port);
public void tliEncode(String am, int ts, String src, int line, TriComponentId c,
    Value val, TriStatus encoderFailure, TriMessage msg, String codec);
public void tliDecode(String am, int ts, String src, int line, TriComponentId c,
    TriMessage msg, TriStatus decoderFailure, Value val, String codec);
public void tliTimeoutDetected(String am, int ts, String src, int line, TriComponentId c,
    TriTimerId timer);
public void tliTimeoutMismatch(String am, int ts, String src, int line, TriComponentId c,
    TciNonValueTemplate timerTmpl);
public void tliTimeout(String am, int ts, String src, int line, TriComponentId c,
    TciNonValueTemplate timerTmpl);
public void tliTStart(String am, int ts, String src, int line, TriComponentId c,
    TriTimerId timer, TriTimerDuration dur);
public void tliTStop(String am, int ts, String src, int line, TriComponentId c,
    TriTimerId timer);
public void tliTRead(String am, int ts, String src, int line, TriComponentId c,
    TriTimerId timer, TriTimerDuration elapsed);
public void tliTRunning(String am, int ts, String src, int line, TriComponentId c,
    TriTimerId timer, TBoolean status);
public void tliSEnter(String am, int ts, String src, int line, TriComponentId c,
    String name, TciParameterList parsValue, String kind);
public void tliSLeave(String am, int ts, String src, int line, TriComponentId c,
    String name, Value returnValue, String kind);
public void tliVar(String am, int ts, String src, int line, TriComponentId c,
    String name, Value varValue);
public void tliModulePar(String am, int ts, String src, int line, TriComponentId c,
    String name, Value parValue);
public void tliGetVerdict(String am, int ts, String src, int line, TriComponentId c,
    VerdictValue verdict);
public void tliSetVerdict(String am, int ts, String src, int line, TriComponentId c,
    VerdictValue verdict);
public void tliLog(String am, int ts, String src, int line, TriComponentId c,
    TciValueList log);
public void tliAEnter(String am, int ts, String src, int line, TriComponentId c);
public void tliALeave(String am, int ts, String src, int line, TriComponentId c);
public void tliADefaults(String am, int ts, String src, int line, TriComponentId c);
public void tliAActivate(String am, int ts, String src, int line, TriComponentId c,
    String name, TciParameterList pars, Value ref);
public void tliADeactivate(String am, int ts, String src, int line, TriComponentId c,
    Value ref);
public void tliANomatch(String am, int ts, String src, int line, TriComponentId c);
public void tliARepet(String am, int ts, String src, int line, TriComponentId c);
public void tliAwait(String am, int ts, String src, int line, TriComponentId c);
}

```

8.5 Факультативные параметры

В разделе 7.3 определено, что зарезервированное значение используется для указания на отсутствие факультативного параметра. Для отображения языка Java значение null в этом языке используется для указания на отсутствие факультативного значения. Например, если в операции `tciReplyConnected` параметр значения опущен, то вызовом операции будет `tciReplyConnected(sender, receiver, signature, parameterList, null)`.

8.6 Инициализация ТСІ

Все методы являются нестатическими, то есть могут вызываться только операции над объектами. Поскольку в настоящей Рекомендации не определяются конкретные стратегии реализации для ТЕ, ТМ, CD и СН, то механизм того, каким образом ТЕ, ТМ, CD или СН могут получить информацию об обработке в соответствующих объектах, выходит за рамки данной Рекомендации.

Поставщики инструментальных средств должны предоставлять разработчикам ТМ, CD и СН методы регистрации ТЕ, ТМ, CD и СН для их соответствующего партнера по связи.

8.7 Обработка ошибок

Все операции, вызываемые из ТМ, СН или CD, которые возвращают значение, были успешными. Если объектом ТЕ определена ситуация с ошибками, то об ошибке тестового примера будет сообщено пользователю путем использования процедур, как определено в п. 7.3.1.2.5 (tciError). Если операция, вызванная ТЕ в ТМ, СН, CD или TL, приводит к ошибке, то об этой ситуации с ошибкой должно быть сообщено ТЕ путем использования процедур, как определено в п. 7.3.2.1.12 (tciErrorReq).

Кроме указанной обработки ошибок в данном отображении языка Java не определена какая-либо дополнительная обработка ошибок. В частности, не определены механизмы обработки особых состояний.

9 Отображение данных языка ANSI-C

9.1 Введение

В данном разделе приведено описание отображения языка ANSI-C интерфейса TRI для данных интерфейса TCI, описанных в п. 7.2, и для операций интерфейса TCI, описанных в п. 7.3.

9.2 Интерфейсы значений

| Интерфейс TCI IDL | Представление в ANSI-C | Примечания и комментарии |
|--|--|---|
| Type | | |
| TciModuleIdType getDefiningModule() | TciModuleIdType tciGetDefiningModule(TciType inst) | |
| TString getName() | String tciGetName(TciType inst) | Повторно используемый строковый тип из IDL (Рекомендация OMG) |
| TciTypeClassType getTypeClass() | TciTypeClassType tciGetTypeClass(TciType inst) | |
| Value newInstance() | TciValue tciNewInstance(TciType inst) | |
| TString getTypeEncoding() | String tciGetTypeEncoding(TciType inst) | |
| TStringSeq getTypeExtension() | String* getTypeExtension(TciType inst) | |
| TString getTypeEncodingVariant() | String tciGetTypeEncodingVariant(TciType inst) | |
| Value | | |
| TString getValueEncoding() | String tciGetValueEncoding(TciValue inst) | |
| TString getValueEncodingVariant() | String tciGetValueEncodingVariant(TciValue inst) | |
| Type getType() | TciType tciGetType(TciValue inst) | |
| TBoolean notPresent() | Boolean tciNotPresent(TciValue inst) | Повторно используемый булев тип из IDL (Рекомендация OMG) |
| IntegerValue | | |
| TInteger getInt() | String tciGetIntAbs(TciValue inst) | Возвращает целое абсолютное значение (с основанием 10) как строку ASCII. |
| | unsigned long int tciGetIntNumberOfDigits(TciValue inst) | Возвращает число цифр в целом значении. |
| | Boolean tciGetIntSign(TciValue inst) | Возвращает true, если число положительное, в противном случае – false |
| | char tciGetIntDigit(TciValue inst, unsigned long int position) | Возвращает значение цифры на позиции 'position', где позиция 0 является наименьшей значащей цифрой. |

| Интерфейс TCI IDL | Представление в ANSI-C | Примечания и комментарии |
|--|---|--|
| void setInt(in TInteger value) | void tciSetIntAbs(TciValue inst, String value) | Устанавливает абсолютное значение целого числа (с основанием 10) через строку ASCII. Первая цифра не должна быть 0, если только значение не равно 0. |
| | void tciSetIntNumberOfDigits (TciValue inst, unsigned long int dig_num) | Устанавливает число цифр в целом значении . |
| | void tciSetIntSign (TciValue inst, Boolean sign) | Устанавливает знак на + (true) или на - (false). |
| | void tciSetIntDigit (TciValue inst, unsigned long int position, char digit) | Устанавливает значение цифры на позиции 'position', где позиция 0 является наименьшей значащей цифрой. |
| FloatValue | | |
| TFloat getFloat() | double tciGetFloatValue(TciValue inst) | |
| void setFloat (in TFloat value) | void tciSetFloatValue(TciValue inst, double value) | |
| BooleanValue | | |
| TBoolean getBoolean() | Boolean tciGetBooleanValue(TciValue inst) | |
| void setBoolean (in TBoolean value) | void tciSetBooleanValue (TciValue inst, Boolean value) | |
| ObjidValue | | |
| TObjid getObjid() | TciObjidValue tciGetTciObjidValue(TciValue inst) | |
| void setObjid(in TObjid value) | void tciSetObjidValue(TciValue inst, TciObjidValue value) | |
| CharstringValue | | |
| TString getString() | TciCharStringValue tciGetCStringValue(TciValue inst) | |
| void setString(in TString value) | void tciSetCStringValue (TciValue inst, TciCharStringValue value) | |
| TChar getChar (in TInteger position) | char tciGetCStringCharValue (TciValue inst, long int position) | |
| void setChar (in TInteger position, in char value) | void tciSetCStringCharValue (TciValue inst, long int position, char value) | |
| TInteger getLength() | unsigned long int tciGetCStringLength (TciValue inst) | |
| void setLength(in TInteger len) | void tciSetCStringLength (TciValue inst, unsigned long int len) | |
| UniversalCharstringValue | | |
| TString getString() | TciUCStringValue tciGetUCStringValue(TciValue inst) | |
| void setString (in TString value) | void tciSetUCStringValue (TciValue inst, TciUCStringValue value) | |
| TUniversalChar getChar (in TInteger position) | void tciGetUCStringCharValue (TciValue inst, unsigned long int position, TciUCValue result) | |
| void setChar (in TInteger position, in TUniversalChar value) | void tciSetUCStringCharValue (TciValue inst, unsigned long int position, TciUCValue value) | |
| TInteger getLength() | unsigned long int tciGetUCStringLength(TciValue inst) | |

| Интерфейс TCI IDL | Представление в ANSI-C | Примечания и комментарии |
|---|--|--|
| void setLength (in TInteger len) | void tciSetUCStringLength (TciValue inst, unsigned long int len) | |
| BitstringValue | | |
| TString getString() | String tciGetBStringValue(TciValue inst) | |
| void setString(in TString value) | void tciSetBStringValue (TciValue inst, String value) | |
| TChar getBit (in integer position) | int tciGetBStringBitValue (TciValue inst, long int position) | |
| void setBit (in TInteger position, in TInteger value) | void tciSetBStringBitValue (TciValue inst, unsigned long int position, int value) | |
| TInteger getLength() | unsigned long int tciGetBStringLength(TciValue inst) | |
| void setLength (in TInteger len) | void tciSetBStringLength (TciValue inst, long int len) | |
| HexstringValue | | |
| TString getString() | String tciGetHStringValue(TciValue inst) | |
| void setString (in TString value) | void tciSetHStringValue (TciValue inst, String value) | |
| TChar getHex (in TInteger position) | int tciGetHStringHexValue (TciValue inst, unsigned long int position) | |
| void setBit (in TInteger position, in TInteger value) | void tciSetHStringHexValue (TciValue inst, unsigned long int position, int value) | |
| TInteger getLength() | long int tciGetHStringLength(TciValue inst) | |
| void setLength(in TInteger len) | void tciSetHStringLength (TciValue inst, unsigned long int len) | |
| OctetstringValue | | |
| TString getString() | String tciGetOStringValue(TciValue inst) | |
| void setString(in TString value) | void tciSetOStringValue (TciValue inst, String value) | |
| TChar getOctet(in TInteger position) | int tciGetOStringOctetValue (TciValue inst, unsigned long int position) | |
| void setOctet (in TInteger position, in TInteger value) | void tciSetOStringOctetValue (TciValue inst, unsigned long int position, int value) | |
| TInteger getLength() | unsigned long int tciGetOStringLength(TciValue inst) | |
| void setLength(in TInteger len) | void tciSetOStringLength (TciValue inst, unsigned long int len) | |
| RecordValue | | |
| Value getField (in TString fieldName) | TciValue tciGetRecFieldValue (TciValue inst, String fieldName) | |
| void setField (in TString fieldName, in Value value) | void tciSetRecFieldValue (TciValue inst, String fieldName, TciValue value) | |
| TString[] getFieldNames() | char** tciGetRecFieldNames(TciValue inst) | Возвращает заканчивающийся на NULL массив имен полей. |
| RecordOfValue | | |
| Value getField (in TInteger position) | TciValue tciGetRecOfFieldValue (TciValue inst, unsigned long int position) | |

| Интерфейс TCI IDL | Представление в ANSI-C | Примечания и комментарии |
|--|---|--|
| void setField (in TInteger position, in Value value) | void tciSetRecOfFieldValue (TciValue inst, unsigned long int position, TciValue value) | |
| void appendField (in Value value) | void tciAppendRecOfFieldValue (TciValue inst, TciValue value) | |
| Type getElementType() | TciType tciGetRecOfElementType (TciValue inst) | |
| TInteger getLength() | unsigned long int tciGetRecOfLength(TciValue inst) | |
| void setLength (in TInteger len) | void tciSetRecOfLength (TciValue inst, unsigned long int len) | |
| UnionValue | | |
| Value getVariant (in TString variantName) | TciValue tciGetUnionVariant (TciValue inst, String variantName) | |
| void setVariant (in TString variantName, in Value value) | void tciSetUnionVariant (TciValue inst, String variantName, TciValue value) | |
| TString getPresentVariantName() | String tciGetUnionPresentVariantName (TciValue inst) | |
| TString[] getVariantNames() | char** tciGetUnionVariantNames (TciValue inst) | Возвращает заканчивающийся на NULL массив имен полей. |
| Ея85numeratedValue | | |
| TString getEnum() | String tciGetEnumValue (TciValue inst) | |
| void setEnum (in TString enumValue) | void tciSetEnumValue (TciValue inst, String enumValue) | |
| VerdictValue | | |
| TInteger getVerdict() | int tciGetVerdictValue (TciValue inst) | |
| void setVerdict (in TInteger verdict) | void tciSetVerdictValue (TciValue inst, int verdict) | |
| AddressValue | | |
| Value getAddress() | TciValue tciGetAddressValue (TciValue inst) | |
| void setAddress (in Value value) | void tciSetAddressValue (TciValue inst, TciValue value) | |

9.3 Интерфейс регистрации

| Интерфейс TCI IDL | Представление в ANSI-C | Примечания и комментарии |
|--------------------------|---|---|
| TciValueTemplate | | |
| TBoolean isOmit() | Boolean tciIsOmit (TciValueTemplate inst) | Повторно используемый булев тип из IDL (Рекомендация OMG) |
| TBoolean isAny() | Boolean tciIsAny (TciValueTemplate inst) | Повторно используемый булев тип из IDL (Рекомендация OMG) |
| TBoolean isAnyOrOmit() | Boolean tciIsAnyOrOmit (TciValueTemplate inst) | Повторно используемый булев тип из IDL (Рекомендация OMG) |
| TString getTemplateDef() | String tciGetTemplateDef (TciValueTemplate inst) | Повторно используемый строковый тип из IDL (Рекомендация OMG) |

| Интерфейс TCI IDL | Представление в ANSI-C | Примечания и комментарии |
|----------------------------|--|---|
| TciNonValueTemplate | | |
| TBoolean isAny() | Boolean tciIsAny(TciNonValueTemplate inst) | Повторно используемый булев тип из IDL (Рекомендация OMG) |
| TBoolean isAll() | Boolean tciIsAll(TciNonValueTemplate inst) | Повторно используемый булев тип из IDL (Рекомендация OMG) |
| TString getTemplateDef() | String tciGetTemplateDef(TciNonValueTemplate inst) | Повторно используемый строковый тип из IDL (Рекомендация OMG) |

9.4 Интерфейсы операций

| TCI-TM required | | |
|--|--|--|
| void tciRootModule (in TciModuleIdType moduleId) | void tciRootModule(String moduleId) | |
| TciModuleParameterListType tciGetModuleParameters (in TciModuleIdType moduleName) | TciModuleParameterListType tciGetModuleParameters (TciModuleIdType moduleName) | |
| TciTestCaseIdListType tciGetTestCases() | TciTestCaseIdListType tciGetTestCases() | |
| TciParameterTypeListType tciGetTestCaseParameters (in TciTestCaseIdType testCaseId) | TciParameterTypeListType tciGetTestCaseParameters (TciTestCaseIdType testCaseId) | |
| TriPortIdListType tciGetTestCaseTSI (in TciTestCaseIdType testCaseId) | TriPortIdList tciGetTestCaseTSI (TciTestCaseIdType testCaseId) | |
| void tciStartTestCase (in TciTestCaseIdType testCaseId, in TciParameterListType parameterlist) | void tciStartTestCase (TciTestCaseIdType testCaseId, TciParameterListType parameterlist) | |
| void tciStopTestCase() | void tciStopTestCase() | |
| TriComponentId tciStartControl() | TriComponentId tciStartControl() | |
| void tciStopControl() | void tciStopControl() | |
| TCI-TM provided | | |
| void tciTestCaseStarted (in TciTestCaseIdType testCaseId, in TciParameterListType parameterList, in Tfloat timer) | void tciTestCaseStarted (TciTestCaseIdType testCaseId, TciParameterListType parameterList, double timer) | |
| void tciTestCaseTerminated (in VerdictValue verdict, in TciParameterListType parameterlist) | void tciTestCaseTerminated (TciVerdictValue verdict, TciParameterListType parameterlist) | |
| void tciControlTerminated() | void tciControlTerminated() | |
| Value tciGetModulePar (in TciModuleParameterIdType parameterId) | tciValue tciGetModulePar (TciModuleParameterIdType parameterId) | |
| void tciLog(in TString message) | void tciLog(String message) | |
| void tciError(in TString message) | void tciError(String message) | |
| TCI-CD required | | |
| Type getTypeForName (in String typeName) | TciType tciGetTypeForName (String typeName) | |
| Type getInteger() | TciType tciGetIntegerType() | |
| Type getFloat() | TciType tciGetFloatType() | |
| Type getBoolean() | TciType tciGetBooleanType() | |
| Type getChar() | TciType tciGetCharType() | |
| Type getUniversalChar() | TciType tciGetUniversalCharType() | |
| Type getObjid() | TciType tciGetTciObjidType() | |
| Type getCharstring() | TciType tciGetTciCharstringType() | |
| Type getUniversalCharstring() | TciType tciGetUniversalCharstringType() | |
| Type getHexstring() | TciType tciGetHexstringType() | |

| | | |
|--|---|---|
| Type getBitstring() | TciType tciGetBitstringType() | |
| Type getOctetstring() | TciType tciGetOctetstringType() | |
| Type getVerdict() | TciType tciGetVerdictType() | |
| void tciErrorReq(in String message) | void tciErrorReq(String message) | |
| TCI-CD provided | | |
| Value decode (in TriMessageType message, in Type decodingHypothesis) | TciValue tciDecode (BinaryString message, TciType decHypothesis) | Повторно используемый тип BinaryString из TRI |
| TriMessageType encode (in Value value) | BinaryString tciEncode(TciValue value) | |
| TCI-CH required | | |
| void tciEnqueueMsgConnected (in TriPortIdType sender, in TriComponentIdType receiver, in Value rcvdMessage) | void tciEnqueueMsgConnected (TriPortId sender, TriComponentId receiver, TciValue rcvdMessage) | |
| void tciEnqueueCallConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList) | void tciEnqueueCallConnected (TriPortId sender, TriComponentId receiver, TriSignatureId signature, TciParameterListType parameterList) | |
| void tciEnqueueReplyConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue) | void tciEnqueueReplyConnected (TriPortId sender, TriComponentId receiver, TriSignatureId signature, TciParameterListType parameterList, TciValue returnValue) | |
| void tciEnqueueRaiseConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in Value exception) | void tciEnqueueRaiseConnected (TriPortId sender, TriComponentId receiver, TriSignatureIdType signature, TciValue exception) | |
| TriComponentIdType tciCreateTestComponent (in TciTestComponentKindType kind in Type componentType, in TString name) | TriComponentId tciCreateTestComponent (TciTestComponentKindType kind, TciType componentType, String name) | |
| void tciStartTestComponent (in TriComponentIdType component, in TciBehaviourIdType behavior, in TciParamaterListType parameterList) | void tciStartTestComponent (TriComponentId component, TciBehaviourIdType behavior, TciParamaterListType parameterList) | |
| void tciStopTestComponent (in TriComponentIdType component) | void tciStopTestComponent (TriComponentId component) | |
| void tciConnect (in TriPortIdType fromPort, in TriPortIdType toPort) | void tciConnect (TriPortId fromPort, TriPortId toPort) | |
| void tciDisconnect (in TriPortIdType fromPort, in TriPortIdType toPort) | void tciDisconnect (TriPortId fromPort, TriPortId toPort) | |
| void tciMap (in TriPortIdType fromPort, in TriPortIdType toPort) | void tciMap (TriPortId fromPort, TriPortId toPort) | |
| void tciUnmap (in TriPortIdType fromPort, in TriPortIdType toPort) | void tciUnmap (TriPortId fromPort, TriPortId toPort) | |
| void tciTestComponentTerminated (in TriComponentIdType component, in VerdictValue verdict) | void tciTestComponentTerminated (TriComponentId component, TciVerdictValue verdict) | |
| boolean tciTestComponentRunning (in TriComponentIdType component) | Boolean tciTestComponentRunning (TriComponentId component) | |
| boolean tciTestComponentDone (in TriComponentIdType component) | boolean tciTestComponentDone (TriComponentId component) | |
| TriComponentIdType tciGetMTC() | TriComponentId tciGetMTC() | |

| | | |
|--|---|--|
| void tciExecuteTestCase (in TciTestCaseIdType testCaseId, in TriPortIdListType tsiPortList) | void tciExecuteTestCase (TciTestCaseIdType testCaseId, TriPortIdListType tsiPortList) | |
| void tciReset() | void tciReset() | |
| void tciKillTestComponent (in TriComponentIdType component) | void tciKillTestComponent (TriComponentIdType component) | |
| TBoolean tciTestComponentAlive (in TriComponentIdType component) | Boolean tciTestComponentAlive (TriComponentIdType component) | |
| TBoolean tciTestComponentKilled (in TriComponentIdType component) | Boolean tciTestComponentKilled (TriComponentIdType component) | |
| TCI-CH provided | | |
| void tciSendConnected (in TriPortIdType sender, in TriComponentIdType receiver, in Value sendMessage) | void tciSendConnected (TriPortIdType sender, TriComponentIdType receiver, TciValue sendMessage) | |
| void tciSendConnectedBC (in TriPortIdType sender, in Value sendMessage) | void tciSendConnectedBC (TriPortIdType sender, TciValue sendMessage) | |
| void tciSendConnectedMC (in TriPortIdType sender, in TriComponentIdListType receivers, in Value sendMessage) | void tciSendConnectedMC (TriPortIdType sender, TriComponentIdListType receivers, TciValue sendMessage) | |
| void tciCallConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList) | void tciCallConnected (TriPortIdType sender, TriComponentIdType receiver, TriSignatureIdType signature, TciParameterListType parameterList) | |
| void tciCallConnectedBC (in TriPortIdType sender, in TriSignatureIdType signature, in TciParameterListType parameterList) | void tciCallConnectedBC (TriPortIdType sender, TriSignatureIdType signature, TciParameterListType parameterList) | |
| void tciCallConnectedMC (in TriPortIdType sender, in TriComponentIdListType receivers, in TriSignatureIdType signature, in TciParameterListType parameterList) | void tciCallConnectedMC (TriPortIdType sender, TriComponentIdListType receivers, TriSignatureIdType signature, TciParameterListType parameterList) | |
| void tciReplyConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue) | void tciReplyConnected (TriPortIdType sender, TriComponentIdType receiver, TriSignatureIdType signature, TciParameterListType parameterList, TciValue returnValue) | |
| void tciReplyConnectedBC (in TriPortIdType sender, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue) | void tciReplyConnectedBC (TriPortIdType sender, TriSignatureIdType signature, TciParameterListType parameterList, TciValue returnValue) | |
| void tciReplyConnectedMC (in TriPortIdType sender, in TriComponentIdListType receivers, in TriSignatureIdType signature, in TciParameterListType parameterList, in Value returnValue) | void tciReplyConnectedMC (TriPortIdType sender, TriComponentIdListType receivers, TriSignatureIdType signature, TciParameterListType parameterList, TciValue returnValue) | |
| void tciRaiseConnected (in TriPortIdType sender, in TriComponentIdType receiver, in TriSignatureIdType signature, in Value exception) | void tciRaiseConnected (TriPortIdType sender, TriComponentIdType receiver, TriSignatureIdType signature, TciValue exception) | |
| void tciRaiseConnectedBC (in TriPortIdType sender, in TriSignatureIdType signature, in Value exception) | void tciRaiseConnectedBC (TriPortIdType sender, TriSignatureIdType signature, TciValue exception) | |

| | | |
|---|---|--|
| void tciRaiseConnectedMC (in TriPortIdType sender, in TriComponentIdListType receivers, in TriSignatureIdType signature, in Value exception) | void tciRaiseConnectedMC (TriPortId sender, TriComponentIdList receivers, TriSignatureId signature, TciValue exception) | |
| TriComponentIdType tciCreateTestComponentReq (in TciTestComponentKindType kind, in Type componentType, in TString name) | TriComponentId tciCreateTestComponentReq (TciTestComponentKindType kind, TciType componentType, String name) | |
| void tciStartTestComponentReq (in TriComponentIdType component, in TciBehaviourIdType behavior, in TciParamaterListType parameterList) | void tciStartTestComponentReq (TriComponentId component, TciBehaviourIdType behavior, TciParamaterListType parameterList) | |
| void tciStopTestComponentReq (in TriComponentIdType component) | void tciStopTestComponentReq (TriComponentId component) | |
| void tciConnectReq (in TriPortIdType fromPort, in TriPortIdType toPort) | void tciConnectReq (TriPortId fromPort, TriPortId toPort) | |
| void tciDisconnectReq (in TriPortIdType fromPort, in TriPortIdType toPort) | void tciDisconnectReq (TriPortId fromPort, TriPortId toPort) | |
| void tciMapReq (in TriPortIdType fromPort, in TriPortIdType toPort) | void tciMapReq (TriPortId fromPort, TriPortId toPort) | |
| void tciUnmapReq (in TriPortIdType fromPort, in TriPortIdType toPort) | void tciUnmapReq (TriPortId fromPort, TriPortId toPort) | |
| void tciTestComponentTerminatedReq (in TriComponentIdType component, in VerdictValue verdict) | void tciTestComponentTerminatedReq (TriComponentId component, TciVerdictValue verdict) | |
| boolean tciTestComponentRunningReq (in TriComponentIdType component) | Boolean tciTestComponentRunningReq (TriComponentId component) | |
| boolean tciTestComponentDoneReq (in TriComponentIdType component) | Boolean tciTestComponentDoneReq (TriComponentId component) | |
| TriComponentIdType tciGetMTCReq() | TriComponentId tciGetMTCReq() | |
| void tciExecuteTestCaseReq (in TciTestCaseIdType testCaseId, in TriPortIdListType tsiPortList) | void tciExecuteTestCaseReq (TciTestCaseIdType testCaseId, TriPortIdList tsiPortList) | |
| void tciResetReq() | void tciResetReq() | |
| void tciKillTestComponentReq (in TriComponentIdType component) | void tciKillTestComponentReq (TriComponentId component) | |
| TBoolean tciTestComponentAliveReq (in TriComponentIdType component) | TBoolean tciTestComponentAliveReq (TriComponentId component) | |
| TBoolean tciTestComponentKilledReq (in TriComponentIdType component) | TBoolean tciTestComponentKilledReq (TriComponentId component) | |
| TCI-TL provided | | |
| void tliTcExecute (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseldType tcId, in TriParameterListType pars, in TriTimerDurationType dur) | void tliTcExecute (String am, int ts, String src, int line, TriComponentId c, TciTestCaseldType tcId, TriParameterList pars, TriTimerDuration dur) | |
| void tliTcStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseldType tcId, in TriParameterListType pars, in TriTimerDurationType dur) | void tliTcStart (String am, int ts, String src, int line, TriComponentId c, TciTestCaseldType tcId, TriParameterList pars, TriTimerDuration dur) | |
| void tliTcStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | void tliTcStop (String am, int ts, String src, int line, TriComponentId c) | |

| | | |
|---|---|--|
| void tliTcStarted (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in TriTimerDurationType dur) | void tliTcStarted (String am, int ts, String src, int line, TriComponentId c, TciTestCaseIdType tcId, TriParameterListType pars, TriTimerDuration dur) | |
| void tliTcTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseIdType tcId, in TriParameterListType pars, in VerdictValue outcome) | void tliTcTerminated (String am, int ts, String src, int line, TriComponentId c, TciTestCaseIdType tcId, TriParameterListType pars, VerdictValue outcome) | |
| void tliCtrlStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | void tliCtrlStart (String am, int ts, String src, int line, TriComponentId c) | |
| void tliCtrlStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | void tliCtrlStop (String am, int ts, String src, int line, TriComponentId c) | |
| void tliCtrlTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | void tliCtrlTerminated (String am, int ts, String src, int line, TriComponentId c) | |
| void tliMSend_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriAddressType address, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure) | void tliMSend_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriAddress address, TriStatus encoderFailure, TriMessage msg, TriStatus transmissionFailure) | |
| void tliMSend_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure) | void tliMSend_m_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriStatus encoderFailure, TriMessage msg, TriStatus transmissionFailure) | |
| void tliMSend_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure) | void tliMSend_m_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriAddressList addresses, TriStatus encoderFailure, TriMessage msg, TriStatus transmissionFailure) | |
| void tliMSend_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdType to, in TriStatusType transmissionFailure) | void tliMSend_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriComponentId to, TriStatus transmissionFailure) | |
| void tliMSend_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriStatusType transmissionFailure) | void tliMSend_c_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriStatus transmissionFailure) | |
| void tliMSend_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure) | void tliMSend_c_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriComponentIdList toList, TriStatus transmissionFailure) | |

| | | |
|---|---|--|
| <p>void tliMDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriMessageType msg, in TriAddressType address)</p> | <p>void tliMDetected_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriMessage msg, TriAddress address)</p> | |
| <p>void tliMDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdType from)</p> | <p>void tliMDetected_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TriComponentId from)</p> | |
| <p>void tliMMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)</p> | <p>void tliMMismatch_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TciValueDifferenceList diffs, TriAddress address, TciValueTemplate addressTmpl)</p> | |
| <p>void tliMMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)</p> | <p>void tliMMismatch_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TciValueDifferenceList diffs, TriComponentId from, TciNonValueTemplate fromTmpl)</p> | |
| <p>void tliMReceive_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)</p> | <p>void tliMReceive_m(String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TriAddress address, TciValueTemplate addressTmpl)</p> | |
| <p>void tliMReceive_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)</p> | <p>void tliMReceive_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, Value msgValue, TciValueTemplate msgTmpl, TriComponentId from, TciNonValueTemplate fromTmpl)</p> | |
| <p>void tliPrCall_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriAddressType address, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)</p> | <p>void tliPrCall_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriAddress address, TriStatus encoderFailure, TriParameterListType pars, TriStatus transmissionFailure)</p> | |
| <p>void tliPrCall_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)</p> | <p>void tliPrCall_m_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriStatus encoderFailure, TriParameterListType pars, TriStatus transmissionFailure)</p> | |

| | | |
|--|--|--|
| <p>void tliPrCall_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)</p> | <p>void tliPrCall_m_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriAddressList addresses, TriStatus encoderFailure, TriParameterList pars, TriStatus transmissionFailure)</p> | |
| <p>void tliPrCall_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdType to, in TriStatusType transmissionFailure)</p> | <p>void tliPrCall_c (String am, int ts, String src int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriComponentId to, TriStatus transmissionFailure)</p> | |
| <p>void tliPrCall_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriStatusType transmissionFailure)</p> | <p>void tliPrCall_c_BC (String am, int ts, String src int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriStatus transmissionFailure)</p> | |
| <p>void tliPrCall_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)</p> | <p>void tliPrCall_c_MC (String am, int ts, String src int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TriComponentIdList toList, TriStatus transmissionFailure)</p> | |
| <p>void tliPrGetCallDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriParameterListType pars, in TriAddressType address)</p> | <p>void tliPrGetCallDetected_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TriParameterList pars, TriAddress address)</p> | |
| <p>void tliPrGetCallDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriComponentIdType from)</p> | <p>void tliPrGetCallDetected_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterList parsValue, TriComponentId from)</p> | |
| <p>void tliPrGetCallMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTpl)</p> | <p>void tliPrGetCallMismatch_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TciValueTemplate parsTpl, TciValueDifferenceList diffs, TriAddress address, TciValueTemplate addressTpl)</p> | |
| <p>void tliPrGetCallMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTpl)</p> | <p>void tliPrGetCallMismatch_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TciValueTemplate parsTpl, TciValueDifferenceList diffs, TriComponentId from, TciNonValueTemplate fromTpl)</p> | |

| | | |
|---|--|--|
| <p>void tliPrGetCall_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)</p> | <p>void tliPrGetCall_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TciValueTemplate parsTmpl, TriAddress address, TciValueTemplate addressTmpl)</p> | |
| <p>void tliPrGetCall_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TciValueTemplate parsTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)</p> | <p>void tliPrGetCall_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, TciValueTemplate parsTmpl, TriComponentId from, TciNonValueTemplate fromTmpl)</p> | |
| <p>void tliPrReply_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriAddressType address, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)</p> | <p>void tliPrReply_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureIdType signature, TciParameterListType parsValue, Value replValue, TriAddress address, TriStatus encoderFailure, TriParameterType repl, TriStatus transmissionFailure)</p> | |
| <p>void tliPrReply_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)</p> | <p>void tliPrReply_m_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureIdType signature, TciParameterListType parsValue, Value replValue, TriStatus encoderFailure, TriParameterType repl, TriStatus transmissionFailure)</p> | |
| <p>void tliPrReply_m_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriAddressListType addresses, in TriStatusType encoderFailure, in TriParameterType repl, in TriStatusType transmissionFailure)</p> | <p>void tliPrReply_m_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureIdType signature, TciParameterListType parsValue, Value replValue, TriAddressListType addresses, TriStatus encoderFailure, TriParameterType repl, TriStatus transmissionFailure)</p> | |
| <p>void tliPrReply_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriComponentIdType to, in TriStatusType transmissionFailure)</p> | <p>void tliPrReply_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TriComponentId to, TriStatus transmissionFailure)</p> | |
| <p>void tliPrReply_c_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriStatusType transmissionFailure)</p> | <p>void tliPrReply_c_BC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TriStatus transmissionFailure)</p> | |

| | | |
|--|---|--|
| <p>void tliPrReply_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)</p> | <p>void tliPrReply_c_MC (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TriComponentIdList toList, TriStatus transmissionFailure)</p> | |
| <p>void tliPrGetReplyDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TriParameterType repl, in TriAddressType address)</p> | <p>void tliPrGetReplyDetected_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TriParameter repl, TriAddress address)</p> | |
| <p>void tliPrGetReplyDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in Value replValue, in TriComponentIdType from)</p> | <p>void tliPrGetReplyDetected_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, Value replValue, TriComponentId from)</p> | |
| <p>void tliPrGetReplyMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTmpl)</p> | <p>void tliPrGetReplyMismatch_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TciValueTemplate replyTmpl, TciValueDifferenceList diffs, TriAddress address, TciValueTemplate addressTmpl)</p> | |
| <p>void tliPrGetReplyMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)</p> | <p>void tliPrGetReplyMismatch_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TciValueTemplate replyTmpl, TciValueDifferenceList diffs, TriComponentId from, TciNonValueTemplate fromTmpl)</p> | |
| <p>void tliPrGetReply_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)</p> | <p>void tliPrGetReply_m (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TciValueTemplate replyTmpl, TriAddress address, TciValueTemplate addressTmpl)</p> | |
| <p>void tliPrGetReply_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)</p> | <p>void tliPrGetReply_c (String am, int ts, String src, int line, TriComponentId c, TriPortId port, TriSignatureId signature, TciParameterListType parsValue, Value replValue, TciValueTemplate replyTmpl, TriComponentId from, TciNonValueTemplate fromTmpl)</p> | |

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

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

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

| | | |
|--|--|--|
| void tliEncode (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value val, in TriStatusType encoderFailure, in TriMessageType msg, in TString codec) | void tliEncode (String am, int ts, String src, int line, TriComponentId c, Value val, TriStatus encoderFailure, TriMessage msg, String codec) | |
| void tliDecode (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriMessageType msg, in TriStatusType decoderFailure, in Value val, in TString codec) | void tliDecode (String am, int ts, String src, int line, TriComponentId c, TriMessage msg, TriStatus decoderFailure, Value val, String codec) | |
| void tliTimeoutDetected (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer) | void tliTimeoutDetected (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer) | |
| void tliTimeoutMismatch (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTpl) | void tliTimeoutMismatch (String am, int ts, String src, int line, TriComponentId c, TciNonValueTemplate timerTpl) | |
| void tliTimeout (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTpl) | void tliTimeout (String am, int ts, String src, int line, TriComponentId c, TciNonValueTemplate timerTpl) | |
| void tliTStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType dur) | void tliTStart (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer, TriTimerDuration dur) | |
| void tliTStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer) | void tliTStop (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer) | |
| void tliTRead (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType elapsed) | void tliTRead (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer, TriTimerDuration elapsed) | |
| void tliTRunning (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TBoolean status) | void tliTRunning (String am, int ts, String src, int line, TriComponentId c, TriTimerId timer, Boolean status) | |
| void tliSEnter (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in TciParameterListType parsValue, in TString kind) | void tliSEnter (String am, int ts, String src, int line, TriComponentId c, String name, TciParameterListType parsValue, String kind) | |
| void tliSLeave (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value returnValue, in TString kind) | void tliSLeave (String am, int ts, String src, int line, TriComponentId c, String name, Value returnValue, String kind) | |
| void tliVar (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value varValue) | void tliVar (String am, int ts, String src, int line, TriComponentId c, String name, Value varValue) | |
| void tliModulePar (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value parValue) | void tliModulePar (String am, int ts, String src, int line, TriComponentId c, String name, Value parValue) | |
| void tliGetVerdict (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict) | void tliGetVerdict (String am, int ts, String src, int line, TriComponentId c, VerdictValue verdict) | |

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

9.5 Данные

| ADT в IDL для TCI | Представление в ANSI-C (Определение типа) | Примечания и комментарии |
|-----------------------------|--|---|
| TciModuleIdType | QualifiedName | |
| TciModuleParameterType | typedef struct TciModuleParameterType { String parName; TciValue defaultValue; } TciModuleParameterType; | |
| TciModuleParameterListType | typedef struct TciModuleParameterListType { long int length; TciModuleParameterType *modParList; } TciModuleParameterListType; | |
| TciParameterType | typedef struct TciParameterType { String parName; TciParameterPassingModeType parPassMode; TciValue parValue; } TciParameterType; | |
| TciParameterPassingModeType | typedef enum { TCI_IN_PAR = 0, TCI_INOUT_PAR = 1, TCI_OUT_PAR = 2 } TciParameterPassingModeType; | |
| TciParameterListType | typedef struct TciParameterListType { long int length; TciParameterType *parList; } TciParameterListType; | длина 0 интерпретируется как "пустой список". |

| ADT в IDL для TCI | Представление в ANSI-C (Определение типа) | Примечания и комментарии |
|--------------------------|---|---|
| TciModuleIdType | QualifiedName | |
| TciParameterTypeListType | <pre>typedef struct TciParameterTypeListType { long int length; TciType *parList; } TciParameterTypeListType;</pre> | длина 0 интерпретируется как "пустой список". |
| TciTestCaseIdListType | <pre>typedef struct TciTestCaseIdListType { long int length; QualifiedName *idList; } TciTestCaseIdListType;</pre> | длина 0 интерпретируется как "пустой список". |
| TciTypeClassType | <pre>typedef enum { TCI_ADDRESS_TYPE, TCI_ANYTYPE_TYPE, TCI_BITSTRING_TYPE, TCI_BOOLEAN_TYPE, TCI_CHAR_TYPE, TCI_CHARSTRING_TYPE, TCI_COMPONENT_TYPE, TCI_ENUMERATED_TYPE, TCI_FLOAT_TYPE, TCI_HEXSTRING_TYPE, TCI_INTEGER_TYPE, TCI_OBJID_TYPE, TCI_OCTETSTRING_TYPE, TCI_RECORD_TYPE, TCI_RECORD_OF_TYPE, TCI_SET_TYPE, TCI_SET_OF_TYPE, TCI_UNION_TYPE, TCI_UNIVERSAL_CHAR_TYPE, TCI_UNIVERSAL_CHARSTRING_TYPE, TCI_VERDICT_TYPE } TciTypeClassType;</pre> | |
| TciTestComponentKindType | <pre>typedef enum { TCI_CTRL_COMP, TCI_MTC_COMP, TCI_PTC_COMP, TCI_SYS_COMP } TciTestComponentKindType;</pre> | |
| TciBehaviourIdType | QualifiedName | |
| TciValueDifference | <pre>typedef struct TciValueDifference { TciValue val; TciValueTemplate tmpl; String desc; } TciValueDifference;</pre> | |
| TciValueDifferenceList | <pre>typedef struct TciValueDifferenceList { long int length; TciValueDifference[] diffList; } TciValueDifferenceList;</pre> | длина 0 интерпретируется как "пустой список". |

9.6 Разное

| Понятие в TCI | Представление в ANSI-C | Примечания и комментарии |
|---|---|--|
| Представление Verdict | | |
| NONE | <code>const int TCI_VERDICT_NONE = 0</code> | Поскольку интерфейс <code>TciVerdictValue</code> определяется исходя из целых чисел, то должно быть согласовано, какой вердикт определяется каким значением. |
| PASS | <code>const int TCI_VERDICT_PASS = 1</code> | |
| INCONC | <code>const int TCI_VERDICT_INCONC = 2</code> | |
| FAIL | <code>const int TCI_VERDICT_FAIL = 3</code> | |
| ERROR | <code>const int TCI_VERDICT_ERROR = 4</code> | |
| Представление Objid | | |
| Objid | <pre>typedef struct TciObjidValue { long int length; TciObjidElem *elements; } TciObjidValue;</pre> | Поскольку значение <code>Objid</code> возвращается как "as is" (исходное) через интерфейс значения <code>Objid</code> , то должно быть определено представление. |
| TciObjidElem | <pre>typedef struct TciObjidElemValue { char* elem_as_ascii; long int elem_as_number; void* aux; } TciObjidElemValue;</pre> | |
| Представление CharstringValue | | |
| TciCharString | <pre>typedef struct TciCharStringValue { unsigned long int length; char* string; } TciCharStringValue</pre> | |
| Представление универсальной символьной строки | | |
| Universal Char | <code>typedef unsigned char[4] TciUCValue</code> | |
| Universal Charstring | <pre>typedef struct TciUCStringValue { unsigned long int length; TciUCType *string; } TciUCStringValue;</pre> | |

10 Отображение языка XML консорциума W3C

10.1 Введение

В данном разделе представлено отображение языка XML интерфейса TCI [5], [6] и [7] для интерфейса регистрации в TCI. Отображение языка XML для интерфейса регистрации указывает, как определения языка IDL, описанные в разделе 7, отображаются на язык XML. Определения схемы для этого отображения представлены в Приложении В.

10.2 Контексты

Модуль `tciInterface` в IDL отображается на схему XML с пространством имен `http://uri.etsi.org/ttcn-3/3.0.0/tci/TLI`.

В этой схеме используются дополнительные схемы:

- [http://uri.etsi.org/ttcn-3/3.0.0/tci/Простые типы](http://uri.etsi.org/ttcn-3/3.0.0/tci/Простые_типы) для отображения простых типов на XML.
- <http://uri.etsi.org/ttcn-3/3.0.0/tci/Типы> для отображения структурных типов на XML.
- <http://uri.etsi.org/ttcn-3/3.0.0/tci/Значения> для отображения значений на XML.
- <http://uri.etsi.org/ttcn-3/3.0.0/tci/Шаблоны> для отображения шаблонов на XML.
- <http://uri.etsi.org/ttcn-3/3.0.0/tci/События> для отображения событий регистрации на XML.

10.3 Отображение типа

10.3.1 Отображение простых типов

10.3.1.1 TBoolean

Тип `TBoolean` в IDL отображается на базовый тип `boolean` в xsd.

10.3.1.2 Tstring

Тип `TString` в IDL отображается на базовый тип `string` в xsd.

10.3.1.3 TInteger

Тип `TInteger` в IDL отображается на базовый тип `integer` в xsd.

10.3.1.4 TriTimerDurationType

Тип `TriTimerDurationType` в IDL отображается на базовый тип `float` в xsd.

10.3.1.5 TciParameterPassingModeType

Тип `TciParameterPassingModeType` в IDL отображается на базовый тип `string` в xsd с перечисляемыми значениями `'in'`, `'out'` и `'inout'`.

10.3.1.6 TriStatusType

Тип `TriStatusType` в IDL отображается на базовый тип `string` в xsd с перечисляемыми значениями `'TRI_Ok'` и `'TRI_Error'`.

10.3.1.7 TciStatusType

Тип `TciStatusType` в IDL отображается на базовый тип `string` с перечисляемыми значениями `'TCI_Ok'` и `'TCI_Error'`.

10.3.2 Отображение комплексных типов

10.3.2.1 TriPortIdType

`TriPortIdType` отображается на следующий комплексный тип:

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

Элементы:

- `comp` Идентификатор компонента TRI;
- `port` Идентификация порта.

Атрибуты:

- отсутствуют.

10.3.2.2 TriComponentIdType

`TriComponentIdType` отображается на следующий комплексный тип:

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

Элементы:

- `id` Идентификатор компонента TRI;
- `null` Идентификатор `null`. Используется, если нет идентификатора компонента TRI.

Атрибуты:

- отсутствуют.

10.3.2.3 TriComponentIdListType

TriComponentIdListType отображается на следующий комплексный тип:

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

Элементы:

- `comp` Идентификаторы компонентов TRI в этом списке.

Атрибуты:

- отсутствуют.

10.3.2.4 Port

Port отображается на следующий комплексный тип:

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

Элементы:

- `id` Идентификатор порта;
- `port` Индекс порта.

Атрибуты:

- отсутствуют.

10.3.2.5 Id

Id используется как идентификация для компонентов, портов и таймеров и отображается на следующий комплексный тип:

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

Элементы:

- `name` Имя компонента, порта или таймера;
- `id` Внутреннее представление компонента, порта или таймера;
- `type` Тип компонента, порта или таймера.

Атрибуты:

- отсутствуют.

10.3.2.6 TriMessageType

TriMessageType отображается на следующий комплексный тип:

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

Элементы:

- val Закодированное сообщение.

Атрибуты:

- отсутствуют.

10.3.2.7 TriParameterType

TriParameterType отображается на следующий комплексный тип:

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

Элементы:

- val Закодированный параметр.

Атрибуты:

- name Имя параметра;
- mode Режим передачи параметра.

10.3.2.8 TriParameterListType

TriParameterListType отображается на следующий комплексный тип:

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

Элементы:

- par Параметры в этом списке.

Атрибуты:

- отсутствуют.

10.3.2.9 TriAddressType

TriAddressType отображается на следующий комплексный тип:

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

Элементы:

- val Значение адреса.

Атрибуты:

- отсутствуют.

10.3.2.10 TriAddressListType

TriAddressListType отображается на следующий комплексный тип:

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

Элементы:

- addr Адреса в этом списке.

Атрибуты:

- отсутствуют.

10.3.2.11 TriExceptionType

TriExceptionType отображается на следующий комплексный тип:

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

Элементы:

- val Особое состояние.

Атрибуты:

- отсутствуют.

10.3.2.12 TriSignatureIdType

TriSignatureIdType отображается на следующий комплексный тип:

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

Элементы:

- val Сигнатура.

Атрибуты:

- отсутствуют.

10.3.2.13 TriAddressType

TriAddressType отображается на следующий комплексный тип:

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

Элементы:

- val Адрес в системе SUT.

Атрибуты:

- отсутствуют.

10.3.2.14 TriTimerIdType

TriTimerIdType отображается на следующий комплексный тип:

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

Элементы:

- id Идентификация таймера.

Атрибуты:

- отсутствуют.

10.3.2.15 TriTimerDurationType:

TriTimerDurationType отображается на следующий комплексный тип:

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

Элементы:

- val Период времени таймера.

Атрибуты:

- отсутствуют.

10.3.2.16 QualifiedName

QualifiedName используется для полной классификации параметров модуля, переменных и т. д. и отображается на следующий комплексный тип:

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

Элементы:

- moduleName Имя модуля для модуля TTCN-3.
- baseName Имя полностью классифицируемого объекта.

Атрибуты:

- отсутствуют.

10.3.2.17 TciBehaviourIdType

TciBehaviourIdType отображается на следующий комплексный тип:

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

Элементы:

- name Классифицируемое имя поведения.

Атрибуты:

- отсутствуют.

10.3.2.18 TciTestCaseIdType

TciTestCaseIdType отображается на следующий комплексный тип:

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

Элементы:

- name Классифицируемое имя тестового примера.

Атрибуты:

- отсутствуют.

10.3.2.19 TciParameterType

TciParameterType отображается на следующий комплексный тип:

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

Элементы:

- val Закодированный параметр.

Атрибуты:

- name Имя параметра.
- mode Режим передачи параметра.

10.3.2.20 TciParameterListType

TciParameterListType отображается на следующий комплексный тип:

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

Последовательность элементов:

- par Параметры в этом списке.

Атрибуты:

- отсутствуют.

10.3.3 Отображение абстрактных значений

10.3.3.1 Value

Value отображается на следующий комплексный тип:

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

```

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

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

```

Выбор элементов:

- integer Целочисленное значение.
- float Значение с плавающей запятой.
- boolean Булево значение.
- objid Значение "идентификатор объекта".
- verdicttype Значение "тип вердикта".
- bitstring Значение "битовая строка".
- hexstring Значение "шестнадцатеричная строка".
- octetstring Значение "строка октетов".
- charstring Значение "строка символов".
- universal_charstring Значение "строка универсальных символов".
- record Значение "запись".
- record_of Значение "запись из".
- set Значение "множество".
- set_of Значение "множество из".
- enumerated Перечисляемое значение.
- union Значение "объединение".
- anytype Значение любого типа.
- address Значение "адрес".

Атрибуты:

- name Имя значения, если известно.
- type Тип значения, если известен.
- module Значение модуля, если известно.

10.3.3.2 IntegerValue

IntegerValue отображается на следующий комплексный тип:

```

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

```

Простое содержимое:

- base Целочисленное значение в виде строки.
- extension Те же атрибуты, что и у значения Value.

10.3.3.3 FloatValue

FloatValue отображается на следующий комплексный тип:

```

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

```


10.3.3.8 HexstringValue

HexstringValue отображается на следующий комплексный тип:

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

Простое содержимое:

- base Значение "шестнадцатеричная строка" в виде строки.
- extension Те же атрибуты, что и у значения Value.

10.3.3.9 OctetstringValue

OctetstringValue отображается на следующий комплексный тип:

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

Простое содержимое:

- base Значение "строка октетов" в виде строки.
- extension Те же атрибуты, что и у значения Value.

10.3.3.10 CharstringValue

CharstringValue отображается на следующий комплексный тип:

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

Простое содержимое:

- base Значение "строка символов" в виде строки.
- extension Те же атрибуты, что и у значения Value.

10.3.3.11 UniversalCharstringValue

UniversalCharstringValue отображается на следующий комплексный тип:

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

Простое содержимое:

- base Значение "строка универсальных символов" в виде строки.
- extension Те же атрибуты, что и у значения Value.

10.3.3.12 RecordValue

RecordValue отображается на следующий комплексный тип:

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

Последовательность элементов:

- integer Целочисленное значение.
- float Значение с плавающей запятой.
- boolean Булево значение.
- objid Значение "идентификатор объекта".
- verdicttype Значение "тип вердикта".
- bitstring Значение "битовая строка".
- hexstring Значение "шестнадцатеричная строка".
- octetstring Значение "строка октетов".
- charstring Значение "строка символов".
- universal_charstring Значение "строка универсальных символов".
- record Значение "Запись".
- record_of Значение "запись из".
- set Значение "множество".
- set_of Значение "множество из".
- enumerated Перечисляемое значение.
- union Значение "объединение".
- anytype Значение любого типа.
- address Значение "адрес".

Атрибуты:

- Те же атрибуты, что и у значения Value.

10.3.3.13 RecordOfValue

RecordOfValue отображается на следующий комплексный тип:

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

Выбор последовательности элементов:

- integer Целочисленное значение.
- float Значение с плавающей запятой.
- boolean Булево значение.
- objid Значение "идентификатор объекта".
- verdicttype Значение "тип вердикта".
- bitstring Значение "битовая строка".
- hexstring Значение "шестнадцатеричная строка".
- octetstring Значение "строка октетов".
- charstring Значение "строка символов".
- universal_charstring Значение "строка универсальных символов".
- record Значение "запись".
- record_of Значение "запись из".
- set Значение "множество".
- set_of Значение "множество из".
- enumerated Перечисляемое значение.
- union Значение "объединение".
- anytype Значение любого типа.
- address Значение "адрес".

Атрибуты:

- Те же атрибуты, что и у значения Value.

10.3.3.14 SetValue

SetValue отображается на следующий комплексный тип:

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

Последовательность элементов:

- integer Целочисленное значение.
- float Значение с плавающей запятой.
- boolean Булево значение.
- objid Значение "идентификатор объекта".
- verdicttype Значение "тип вердикта".


```

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

```

Выбор последовательности элементов:

- | | |
|------------------------|---|
| • integer | Целочисленное значение. |
| • float | Значение с плавающей запятой. |
| • boolean | Булево значение. |
| • objid | Значение "идентификатор объекта". |
| • verdicttype | Значение "тип вердикта". |
| • bitstring | Значение "битовая строка". |
| • hexstring | Значение "шестнадцатеричная строка". |
| • octetstring | Значение "строка октетов". |
| • charstring | Значение "строка символов". |
| • universal_charstring | Значение "строка универсальных символов". |
| • record | Значение "запись". |
| • record_of | Значение "запись из". |
| • set | Значение "множество". |
| • set_of | Значение "множество из". |
| • enumerated | Перечисляемое значение. |
| • union | Значение "объединение". |
| • anytype | Значение любого типа. |
| • address | Значение "адрес". |

Атрибуты:

- Атрибуты те же, что и у значения Value.

10.3.3.16 EnumeratedValue

EnumeratedValue отображается на следующий комплексный тип:

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

Последовательность элементов:

- element Перечисляемое значение.

Атрибуты:

- Те же атрибуты, что и у значения Value.

10.3.3.17 UnionValue

UnionValue отображается на следующий комплексный тип:

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

Выбор элементов:

- | | |
|------------------------|---|
| • integer | Целочисленное значение. |
| • float | Значение с плавающей запятой. |
| • boolean | Булево значение. |
| • objid | Значение "идентификатор объекта". |
| • verdicttype | Значение "тип вердикта". |
| • bitstring | Значение "битовая строка". |
| • hexstring | Значение "шестнадцатеричная строка". |
| • octetstring | Значение "строка октетов". |
| • charstring | Значение "строка символов". |
| • universal_charstring | Значение "строка универсальных символов". |
| • record | Значение "запись". |
| • record_of | Значение "запись из". |
| • set | Значение "множество". |
| • set_of | Значение "множество из". |
| • enumerated | Перечисляемое значение. |
| • union | Значение "объединение". |
| • anytype | Значение любого типа. |
| • address | Значение "адрес". |

Атрибуты:

- Те же атрибуты, что и у значения Value.

10.3.3.18 AnytypeValue

AnytypeValue отображается на следующий комплексный тип:

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

Выбор элементов:

- | | |
|------------------------|---|
| • integer | Целочисленное значение. |
| • float | Значение с плавающей запятой. |
| • boolean | Булево значение. |
| • objid | Значение "идентификатор объекта". |
| • verdicttype | Значение "тип вердикта". |
| • bitstring | Значение "битовая строка". |
| • hexstring | Значение "шестнадцатеричная строка". |
| • octetstring | Значение "строка октетов". |
| • charstring | Значение "строка символов". |
| • universal_charstring | Значение "строка универсальных символов". |
| • record | Значение "запись". |
| • record_of | Значение "запись из". |
| • set | Значение "множество". |
| • set_of | Значение "множество из". |
| • enumerated | Перечисляемое значение. |
| • union | Значение "объединение". |
| • address | Значение "адрес". |

Атрибуты:

- Те же атрибуты, что и у значения Value.

10.3.3.19 AddressValue

AddressValue отображается на следующий комплексный тип:

```
<xsd:complexType name="AddressValue">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="integer" type="Values:IntegerValue"/>
    <xsd:element name="float" type="Values:FloatValue"/>
    <xsd:element name="boolean" type="Values:BooleanValue"/>
    <xsd:element name="objid" type="Values:ObjidValue"/>
    <xsd:element name="verdicttype" type="Values:VerdictValue"/>
    <xsd:element name="bitstring" type="Values:BitstringValue"/>
    <xsd:element name="hexstring" type="Values:HexstringValue"/>
    <xsd:element name="octetstring" type="Values:OctetstringValue"/>
    <xsd:element name="charstring" type="Values:OctetstringValue"/>
  </xsd:choice>
  <xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>
```

```

<xsd:element name="universal_charstring"
  type="Values:UniversalCharstringValue"/>
<xsd:element name="record" type="Values:RecordValue"/>
<xsd:element name="record_of" type="Values:RecordOfValue"/>
<xsd:element name="set" type="Values:SetValue"/>
<xsd:element name="set_of" type="Values:SetOfValue"/>
<xsd:element name="enumerated" type="Values:EnumeratedValue"/>
<xsd:element name="union" type="Values:UnionValue"/>
<xsd:element name="anytype" type="Values:AnytypeValue"/>
</xsd:choice>
<xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>

```

Выбор элементов:

- | | |
|------------------------|---|
| • integer | Целочисленное значение. |
| • float | Значение с плавающей запятой. |
| • boolean | Булево значение. |
| • objid | Значение "идентификатор объекта". |
| • verdicttype | Значение "тип вердикта". |
| • bitstring | Значение "битовая строка". |
| • hexstring | Значение "шестнадцатеричная строка". |
| • octetstring | Значение "строка октетов". |
| • charstring | Значение "строка символов". |
| • universal_charstring | Значение "строка универсальных символов". |
| • record | Значение "запись". |
| • record_of | Значение "запись из". |
| • set | Значение "множество". |
| • set_of | Значение "множество из". |
| • enumerated | Перечисляемое значение. |
| • union | Значение "объединение". |
| • anytype | Значение любого типа. |

Атрибуты:

- Те же атрибуты, что и у значения Value.

10.3.4 Отображение абстрактных типов регистрации

Для облегчения регистрации совпадений между значениями и шаблонами определяются дополнительные типы.

10.3.4.1 TciValueTemplate

TciValueTemplate отображается на следующий комплексный тип:

```

<xsd:complexType name="TciValueTemplate">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Values:Value">
      <xsd:choice>
        <xsd:element name="integer" type="Templates:IntegerTemplate"/>
        <xsd:element name="float" type="Templates:FloatTemplate"/>
        <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
        <xsd:element name="objid" type="Templates:ObjidTemplate"/>
        <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
        <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
        <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
        <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
        <xsd:element name="universal_charstring"
          type="Templates:UniversalCharstringValue"/>
        <xsd:element name="record" type="Templates:RecordTemplate"/>
        <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
        <xsd:element name="set" type="Templates:SetTemplate"/>
        <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
        <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

<xsd:element name="union" type="Templates:UnionTemplate"/>
<xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
<xsd:element name="address" type="Templates:AddressTemplate"/>
<xsd:element name="omit" type="Templates:omit"/>
<xsd:element name="any" type="Templates:any"/>
<xsd:element name="anyoromit" type="Templates:anyoromit"/>
<xsd:element name="templateDef" type="SimpleTypes:TString"/>
</xsd:choice>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

Выбор элементов:

- | | |
|------------------------|---|
| • integer | Шаблон целого числа. |
| • float | Шаблон значения с плавающей запятой. |
| • boolean | Шаблон булева значения. |
| • objid | Шаблон "идентификатор объекта". |
| • verdicttype | Шаблон "тип вердикта". |
| • bitstring | Шаблон "битовая строка". |
| • hexstring | Шаблон "шестнадцатеричная строка". |
| • octetstring | Шаблон "строка октетов". |
| • charstring | Шаблон "строка символов". |
| • universal_charstring | Шаблон "строка универсальных символов". |
| • record | Шаблон "запись". |
| • record_of | Шаблон "запись из". |
| • set | Шаблон "множество". |
| • set_of | Шаблон "множество из". |
| • enumerated | Перечисляемый шаблон. |
| • union | Шаблон "объединение". |
| • anytype | Шаблон любого типа. |
| • address | Шаблон "адрес". |
| • omit | Шаблон omit. |
| • any | Шаблон any. |
| • anyoromit | Шаблон anyoromit. |
| • templateDef | Определение комплексного шаблона. |

Атрибуты:

- отсутствуют.

10.3.4.2 TciNonValueTemplate

TciNonValueTemplate отображается на следующий комплексный тип:

```

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

```

Выбор элементов:

- | | |
|---------------|-----------------------------------|
| • any | Шаблон any. |
| • all | Шаблон all. |
| • templateDef | Определение комплексного шаблона. |

Атрибуты:

- отсутствуют.

10.3.4.3 TciValueList

TciValueList отображается на следующий комплексный тип:

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

Последовательность элементов:

- val Значения в списке значений.

Атрибуты:

- отсутствуют.

10.3.4.4 TciValueDifference

TciValueDifference отображается на следующий комплексный тип:

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

Последовательность элементов:

- desc Причина несовпадения.
- val Ссылка на несовпадающее значение.
- tmpl Ссылка на шаблон.

Атрибуты:

- отсутствуют.

10.3.4.5 TciValueDifferenceList

TciValueDifferenceList отображается на следующий комплексный тип:

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

Последовательность элементов:

- diff Различия значений/шаблонов в списке различий между значениями.

Атрибуты:

- отсутствуют.

10.4 Отображение операций в интерфейсе регистрации

Каждая операция, предоставляемая в интерфейсе регистрации, имеет соответствующее определение комплексного типа в XML. Эти определения комплексного типа являются расширениями операции Event (Событие).

10.4.1 Event

Event отображается на следующий комплексный тип:

```
<!-- common definition for all events -->
<xsd:complexType name="Event" mixed="true">
  <xsd:sequence>
    <xsd:element name="am" type="SimpleTypes:TString"/>
  </xsd:sequence>
  <xsd:attribute name="ts" type="xsd:time" use="required"/>
  <xsd:attribute name="src" type="SimpleTypes:TString" use="optional"/>
  <xsd:attribute name="line" type="SimpleTypes:TInteger" use="optional"/>

  <!-- general identifier structure for test components, ports and timer -->
  <xsd:attribute name="name" type="SimpleTypes:TString" use="required"/>
  <xsd:attribute name="id" type="SimpleTypes:TInteger" use="required"/>
  <xsd:attribute name="type" type="SimpleTypes:TString" use="required"/>
</xsd:complexType>
```

Элементы:

- am Сообщение, используемое для дополнительной информации в файле регистрации.

Атрибуты:

- ts Время генерирования события.
- src Исходный файл спецификации теста.
- line Номер строки, где выполняется запрос.
- name Имя компонента, генерирующего это событие.
- id Идентификатор компонента, генерирующего это событие.
- type Тип компонента, генерирующего это событие.

10.4.2 Отображение операций

Отображение операций приводится ниже.

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

| TCI-TL provided | |
|--|---|
| void tliTcStarted (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseldType tcId, in TriParameterListType pars, in TriTimerDurationType dur) | <pre> <xsd:complexType name="tliTcStarted"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="tcId" type="Types:TciTestCaseIdType"/> <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="dur" type="Types:TriTimerDurationType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliTcTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciTestCaseldType tcId, in TriParameterListType pars, in VerdictValue outcome) | <pre> <xsd:complexType name="tliTcTerminated"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="tcId" type="Types:TciTestCaseIdType"/> <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="outcome" type="Values:VerdictValue"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliCtrlStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | <pre> <xsd:complexType name="tliCtrlStart"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliCtrlStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | <pre> <xsd:complexType name="tliCtrlStop"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliCtrlTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | <pre> <xsd:complexType name="tliCtrlTerminated"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliMSend_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriAddressType address, in TriStatusType encoderFailure, in TriMessageType msg, in TriStatusType transmissionFailure) | <pre> <xsd:complexType name="tliMSend_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TciStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |

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

| TCI-TL provided | |
|--|---|
| <p>void tliMSend_c_MC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdListType toList, in TriStatusType transmissionFailure)</p> | <pre><xsd:complexType name="tliMSend_c_MC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliMDetected_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriMessageType msg, in TriAddressType address)</p> | <pre><xsd:complexType name="tliMDetected_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Types:TriMessageType"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliMDetected_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TriComponentIdType from)</p> | <pre><xsd:complexType name="tliMDetected_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Types:TriMessageType"/> <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliMMismatch_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTpl, in TciValueDifferenceList diffs, in TriAddressType address, in TciValueTemplate addressTpl)</p> | <pre><xsd:complexType name="tliMMismatch_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="msgTpl" type="Templates:TciValueTemplate"/> <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:element name="addressTpl" type="Templates:TciValueTemplate" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliMMismatch_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTpl, in TciValueDifferenceList diffs, in TriComponentIdType from, in TciNonValueTemplate fromTpl)</p> | <pre><xsd:complexType name="tliMMismatch_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value"/> <xsd:element name="msgTpl" type="Templates:TciValueTemplate"/> <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/> <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/> <xsd:element name="fromTpl" type="Templates:TciNonValueTemplate" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |

TCI-TL provided

| | |
|---|--|
| <p>void tliMReceive_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TriAddressType address, in TciValueTemplate addressTmpl)</p> | <pre><xsd:complexType name="tliMReceive_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value" minOccurs="0"/> <xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:element name="addressTmpl" type="Templates:TciValueTemplate" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliMReceive_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in Value msgValue, in TciValueTemplate msgTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)</p> | <pre><xsd:complexType name="tliMReceive_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="msgValue" type="Values:Value" minOccurs="0"/> <xsd:element name="msgTmpl" type="Templates:TciValueTemplate" minOccurs="0"/> <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/> <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliPrCall_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriAddressType address, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)</p> | <pre><xsd:complexType name="tliPrCall_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliPrCall_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in TriStatusType encoderFailure, in TriParameterListType pars, in TriStatusType transmissionFailure)</p> | <pre><xsd:complexType name="tliPrCall_m_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType" minOccurs="0"/> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |

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

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

TCI-TL provided

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

TCI-TL provided

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

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

TCI-TL provided

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

| TCI-TL provided | |
|--|---|
| <p>void tliPrGetReply_c (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value replValue, in TciValueTemplate replyTmpl, in TriComponentIdType from, in TciNonValueTemplate fromTmpl)</p> | <pre><xsd:complexType name="tliPrGetReply_c"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="replValue" type="Values:Value"/> <xsd:element name="replTmpl" type="Values:Value"/> <xsd:element name="from" type="Types:TriComponentIdType"/> <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliPrRaise_m (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriAddressType address, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure)</p> | <pre><xsd:complexType name="tliPrRaise_m"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="excValue" type="Values:Value"/> <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TciStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliPrRaise_m_BC (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue, in Value excValue, in TriStatusType encoderFailure, in TriExceptionType exc, in TriStatusType transmissionFailure)</p> | <pre><xsd:complexType name="tliPrRaise_m_BC"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TciParameterListType"/> <xsd:element name="excValue" type="Values:Value"/> <xsd:choice> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> <xsd:sequence> <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/> <xsd:element name="transmission-failure" type="SimpleTypes:TciStatusType" minOccurs="0"/> </xsd:sequence> </xsd:choice> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |

TCI-TL provided

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

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

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

| TCI-TL provided | |
|---|---|
| void tliPrCatchTimeout (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port, in TriSignatureIdType signature, in TciParameterListType parsValue) | <xsd:complexType name="tliPrCatchTimeout"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="port" type="Types:TriPortIdType"/> <xsd:element name="signature" type="Types:TriSignatureIdType"/> <xsd:element name="parsValue" type="Types:TriParameterListType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliCCreate (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TString name) | <xsd:complexType name="tliCCreate"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="name" type="SimpleTypes:TString"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliCStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TciBehaviourIdType name, in TciParameterListType parsValue) | <xsd:complexType name="tliCStart"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="name" type="Types:TciBehaviourIdType"/> <xsd:element name="pars" type="Types:TciParameterListType" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliCRunning (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status) | <xsd:complexType name="tliCRunning"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="status" type="SimpleTypes:TBoolean"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliCAlive, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status) | <xsd:complexType name="tliCAlive"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="status" type="SimpleTypes:TBoolean"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliCStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType comp) | <xsd:complexType name="tliCStop"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |

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

| TCI-TL provided | |
|--|---|
| void tliCTerminated (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict) | <xsd:complexType name="tliCTerminated"> without verdict) --> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="comp" type="Types:TriComponentIdType"/> <xsd:element name="verdict" type="Values:VerdictValue" maxOccurs="1"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliPConnect (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2) | <xsd:complexType name="tliPConnect"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType> |
| void tliPDisconnect (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2) | <xsd:complexType name="tliPDisconnect"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType> |
| void tliPMap (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2) | <xsd:complexType name="tliPMap"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType> |
| void tliPUnmap (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1, in TriComponentIdType c2, in TriPortIdType port2) | <xsd:complexType name="tliPUnmap"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType> |
| void tliPClear (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port) | <xsd:complexType name="tliPClear"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType> |
| void tliPStart(in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port) | <xsd:complexType name="tliPStart"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType> |
| void tliPStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port) | <xsd:complexType name="tliPStop"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType> |
| void tliPHalt (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriPortIdType port) | <xsd:complexType name="tliPHalt"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:PortConfiguration"/> </xsd:complexContent> </xsd:complexType> |

| TCI-TL provided | |
|--|---|
| <p>void tliEncode (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in Value val, in TriStatusType encoderFailure, in TriMessageType msg, in TString codec)</p> | <pre><xsd:complexType name="tliEncode"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="val" type="Values:Value"/> <xsd:choice> <xsd:element name="msg" type="Types:TriMessageType"/> <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/> </xsd:choice> </xsd:sequence> <xsd:attribute name="codec" type="SimpleTypes:TString" use="optional"/> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliDecode (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriMessageType msg, in TriStatusType decoderFailure, in Value val, in TString codec)</p> | <pre><xsd:complexType name="tliDecode" mixed="true"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:choice> <xsd:element name="val" type="Values:Value"/> <xsd:element name="decoder-failure" type="SimpleTypes:TciStatusType"/> </xsd:choice> <xsd:element name="msg" type="Types:TriMessageType"/> </xsd:sequence> <xsd:attribute name="codec" type="SimpleTypes:TString" use="optional"/> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliTTimeoutDetected (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer)</p> | <pre><xsd:complexType name="tliTTimeoutDetected"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1" minOccurs="1"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliTTimeoutMismatch (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTmpl)</p> | <pre><xsd:complexType name="tliTTimeoutMismatch"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1" minOccurs="1"/> <xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" maxOccurs="1" minOccurs="1"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |
| <p>void tliTTimeout (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciNonValueTemplate timerTmpl)</p> | <pre><xsd:complexType name="tliTTimeout"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1" minOccurs="1"/> <xsd:element name="timerTmpl" type="Templates:TciNonValueTemplate" maxOccurs="1" minOccurs="1"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType></pre> |

| TCI-TL provided | |
|---|---|
| void tliTStart (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType dur) | <pre> <xsd:complexType name="tliTStart"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType"/> <xsd:element name="dur" type="Types:TriTimerDurationType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliTStop (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer) | <pre> <xsd:complexType name="tliTStop"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliTRead (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TriTimerDurationType elapsed) | <pre> <xsd:complexType name="tliTRead"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType"/> <xsd:element name="elapsed" type="Types:TriTimerDurationType"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliTRunning (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TriTimerIdType timer, in TBoolean status) | <pre> <xsd:complexType name="tliTRunning"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="timer" type="Types:TriTimerIdType"/> </xsd:sequence> <xsd:attribute name="status" type="SimpleTypes:TBoolean"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliSEnter (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in TciParameterListType parsValue, in TString kind) | <pre> <xsd:complexType name="tliSEnter"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/> <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/> <xsd:element name="kind" type="SimpleTypes:TString"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |
| void tliSLeave (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value returnValue, in TString kind) | <pre> <xsd:complexType name="tliSLeave"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/> <xsd:element name="return" type="Values:Value" minOccurs="0"/> <xsd:element name="kind" type="SimpleTypes:TString"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre> |

| TCI-TL provided | |
|---|---|
| void tliVar (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value varValue) | <xsd:complexType name="tliVar"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/> <xsd:element name="val" type="Values:Value" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliModulePar (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TString name, in Value parValue) | <xsd:complexType name="tliModulePar"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="name" type="Types:QualifiedName" minOccurs="1" maxOccurs="1"/> <xsd:element name="val" type="Values:Value" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliGetVerdict (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict) | <xsd:complexType name="tliGetVerdict"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="verdict" type="Values:VerdictValue"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliSetVerdict (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in VerdictValue verdict) | <xsd:complexType name="tliSetVerdict"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="verdict" type="Values:VerdictValue"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliLog (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c, in TciValueList log) | <xsd:complexType name="tliLog"> <xsd:complexContent mixed="true"> <xsd:extension base="Events:Event"> <xsd:sequence> <xsd:element name="log" type=" Values:Value"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> |
| void tliAEnter (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | <xsd:complexType name="tliAEnter"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType> |
| void tliALeave (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | <xsd:complexType name="tliALeave"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType> |
| void tliADefaults (in TString am, in TInteger ts, in TString src, in TInteger line, in TriComponentIdType c) | <xsd:complexType name="tliADefaults"> <xsd:complexContent> <xsd:extension base="Events:Event"/> </xsd:complexContent> </xsd:complexType> |

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

11 Сценарии использования

В данном разделе представлены сценарии использования, которые должны помочь пользователям интерфейса TCI и поставщикам инструментальных средств, предоставляющим TCI, в понимании семантики определяемых в настоящей Рекомендации операций.

Указанные сценарии определяются при помощи диаграмм последовательности языка UML. На диаграмме последовательности показаны взаимодействия между объектами интерфейса TCI. Эти сценарии сопровождаются надлежащими объяснениями, а там, где применимо, – подкрепляются фрагментом TTCN-3, соответствующим конкретному сценарию.

11.1 Инициализация, сбор информации, регистрация

11.1.1 Сценарий использования: инициализация

В сценарии на Рисунке 9 показана фаза инициализации для тестовой системы, когда для выполнения должен быть выбран модуль TTCN-3. Сначала с помощью `tciRootModule` должен быть установлен корневой модуль. Параметры корневого модуля могут быть получены с помощью `tciGetModuleParameters`. Информация о параметрах модуля может применяться для запроса у пользователя тестовой системы конкретных значений для каждого параметра модуля. Список тестовых примеров, доступных в корневом модуле, может быть выведен с помощью `tciGetTestCases`. Выполнение этих тестовых примеров может осуществляться непосредственно в процессе администрирования теста. Их параметры и интерфейс их тестовой системы могут быть получены соответственно с помощью `tciGetTestCaseParameters` и `tciGetTestCaseTSI`.

11.1.1.1 Диаграмма последовательности

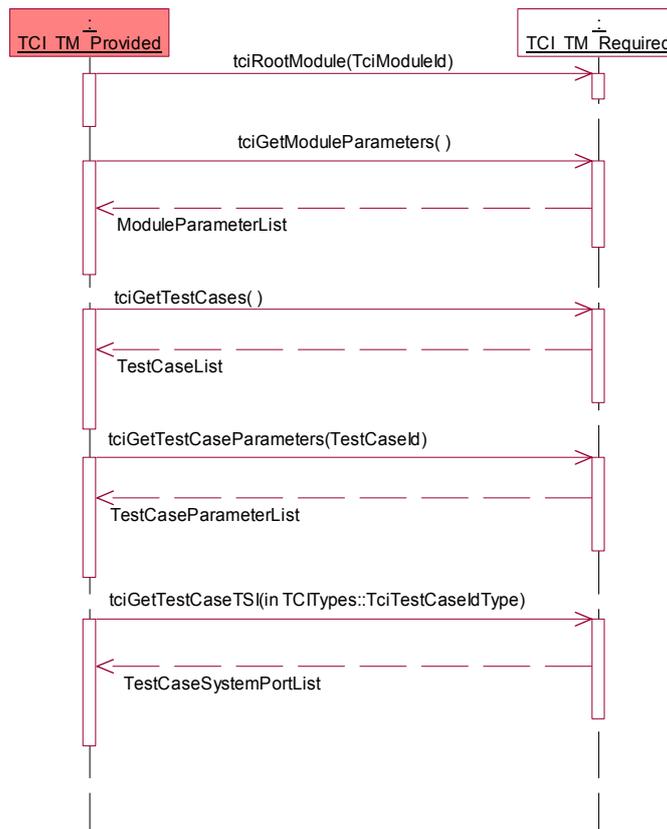


Рисунок 9/Z.145 – Сценарий использования – Инициализация

11.1.1.2 Фрагмент TTCN-3

Инициализация выходит за рамки TTCN-3.

11.1.2 Сценарий использования: запрос параметров модуля

Сценарий на Рисунке 10 показывает, как тестовый компонент запрашивает фактическое значение параметра модуля, необходимое для выполнения его тестового поведения. Сначала запрашивается тип параметра модуля, затем с помощью ТМ может быть сконструировано это значение и передано на ТЕ.

11.1.2.1 Диаграмма последовательности

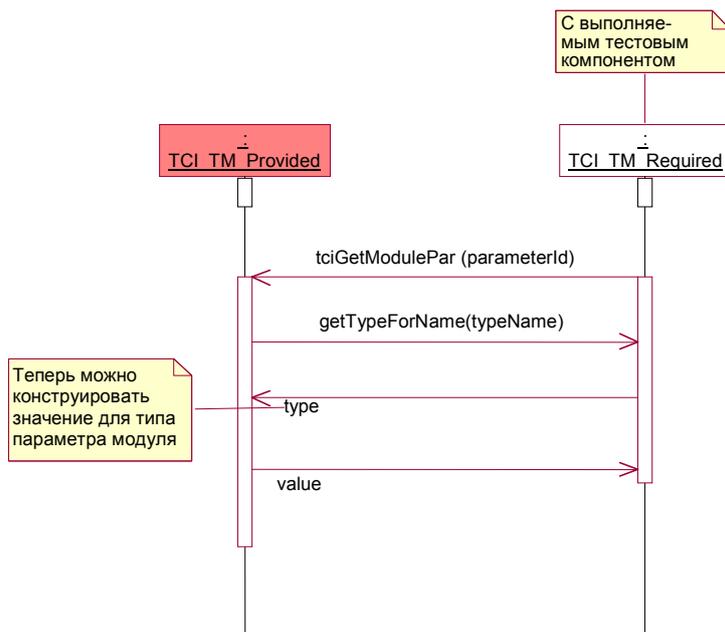


Рисунок 10/Z.145 – Сценарий использования – Запрос параметров модуля

11.1.2.2 Фрагмент TTCN-3

```
module AModule {
  ...
  modulepar {
    integer AModulePar
  }
  ...
  function AFunction (...) ... {
    integer x;
    ...
    x:= 2+AModulePar; // выражение с параметром модуля
    ...
  }
  ...
}
```

11.1.3 Сценарий использования: регистрация

В сценарии на Рисунке 11 показана регистрация информации во время выполнения тестового поведения тестовым компонентом. Сообщение, подлежащее регистрации, передается на тестовую регистрацию.

11.1.3.1 Диаграмма последовательности

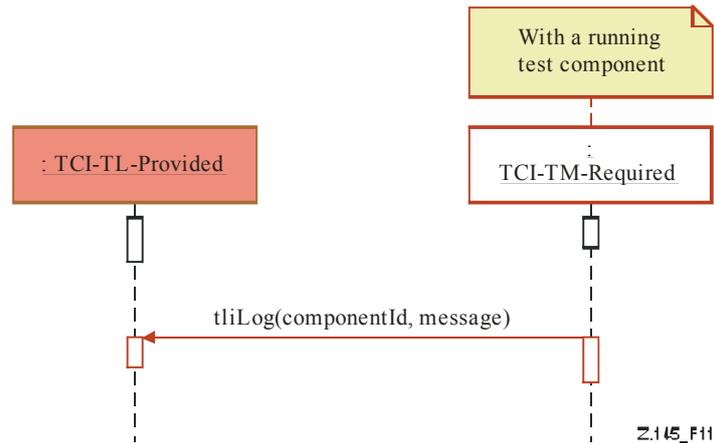


Рисунок 11/Z.145 – Сценарий использования – Регистрация

11.1.3.2 Фрагмент TTCN-3

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

11.2 Выполнение тестовых примеров и управления

11.2.1 Сценарий использования: выполнение управления

В сценарии на Рисунке 12 показана последовательность операций для выполнения управляющей части модуля TTCN-3. Сначала выбирается модуль, содержащий управляющую часть, затем запускается управление и выполняется до тех пор, пока ТЕ не завершит это выполнение.

11.2.1.1 Диаграмма последовательности

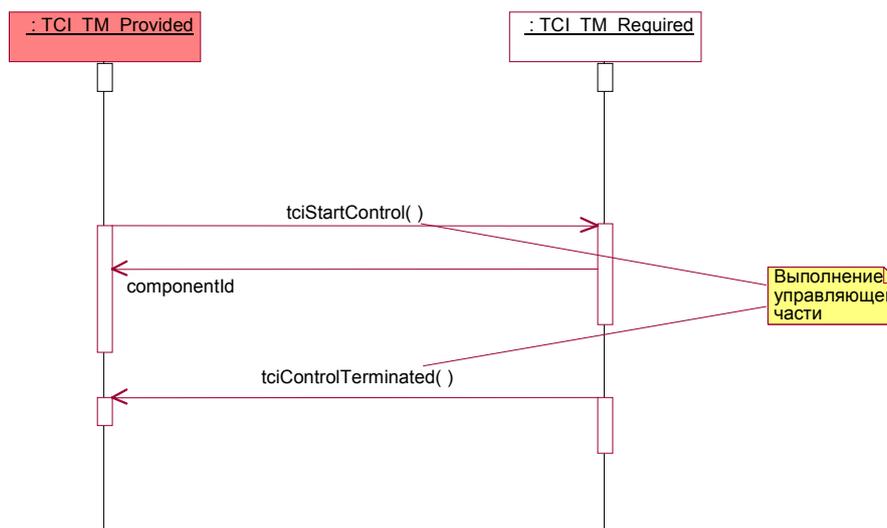


Рисунок 12/Z.145 – Сценарий использования – Выполнение управления

11.2.1.2 Фрагмент TTCN-3

```
module AModule {  
  ...  
  control {  
    ...  
  }  
  ...  
}
```

11.2.2 Сценарий использования: выполнение тестового примера в пределах управления

В сценарии на Рисунке 13 показано, как тестовый пример выполняется в пределах управляющей части.

11.2.2.1 Диаграмма последовательности

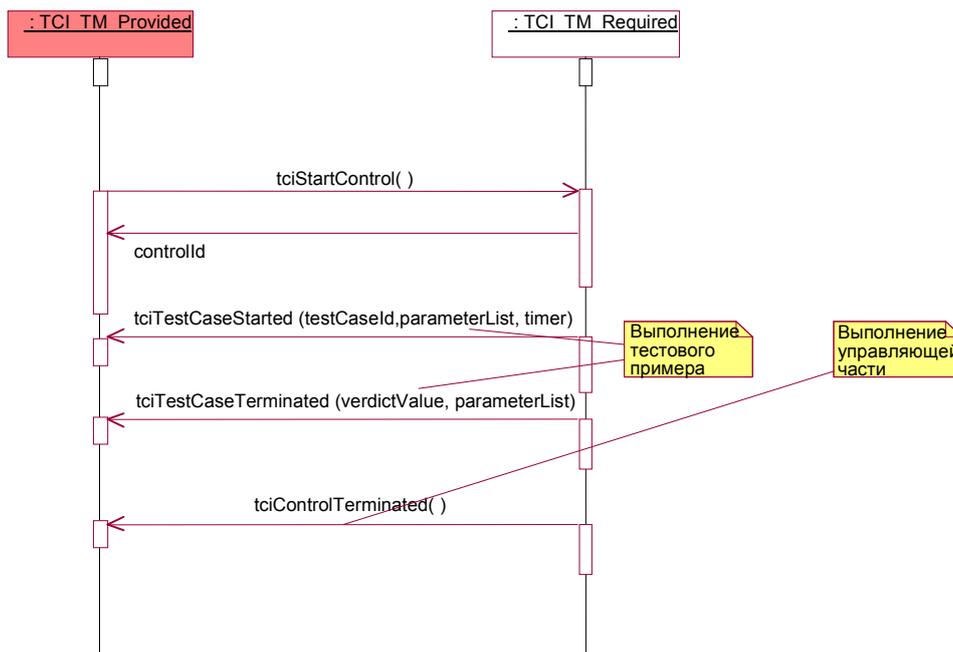


Рисунок 13/Z.145 – Сценарий использования – Выполнение тестового примера в пределах управления

11.2.2.2 Фрагмент TTCN-3

```
module AModule {  
  ...  
  testcase ATestCase(...)... {  
    ... // поведение тестового примера  
  }  
  ...  
  control {  
    ...  
    execute (ATestCase (...));  
    ...  
  }  
  ...  
}
```

11.2.3 Сценарий использования: непосредственное выполнение тестового примера

В сценарии на Рисунке 14 показано, как с помощью администрирования тестом можно непосредственно выполнить тестовый пример вне управляющей части. После выбора модуля TTCN-3, содержащего подлежащий выполнению тестовый пример, требуется запуск тестового примера. По завершении выполнения этого тестового примера ТЕ информирует администрирование тестом о завершении тестового примера.

11.2.3.1 Диаграмма последовательности

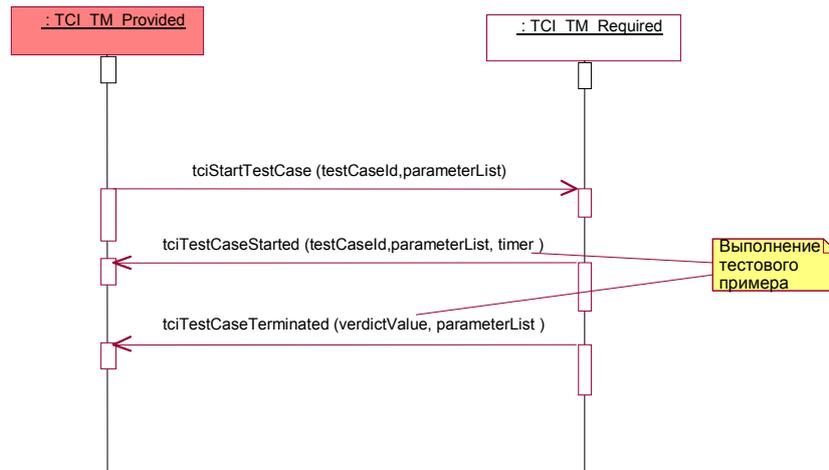


Рисунок 14/Z.145 – Сценарий использования – Непосредственное выполнение тестового примера

11.2.3.2 Фрагмент TTCN-3

Непосредственное выполнение тестового примера выходит за рамки TTCN-3.

11.2.4 Сценарий использования: информирование TRI о выполнении тестового примера

В сценарии на Рисунке 15 показано, как интерфейс TRI информируют о выполнении тестового примера с тем, чтобы он мог установить и инициализировать системные порты, когда это необходимо. Запрос на выполнение тестового примера должен быть сгенерирован до запуска тестового поведения в МТС текущего тестового примера.

11.2.4.1 Диаграмма последовательности

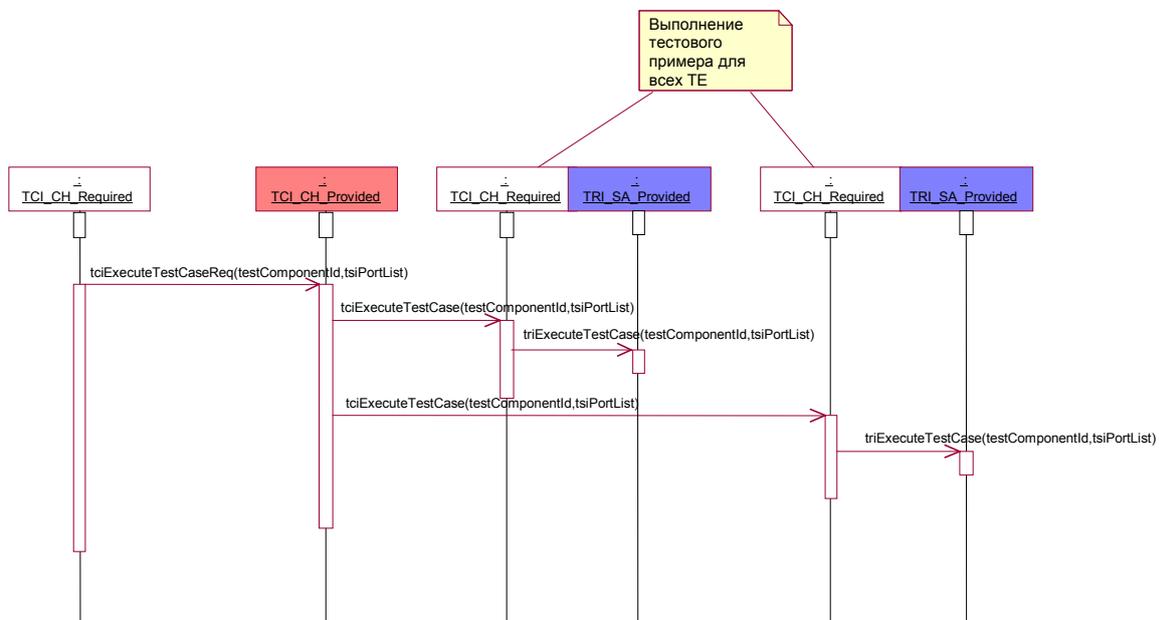


Рисунок 15/Z.145 – Сценарий использования – Информирование TRI о выполнении тестового примера

11.2.4.2 Фрагмент TTCN-3

```
module AModule {  
  ...  
  testcase ATestCase(...)... {  
    ... //поведение тестового примера  
  }  
  ...  
  control {  
    ...  
    execute (ATestCase (...));  
    ...  
  }  
  ...  
}
```

11.3 Обработка компонента

11.3.1 Сценарий использования: создание локального управляющего компонента

Сценарий на Рисунке 16 демонстрирует создание управляющего компонента в том же узле, где располагается интерфейс пользователя к интерфейсу TCI-TM администрирования тестом. Управляющий компонент создается всякий раз, когда выполняется управляющая часть модуля TTCN-3. Когда интерфейс TCI-TM администрирования тестом генерирует запуск управляющей части, запрос на создание тестового компонента посылается в интерфейс TCI-CH, который передает его на ТЕ, где должен быть создан управляющий компонент. В данном случае это – ТЕ в том же узле. Идентификатор для управляющего компонента возвращается и передается интерфейсу TCI-TM. Затем этот идентификатор используется для запуска поведения управляющей части в управляющем компоненте.

11.3.1.1 Диаграмма последовательности

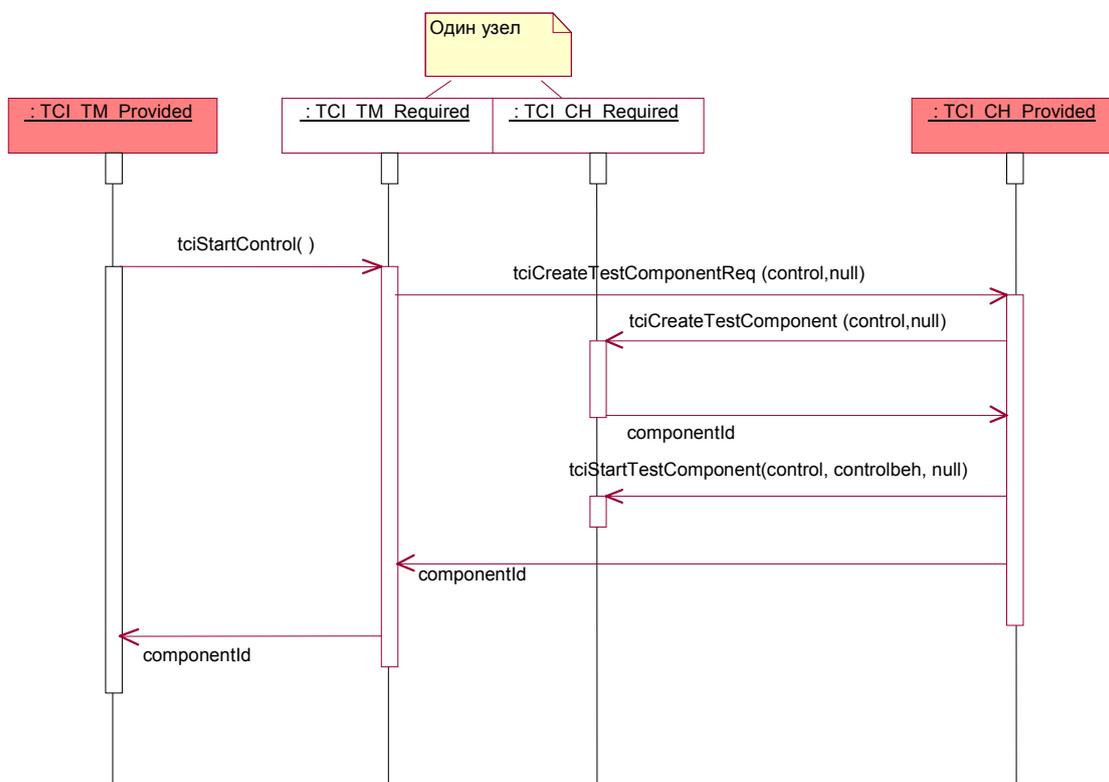


Рисунок 16/Z.145 – Сценарий использования – Создание локального управляющего компонента

11.3.1.2 Фрагмент TTCN-3

```
module AModule {  
    ...  
    control {  
        ...  
    }  
    ...  
}
```

11.3.2 Сценарий использования: создание удаленного управляющего компонента

В сценарии на Рисунке 17 демонстрируется создание управляющего компонента на другом узле, а не на том, где располагается интерфейс пользователя к интерфейсу TCI-TM администрирования тестом. Управляющий компонент создается всякий раз, когда выполняется управляющая часть модуля TTCN-3. Когда интерфейс TCI-TM администрирования тестом генерирует запуск управляющей части, запрос на создание тестового компонента посылается в интерфейс TCI-CH, который передает его на ТЕ, где должен быть создан управляющий компонент. В данном случае это – ТЕ на другом удаленном узле. Идентификатор для управляющего компонента возвращается и передается в интерфейс TCI-TM. Затем этот идентификатор используется для запуска поведения управляющей части в управляющем компоненте.

11.3.2.1 Диаграмма последовательности

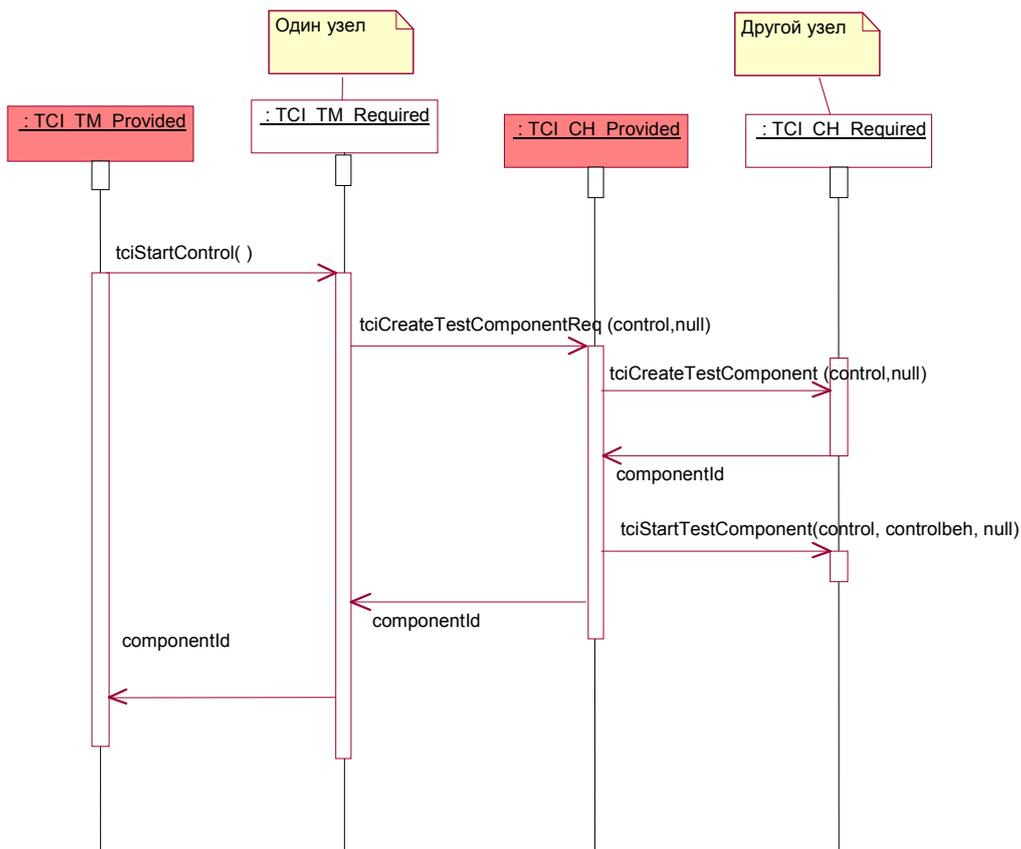


Рисунок 17/Z.145 – Сценарий использования – Создание удаленного управляющего компонента

11.3.2.2 Фрагмент TTCN-3

```
module AModule {  
  ...  
  control {  
    ...  
  }  
  ...  
}
```

11.3.3 Сценарий использования: локальное создание МТС

Сценарий на Рисунке 18 демонстрирует локальное создание главного тестового компонента. Локальное создание подразумевает два случая:

- 1) создание на том же узле, где располагается интерфейс пользователя к интерфейсу TCI-TM администрирования теста (когда тестовый пример запускается непосредственно); или
- 2) создание на том же узле, где располагается управляющий компонент (когда тестовый пример выполняется из управляющей части).

Главный тестовый компонент создается всякий раз, когда выполняется тестовый пример: запрос на создание тестового компонента посылается в интерфейс TCI-CH, который передает его на ТЕ, где должен быть создан главный тестовый компонент. В данном случае это – ТЕ в том же узле. Идентификатор для главного тестового компонента возвращается и передается интерфейсу TCI-TM. Этот идентификатор затем используется для запуска поведения тестового примера в главном тестовом компоненте (здесь это не показано, но обработка та же, что и в сценариях, описанных в пп. 11.3.5 и 11.3.6).

11.3.3.1 Диаграмма последовательности

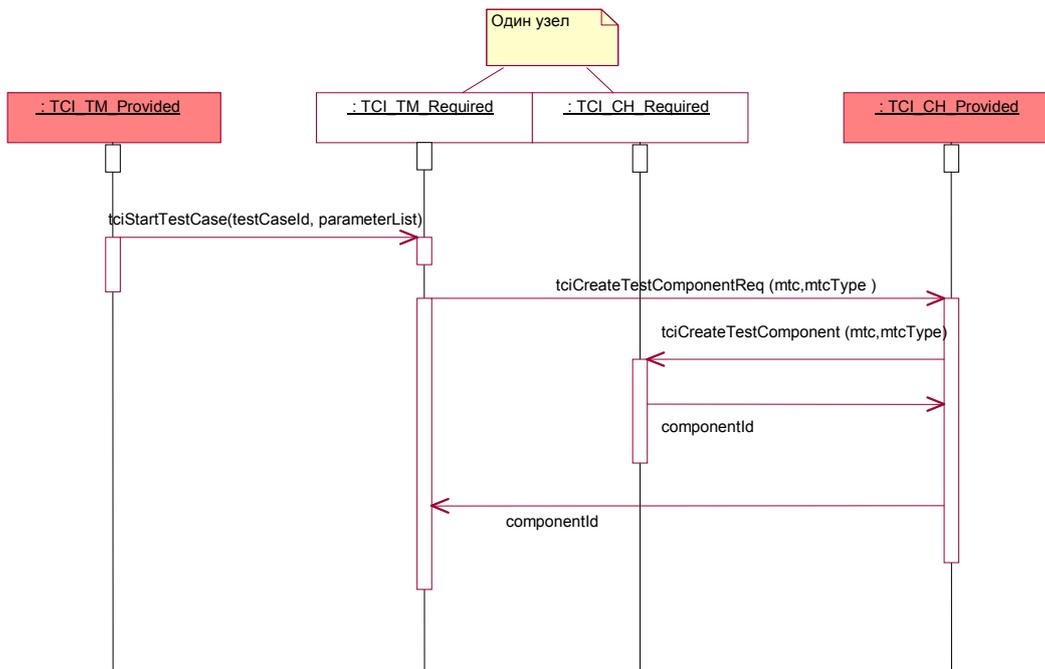


Рисунок 18/Z.145 – Сценарий использования – Локальное создание МТС

11.3.3.2 Фрагмент TTCN-3

```
module AModule {  
  ...  
  testcase ATestCase (...) runs on MTCType... {  
    ... //поведение тестового примера  
  }  
  ...  
}
```

11.3.4 Сценарий использования: удаленное создание МТС

Сценарий на Рисунке 19 демонстрирует удаленное создание главного тестового компонента. Удаленное создание подразумевает два случая:

- 1) создание на другом узле, а не на том, где располагается интерфейс пользователя к интерфейсу TCI-TM администрирования тестом (когда тестовый пример запускается непосредственно); или
- 2) создание на другом узле, а не на том, где располагается управляющий компонент (когда тестовый пример выполняется из управляющей части).

Главный тестовый компонент создается всякий раз, когда выполняется тестовый пример: запрос на создание тестового компонента посылается в интерфейс TCI-CH, который передает его на ТЕ, где должен быть создан главный тестовый компонент. В данном случае это – ТЕ в другом узле. Идентификатор для главного тестового компонента возвращается и передается интерфейсу TCI-TM. Этот идентификатор затем используется для запуска поведения тестового примера в главном тестовом компоненте (здесь это не показано, но обработка та же, что и в сценариях, описанных в пп. 11.3.5 и 11.3.6).

11.3.4.1 Диаграмма последовательности

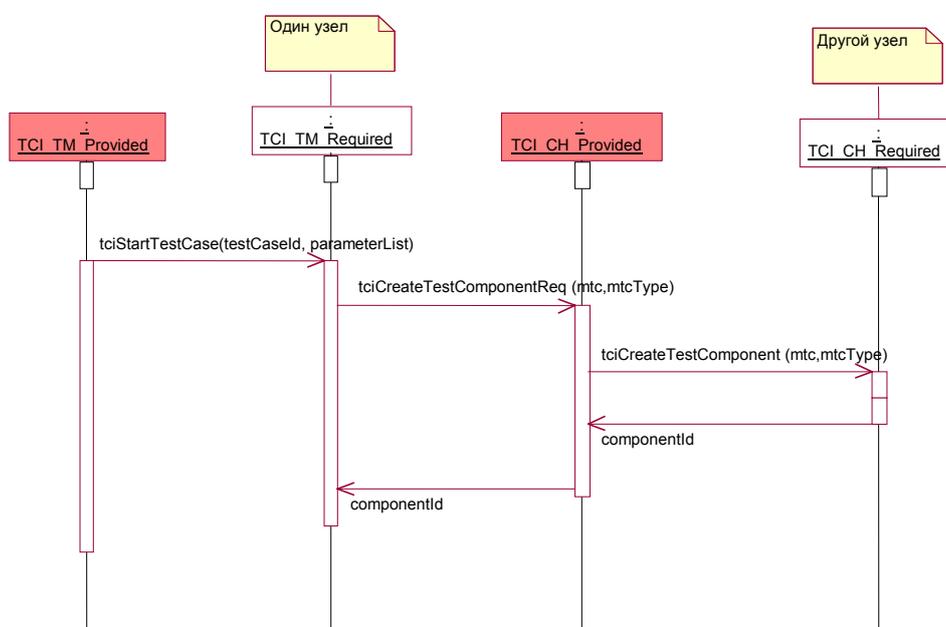


Рисунок 19/Z.145 – Сценарий использования – Удаленное создание МТС

11.3.4.2 Фрагмент TTCN-3

```
module AModule {  
  ...  
  testcase ATestCase(...) runs on MTCType ... {  
    ... //поведение тестового примера  
  }  
  ...  
}
```

11.3.5 Сценарий использования: обработка компонента для выполнения тестового примера в пределах управления

В сценарии на Рисунке 20 демонстрируется обработка компонентов для выполнения тестового примера в пределах управляющей части. При запуске управляющей части создается управляющий компонент, а его идентификатор компонента возвращается к администрированию тестом. Для каждого тестового примера, подлежащего выполнению в пределах управляющей части, создается главный тестовый компонент, а идентификатор компонента возвращается управляющему компоненту. После этого в главном тестовом компоненте запускается поведение тестового примера, а администрирование тестом

информируется о запуске тестового примера. Когда главный тестовый компонент завершается, то передается запрос на его завершение вместе с локальным вердиктом главного тестового компонента, чтобы позволить вывести глобальный тестовый вердикт и предоставить информацию о завершении тестового примера.

11.3.5.1 Диаграмма последовательности

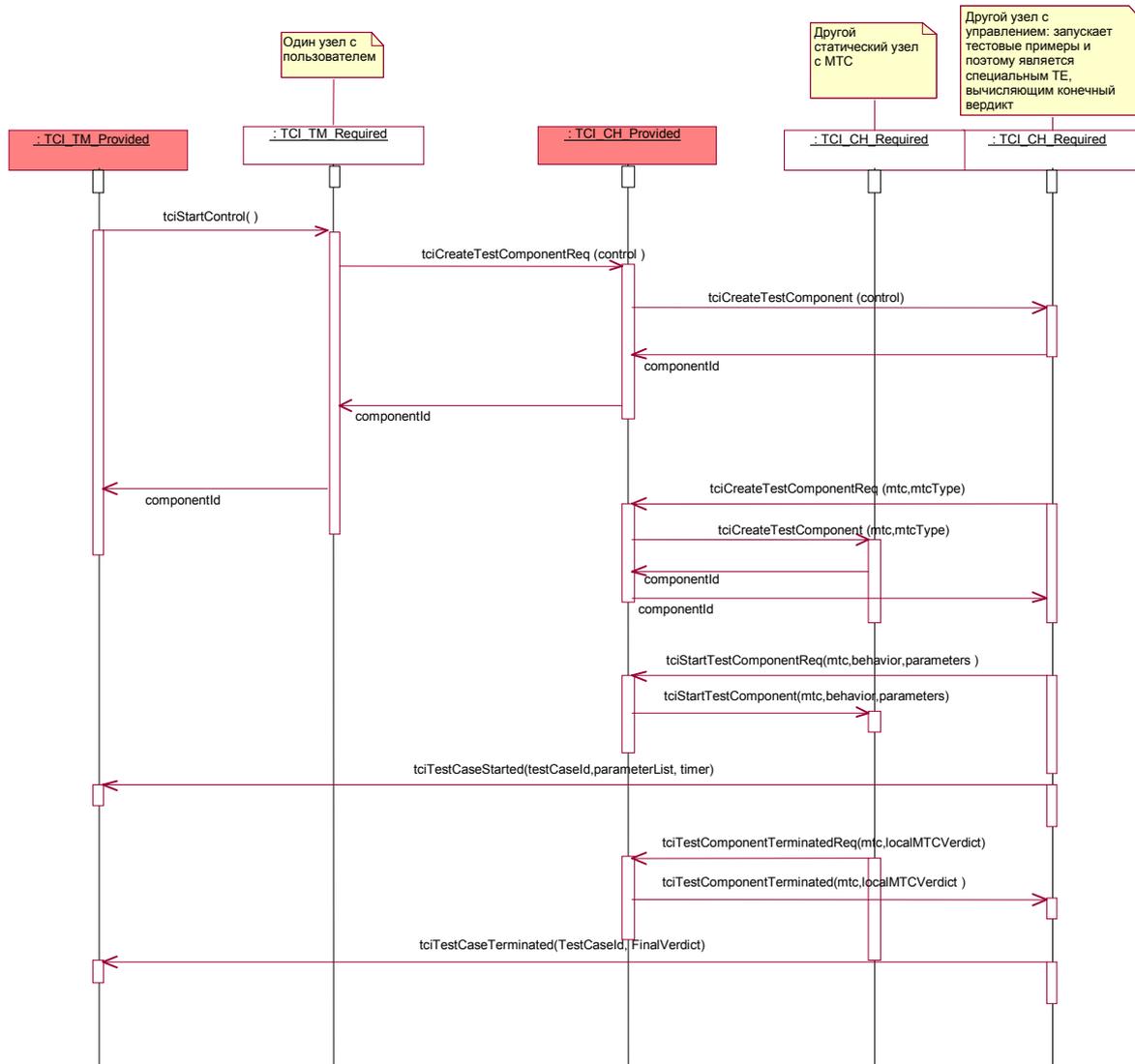


Рисунок 20/Z.145 – Сценарий использования – Обработка компонента для выполнения тестового примера в пределах управления

11.3.5.2 Фрагмент TTCN-3

```

module AModule {
  ...
  testcase ATestCase(...)... {
    ... //поведение тестового примера
  }
  ...
  control {
    ...
    execute(ATestCase(...));
    ...
  }
  ...
}
  
```

11.3.6 Сценарий использования: обработка компонента для непосредственного выполнения тестового примера

В сценарии на Рисунке 21 показано, как обрабатываются тестовые компоненты, когда тестовый пример выполняется непосредственно, то есть вне управляющей части. Когда запускается тестовый пример, создается главный тестовый компонент и сначала запускается поведение тестового примера в данном главном тестовом компоненте. Всякий раз, когда в тестовом примере используется параллельный тестовый компонент, он обрабатывается одним и тем же образом: сначала запускается параллельный тестовый компонент: при этом запрос на создание тестового компонента передается объекту TCI-CH, который передает этот запрос на ТЕ, в котором создается этот параллельный тестовый компонент. Возвращается идентификатор для созданного параллельного тестового компонента. Этот идентификатор затем используется для запуска поведения РТС при операции запуска. Когда РТС завершает свое выполнение, генерируется запрос на завершение тестового компонента вместе с локальным тестовым вердиктом для информирования объекта TCI-CH об этом завершении. То же самое делается, когда завершается главный тестовый компонент. Кроме того, завершение главного тестового компонента приводит к общему завершению тестового примера.

11.3.6.1 Диаграмма последовательности

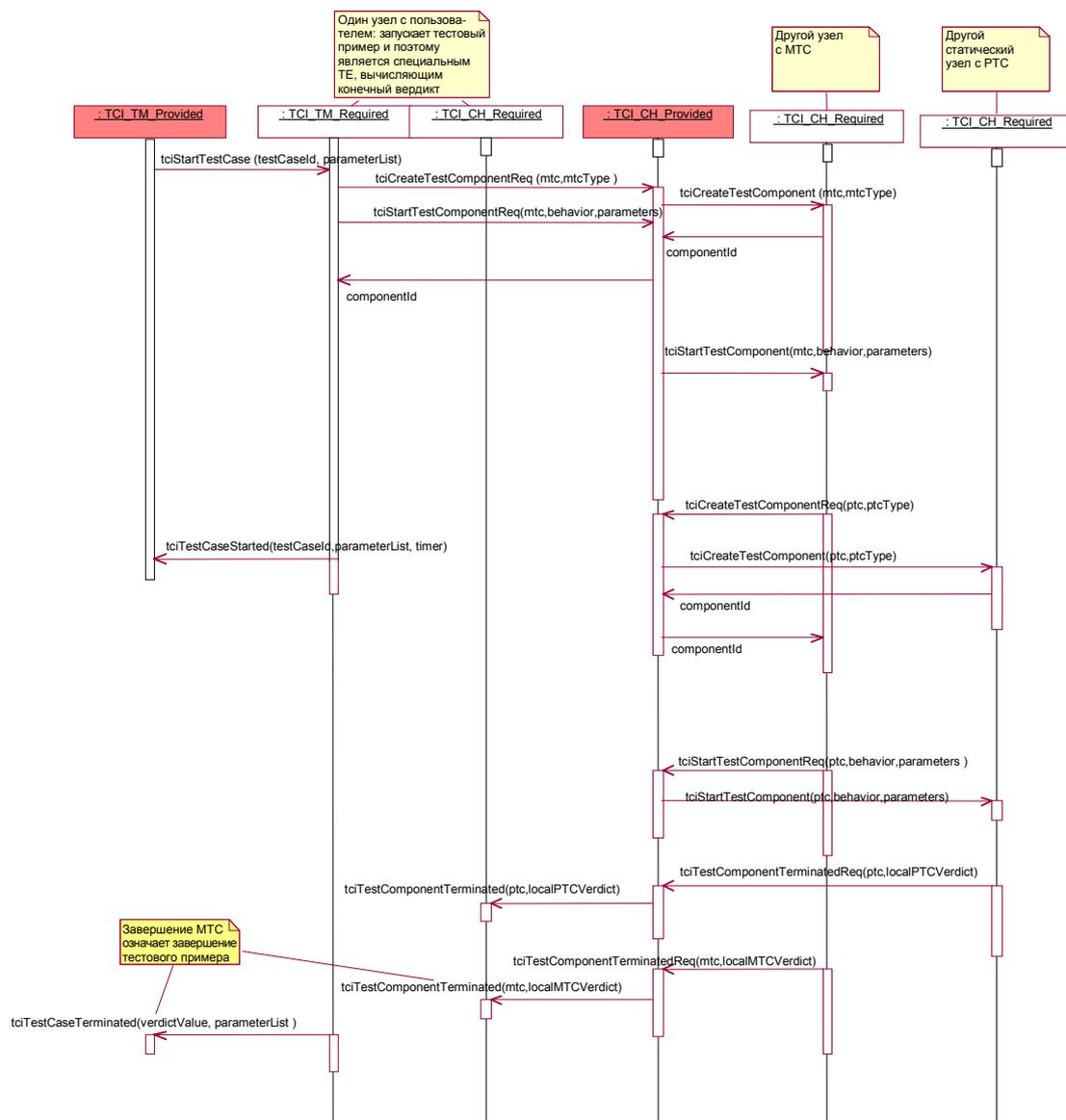


Рисунок 21/Z.145 – Сценарий использования – Обработка компонента для непосредственного выполнения тестового примера

11.3.6.2 Фрагмент TTCN-3

```

module AModule {
  ...
  function APTCBehaviour(...) runs on APTCType {
    ... //поведение PTC
  }
  ...
  testcase ATestCase(...)... {
    ... //поведение тестового примера
    var APTCType PTC:= APTCType.create;
    ...
    PTC.start(APTCBehaviour(...));
    ...
  }
  ...
}

```

11.3.7 Сценарий использования: передача запроса на отображение/соединение

В сценарии на Рисунке 22 показано, как отображаются порты. Запрос на отображение порта передается на ТЕ, где в итоге выполняется отображение. Передача запросов на соединение осуществляется аналогично.

11.3.7.1 Диаграмма последовательности

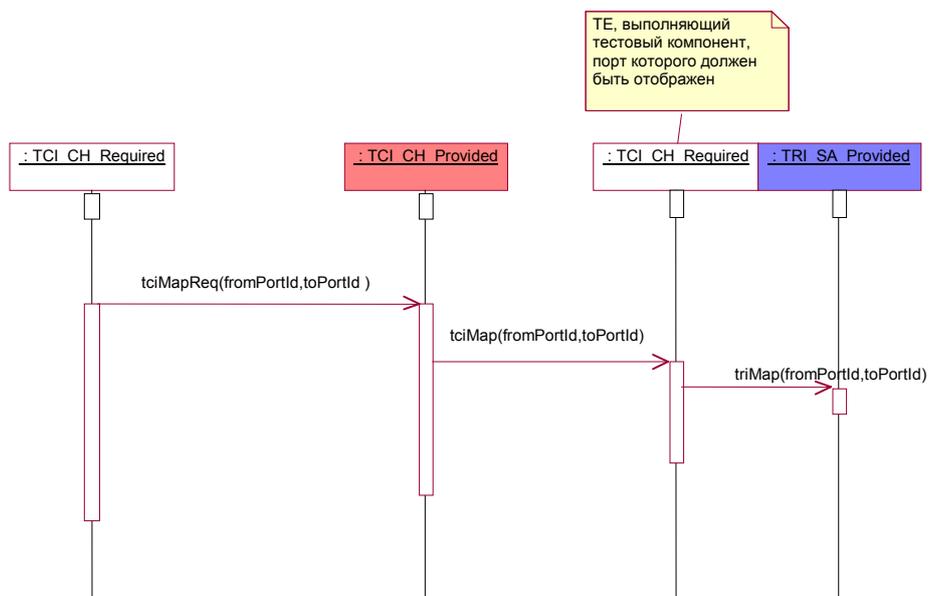


Рисунок 22/Z.145 – Сценарий использования – Передача запроса на отображение

11.3.7.2 Фрагмент TTCN-3

```

module AModule {
  ...
  type port A { ... }
  type component CA { port A a }
  type component CB { port A a }
  ...
  testcase ATestCase(...)runs on CA system CB {
    var CA ptc := CA.create;
    ... //поведение тестового примера
    map(ptc:a,system:a);
    ...
  }
  ...
}

```

11.3.8 Сценарий использования: передача запроса на прекращение отображения/разъединение

В сценарии на Рисунке 23 показано, как прекращается отображение портов. Запрос на прекращение отображения порта передается на ТЕ, где окончательно производится прекращение отображения. Передача запросов на разъединение осуществляется аналогично.

11.3.8.1 Диаграмма последовательности

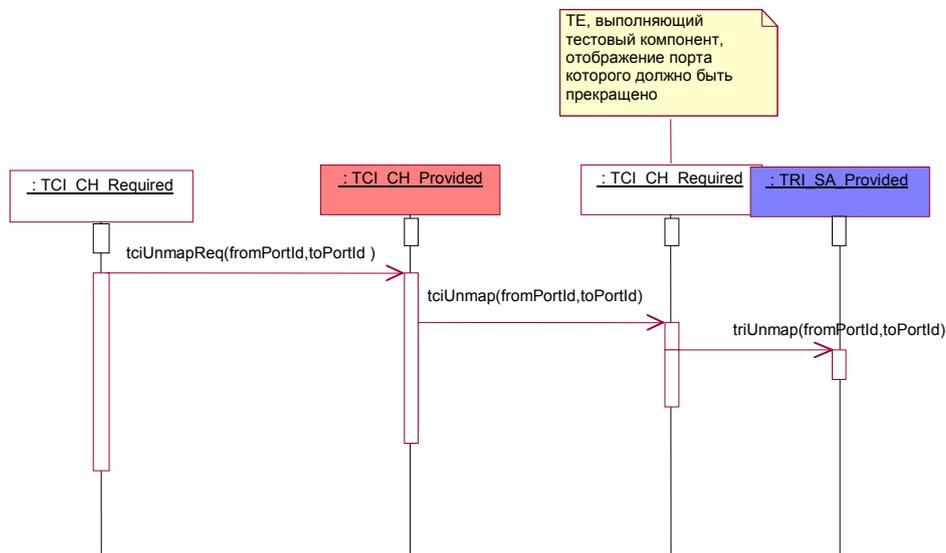


Рисунок 23/Z.145 – Сценарий использования – Передача запроса на прекращение отображения

11.3.8.2 Фрагмент TTCN-3

```
module AModule {
  ...
  type port A { ... }
  type component CA { port A a }
  type component CB { port A a }
  ...
  testcase ATestCase(...) runs on CA system CB {
    var CA ptc := CA.create;
    ... //поведение тестового примера
    unmap(ptc:a, system:a);
  }
  ...
}
```

11.4 Завершение тестовых примеров и управления

11.4.1 Сценарий использования: останов тестового примера

В сценарии на Рисунке 24 показано, как во время выполнения тестового примера администрирование тестом останавливает тестовый пример. Как только ТМ получил информацию о запущенном тестовом примере, может быть запрошен останов тестового примера до приема информации о том, что тестовый пример был завершен. При останове тестового примера будут остановлены все параллельные тестовые компоненты, а тестовая система будет возвращена в исходное состояние.

11.4.1.1 Диаграмма последовательности

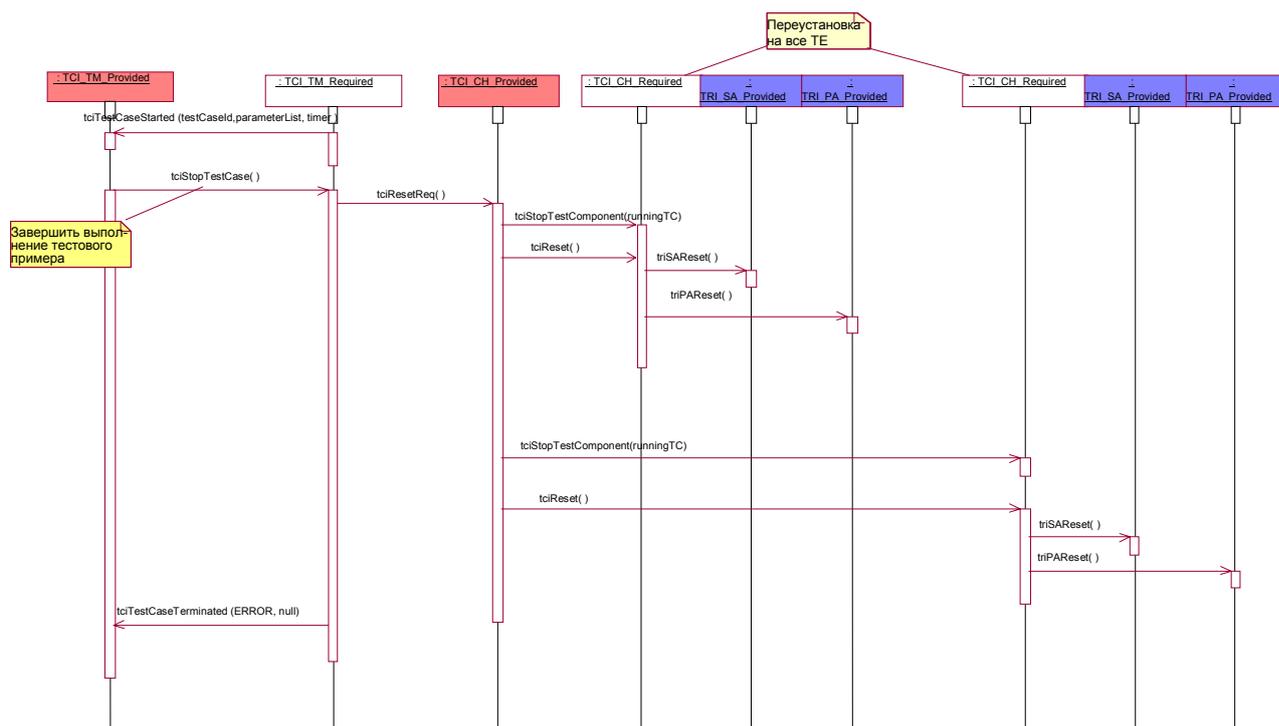


Рисунок 24/Z.145 – Сценарий использования – Останов тестового примера

11.4.1.2 Фрагмент TTCN-3

В TTCN-3 нет программы, касающейся того, как ТМ собирается реализовать завершение тестового примера. Этот вопрос выходит за рамки TTCN-3.

11.4.2 Сценарий использования: останов управления

В сценарии на Рисунке 25 показано, как во время выполнения управляющей части администрирование тестом останавливает управляющую часть. Управляющая часть может быть остановлена между запуском управления и его завершением. Если во время выполнения тестового примера управляющая часть получает запрос на останов тестового примера, то выполнение тестового примера останавливается. Кроме того, тестовая система устанавливается в исходное состояние, как показано на Рисунке 24.

11.4.2.1 Диаграмма последовательности

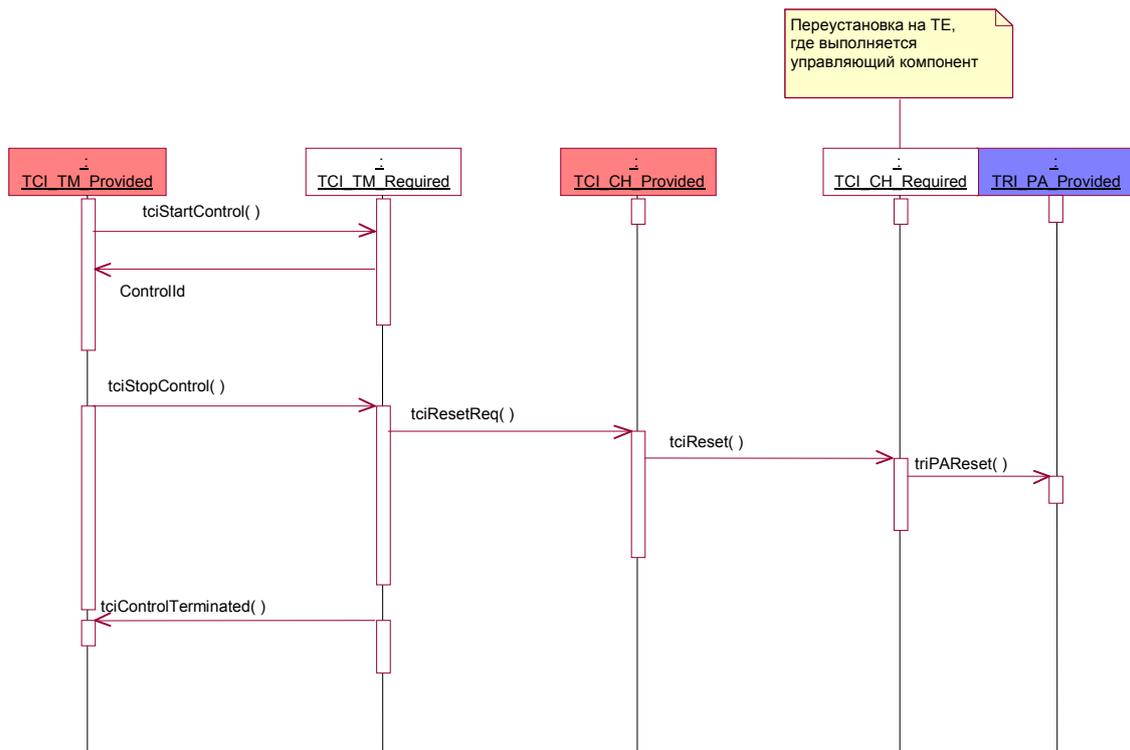


Рисунок 25/Z.145 – Сценарий использования – Останов управления

11.4.2.2 Фрагмент TTCN-3

Вопрос останова управляющей части с помощью администрирования тестом выходит за рамки TTCN-3, так что соответствующего фрагмента TTCN-3 не существует.

11.4.3 Сценарий использования: завершение управления после ошибки

В сценарии на Рисунке 26 показана обработка ситуаций с ошибками во время выполнения управляющей части, когда тестовый пример не выполняется. Администрирование тестом получает информацию о ситуации с ошибкой и должно тогда явным образом завершить выполнение управляющей части. По завершении выполнения управляющей части тестовая система будет установлена в исходное состояние.

11.4.3.1 Диаграмма последовательности

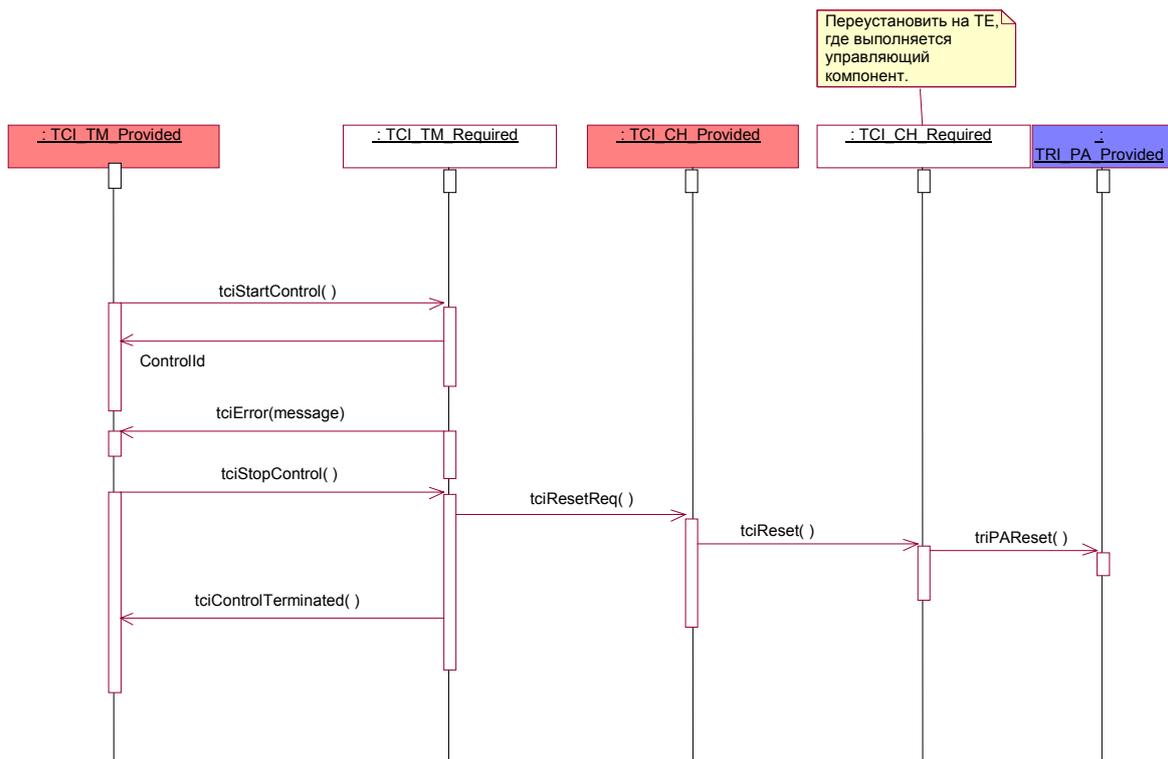


Рисунок 26/Z.145 – Сценарий использования – Завершение управления после ошибки

11.4.3.2 Фрагмент TTCN-3

Для этого сценария отсутствует фрагмент TTCN-3, поскольку ситуации с ошибками являются исключительными случаями в тестовой системе, а не в концепции TTCN-3 как таковой. Скорее, в семантике TTCN-3 описываются различные возможные ситуации с ошибками в тестовой системе.

11.4.4 Сценарий использования: завершение тестового примера после ошибки

На Рисунке 27 показана обработка ситуаций с ошибками во время непосредственного выполнения тестового примера. Администрирование тестом получает информацию о ситуации с ошибкой. Тогда ТМ должно явным образом завершить выполнение тестового примера. При останове тестового примера будут остановлены параллельные тестовые компоненты, а тестовая система устанавливается в исходное состояние.

11.4.4.1 Диаграмма последовательности

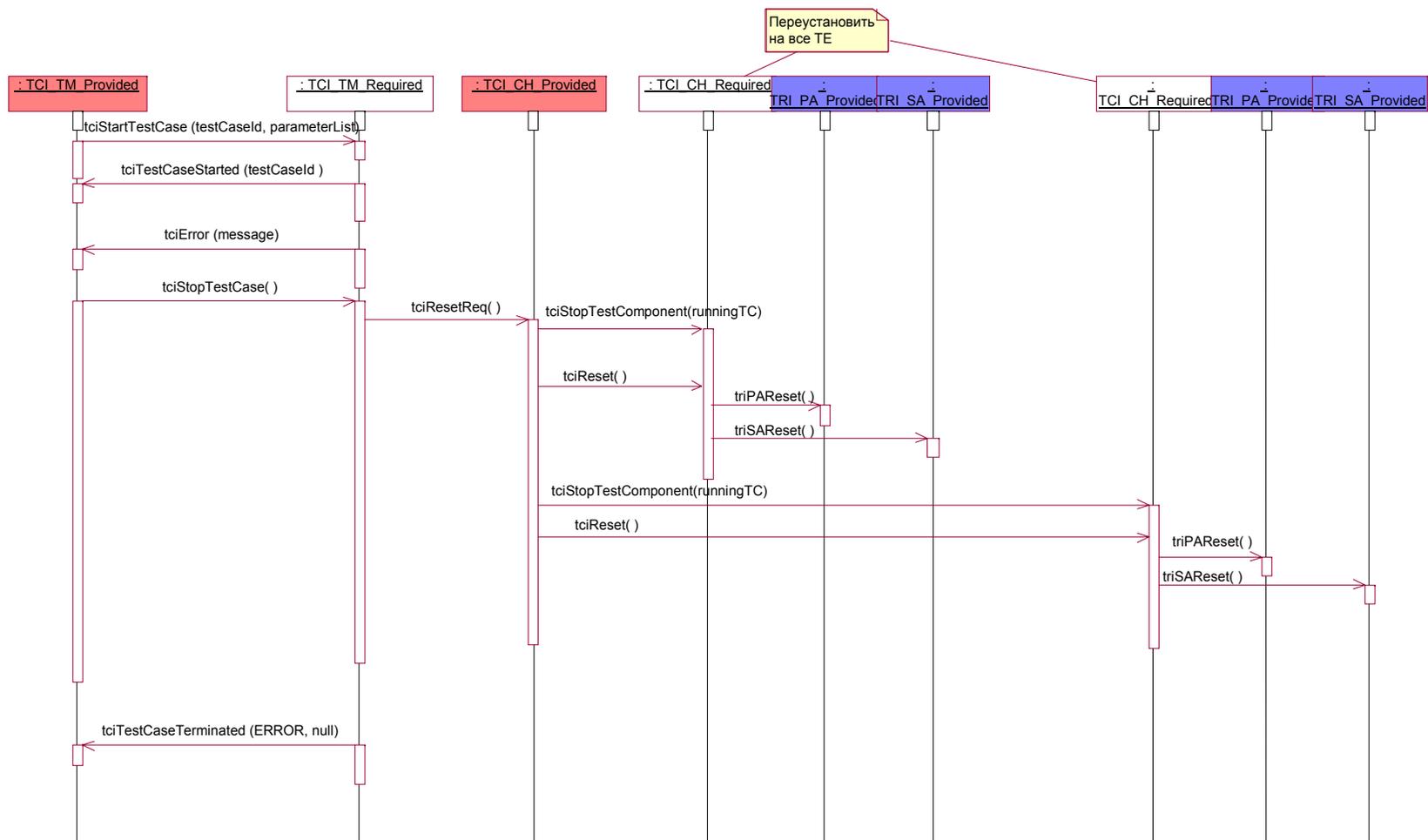


Рисунок 27/Z.145 – Сценарий использования – Завершение тестового примера после ошибки

11.4.4.2 Фрагмент TTCN-3

Для этого сценария отсутствует фрагмент TTCN-3, поскольку ситуации с ошибками являются исключительными случаями в тестовой системе, а не в концепции TTCN-3 как таковой. Скорее, в семантике TTCN-3 описываются различные возможные ситуации с ошибками в тестовой системе.

11.4.5 Сценарий использования: установка в исходное состояние

В сценарии на Рисунке 28 показана установка тестовой системы в исходное состояние (переустановка). В этом случае переустанавливаются все используемые ТЕ вместе с их адаптерами системы (SA) и адаптерами платформы (PA) интерфейса TRI.

11.4.5.1 Диаграмма последовательности

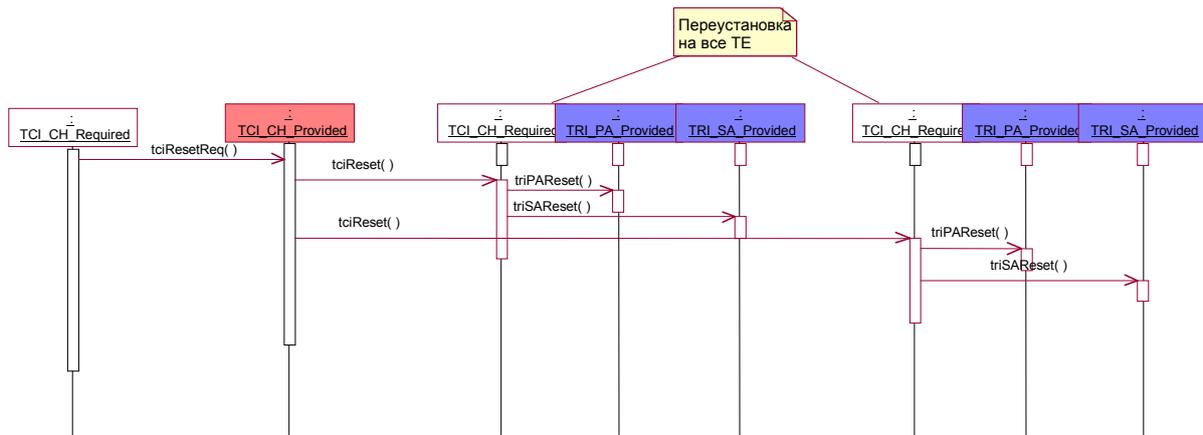


Рисунок 28/Z.145 – Сценарий использования – Установка в исходное состояние

11.4.5.2 Фрагмент TTCN-3

Фрагмента TTCN-3 для этого сценария не существует, поскольку установка в исходное состояние, требуемая после ситуаций с ошибками, является исключительным случаем в тестовой системе, а не в концепции TTCN-3 как таковой.

11.5 Связь

11.5.1 Сценарий использования: локальная межкомпонентная связь

В сценарии на Рисунке 29 показана связь между тестовыми компонентами (главным тестовым компонентом или параллельными тестовыми компонентами), которые располагаются в одном и том же узле. Запрос на связь передается в интерфейс TCI-CH, который затем решает, где поставить в очередь этот шаблон связи. В данном случае связь осуществляется локально через ТЕ на том же узле. В этом сценарии показана связь на базе сообщений, когда используется операция send – причем сценарий один и тот же для операций call, reply и raise.

11.5.1.1 Диаграмма последовательности

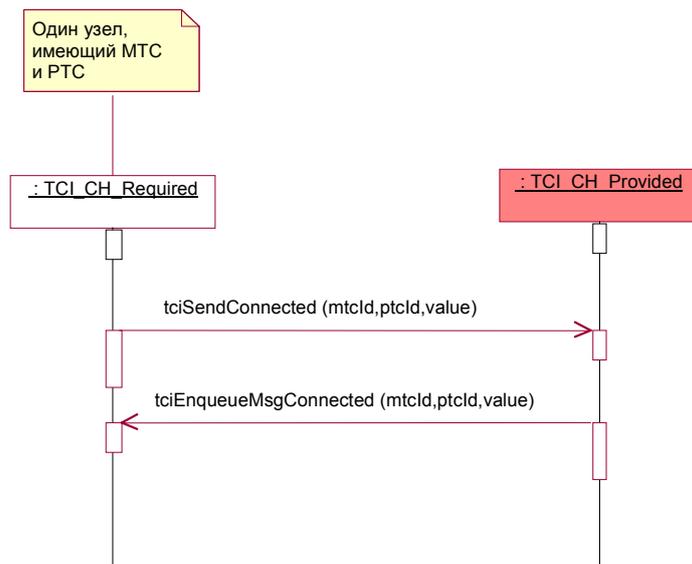


Рисунок 29/Z.145 – Сценарий использования – Локальная межкомпонентная связь

11.5.1.2 Фрагмент TTCN-3

```
module AModule {
  ...
  type port APortType message { ... }
  ...
  type component ATCType {
    ...
    APortType APort;
    ...
  }
  ...
  template AType AMessageTemplate { ... }
  ...
  function APTCBehaviour(...) runs on APTCType {
    ... //поведение PTC
  }
  ...
  testcase ATestCase(...) runs on ATCType... {
    ... //поведение тестового примера
    var ATCType PTC1:= ATCType.create;
    connect (PTC1:APort,mtc:APort);
    ...
    PTC1.start(APTCBehaviour(...));
    APort.send(AMessageTemplate); //передача данных тестовому компоненту
    ...
  }
  ...
}
```

11.5.2 Сценарий использования: межузловая связь между тестовыми компонентами

В сценарии на Рисунке 30 показана связь между тестовыми компонентами (главным тестовым компонентом или параллельными тестовыми компонентами), которые располагаются в разных узлах. Запрос на связь передается в интерфейс TCI-CH, который затем решает, где поставить в очередь этот шаблон связи. В данном случае связь осуществляется удаленно через TE на другом узле. В этом сценарии показана связь на базе сообщений, когда используется операция send – причем сценарий один и тот же для операций call, reply и raise.

11.5.2.1 Диаграмма последовательности

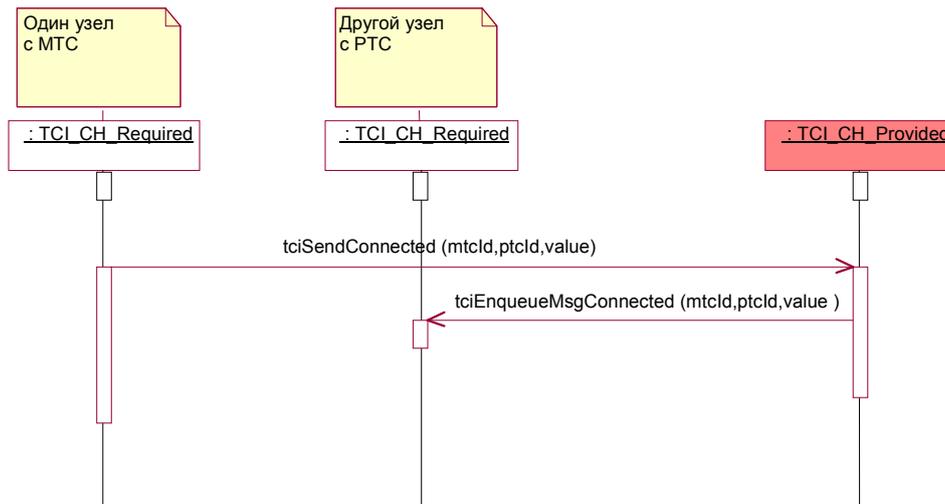


Рисунок 30/Z.145 – Сценарий использования – Межузловая связь между тестовыми компонентами

11.5.2.2 Фрагмент TTCN-3

```
module AModule {
  ...
  type port APortType message { ... }
  ...
  type component ATCType {
    ...
    APortType APort;
    ...
  }
  ...
  template AType AMessageTemplate { ... }
  ...
  function APTCBehaviour(...) runs on APTCType {
    ... //поведение РТС
  }
  ...
  testcase ATestCase(...) runs on ATCType... {
    ... //поведение тестового примера
    var ATCType PTC1:= ATCType.create;
    connect(PTC1:APort,mtc:APort);
    ...
    PTC1.start(APTCBehaviour(...));
    APort.send(AMessageTemplate); //передача данных тестовому компоненту
    ...
  }
  ...
}
```

11.5.3 Сценарий использования: кодирование

В сценарии на Рисунке 31 показано кодирование данных, которые посылаются системе SUT. Закодированные данные принимаются от объекта кодирования/декодирования через TCI-CD. Закодированное значение посылается системе SUT через TRI-SA. Сценарий один и тот же для операций call, reply и raise .

11.5.3.1 Диаграмма последовательности

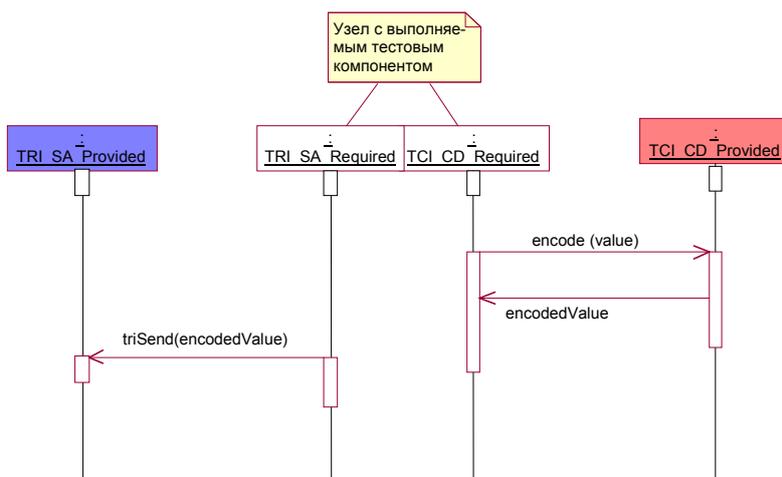


Рисунок 31/Z.145 – Сценарий использования – Кодирование

11.5.3.2 Фрагмент TTCN-3

```
module AModule {
  ...
  type port APortType message { ... }
  ...
  type component APTCType {
    ...
    APortType APort;
    ...
  }
  ...
  template AType AMessageTemplate { ... }
  ...
  testcase ATestCase(...) runs on APTCType system APTCType {
    ... //поведение тестового примера
    map(mtc:APort, system:APort);
    ...
    APort.send(AMessageTemplate); //передача данных системе SUT
    ...
  }
  ...
} with { encoding = '...' }
```

11.5.4 Сценарий использования: декодирование

На Рисунке 32 показан сценарий декодирования данных, которые принимаются от системы SUT через интерфейс TRI-SA. Декодированные данные принимаются от объекта кодирования/декодирования через интерфейс TCI-CD. Этот сценарий является одним и тем же для операций receive, getcall, getreply, catch и check.

11.5.4.1 Диаграмма последовательности

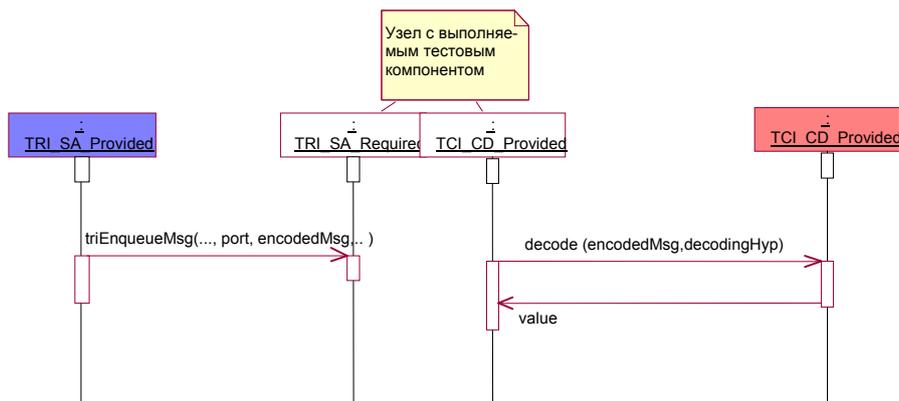


Рисунок 32/Z.145 – Сценарий использования – Декодирование

11.5.4.2 Фрагмент TTCN-3

```
module AModule {
  ...
  type port APortType message { ... }
  ...
  type component APTCType {
    ...
    APortType APort;
    ...
  }
  ...
  template AType AMessageTemplate { ... }
  ...
  testcase ATestCase(...) runs on APTCType system APTCType {
    ... //поведение тестового компонента
    map(mtc:APort, system:APort);
    ...
    APort.receive(AMessageTemplate); //прием данных от системы SUT
    ...
  }
  ...
} with { encoding = '...' }
```

Приложение А

Спецификация TCI на базе языка IDL

В данном приложении приводится определение интерфейсов управления TTCN-3 на базе языка описания интерфейса (IDL).

```
// *****
// * Определения интерфейса для интерфейсов управления TTCN-3
// *****

module tciInterface {

    /* Упреждающее объявление */
    interface Value;
    interface Type;

    // *****
    // * Типы данных, взятые из определений интерфейса TRI
    // *****

    // Соединение
    native TriPortIdType ;
    native TriPortIdListType;
    native TriComponentIdType ;
    native TriComponentIdListType;

    // Связь
    native TriMessageType;
    native TriParameterType;
    native TriParameterListType;
    native TriAddressType;
    native TriAddressListType;
    native TriExceptionType;
    native TriSignatureIdType;

    // Разное
    native TriStatusType;
    native TriTimerIdType;
    native TriTimerDurationType;

    // *****
    // * Основные абстрактные типы данных
    // *****

    // Базовые определения
    native TBoolean;
    native TFloat;
    native TChar;
    native TInteger;
    native TString;
    native TUniversalChar;
    typedef sequence <TString> TStringSeq;
    native TObjid;

    struct QualifiedName {
        TString moduleName;
        TString baseName;
    };

    // Основные абстрактные типы данных интерфейса TCI
    typedef QualifiedName TciBehaviourIdType;
    typedef QualifiedName TciModuleIdType;
    typedef QualifiedName TciModuleParameterIdType;
    typedef QualifiedName TciTestCaseIdType;

    enum TciParameterPassingModeType {
        IN_MODE,
        OUT_MODE,
        INOUT_MODE
    };

    struct TciParameterType {
        TciModuleParameterIdType parameterName;
        Value parameterValue;
        TciParameterPassingModeType mode;
    };
};
```

```

typedef sequence <TciParameterType> TciParameterListType;

struct TciParameterTypeType {
    Type parameterType;
    TciParameterPassingModeType mode;
};

typedef sequence <TciParameterTypeType> TciParameterTypeListType;

struct TciModuleParameterType {
    TciModuleParameterIdType parameterName;
    Value defaultValue;
};

typedef sequence <TciModuleIdType> TciModuleIdListType ;

typedef sequence <TciModuleParameterType> TciModuleParameterListType;

typedef sequence <TciTestCaseIdType> TciTestCaseIdListType;

enum TciTestComponentKindType {
    MTC,
    PTC,
    CONTROL
};

enum TciTypeClassType {
    ADDRESS_CLASS,
    ANYTYPE_CLASS,
    BITSTRING_CLASS,
    BOOLEAN_CLASS,
    CHAR_CLASS,
    CHARSTRING_CLASS,
    COMPONENT_CLASS,
    ENUMERATED_CLASS,
    FLOAT_CLASS,
    HEXSTRING_CLASS,
    INTEGER_CLASS,
    OBJID_CLASS,
    OCTETSTRING_CLASS,
    RECORD_CLASS,
    RECORDOF_CLASS,
    SET_CLASS,
    SETOF_CLASS,
    UNION_CLASS,
    UNIVERSALCHAR_CLASS,
    UNIVERSALCHARSTRING_CLASS,
    VERDICT_CLASS
};

// *****
// * Абстрактные типы данных и значения TTCN-3
// *****

// Абстрактный тип данных "Type"
interface Type {
    TciModuleIdType getDefiningModule ();
    TString getName ();
    TciTypeClassType getTypeClass ();
    Value newInstance ();
    TString getTypeEncoding ();
    TString getTypeEncodingVariant ();
    TStringSeq getTypextension ();
};

// Абстрактные значения TTCN-3
interface Value {
    TString getValueEncoding ();
    TString getValueEncodingVariant ();
    Type getType ();
    TBoolean notPresent ();
};

interface RecordOfValue : Value {
    Value getField (in TInteger position);
    void setField (
        in TInteger position,

```

```

        in Value value
    );
    void    appendField (in Value value);
    Type    getElementType ();
    TInteger getLength ();
    void    setLength (in TInteger len);
};

interface RecordValue : Value {
    Value    getField (in TString fieldName);
    void    setField (
        in TString fieldName,
        in Value value
    );
    TStringSeq getFieldNames ();
};

interface VerdictValue : Value {
    TInteger getVerdict ();
    void    setVerdict (in TInteger verdict);
};

interface BitstringValue : Value {
    TString    getString ();
    void    setString (in TString value);
    TInteger    getBit (in TInteger position);
    void    setBit (
        in TInteger position,
        in TInteger value
    );
    TInteger    getLength ();
    void    setLength (in TInteger len);
};

interface OctetstringValue : Value {
    TString    getString ();
    void    setString (in TString value);
    TInteger    getOctet (in TInteger position);
    void    setOctet (
        in TInteger position,
        in TInteger value
    );
    TInteger    getLength ();
    void    setLength (in TInteger len);
};

interface FloatValue : Value {
    TFloat    getFloat ();
    void    setFloat (in TFloat value);
};

interface HexstringValue : Value {
    TString    getString ();
    void    setString (in TString value);
    TInteger    getHex (in TInteger position);
    void    setHex (
        in TInteger position,
        in TInteger value
    );
    TInteger    getLength ();
    void    setLength (in TInteger len);
};

interface ObjidValue : Value {
    TObjid    getObjid ();
    void    setObjid (in TObjid value);
};

interface EnumeratedValue : Value {
    void    setEnum (in TString enumValue);
    TString    getEnum ();
};

interface IntegerValue : Value {
    TInteger    getInt ();
    void    setInt (in TInteger value);
};

interface CharValue : Value {
    TChar    getChar ();
    void    setChar (in TChar value);
};

```

```

};

interface CharstringValue : Value {
    TString getString ();
    void setString (in TString value);
    TChar getChar (in TInteger position);
    void setChar (
        in TInteger position,
        in TChar value
    );
    TInteger getLength ();
    void setLength (in TInteger len);
};

interface BooleanValue : Value {
    TBoolean getBoolean ();
    void setBoolean (in TBoolean value);
};

interface UniversalCharValue : Value {
    TUniversalChar getUniversalChar ();
    void setUniversalChar (in TUniversalChar value);
};

interface UniversalCharstringValue : Value {
    TString getString ();
    void setString (in TString value);
    TUniversalChar getChar (in TInteger position);
    void setChar (
        in TInteger position,
        in TUniversalChar value
    );
    TInteger getLength ();
    void setLength (in TInteger len);
};

interface UnionValue : Value {
    Value getVariant (in TString variantName);
    void setVariant (
        in TString variantName,
        in Value value
    );
    TString getPresentVariantName ();
    TStringSeq getVariantNames ();
};

// *****
// * Абстрактные типы регистрации
// *****

interface TciValueTemplate : Value {
    boolean isOmit ();
    boolean isAny();
    boolean isAnyOrOmit();
    TString getTemplateDef();
};

interface TciNonValueTemplate {
    boolean isAny();
    boolean isAll();
    TString getTemplateDef();
};

typedef sequence <Value> TciValueList;

struct TciValueDifference
{
    TString desc;
    Value val;
    TciValueTemplate tmpl;
};

typedef sequence <TciValueDifference> TciValueDifferenceList;

// *****
// Интерфейс кодирования/декодирования
// - Требуемый
// *****

```

```

interface TCI_CD_Required {
    Type getTypeForName (in TString typeName);
    Type getInteger ();
    Type getFloat ();
    Type getBoolean ();
    Type getChar ();
    Type getUniversalChar ();
    Type getObjid ();
    Type getCharstring ();
    Type getUniversalCharstring ();
    Type getHexstring ();
    Type getBitstring ();
    Type getOctetstring ();
    Type getVerdict ();
    void tciErrorReq (in TString message);
};

// *****
// Интерфейс кодирования/декодирования
// - Предоставляемый
// *****

interface TCI_CD_Provided {
    Value decode (
        in TriMessageType message,
        in Type decodingHypothesis
    );
    TriMessageType encode (in Value value);
};

// *****
// Интерфейс администрирования тестом
// - Требуемый
// *****

interface TCI_TM_Required : TCI_CD_Required {
    void tciRootModule (in TciModuleIdType moduleName);
    TciModuleIdListType getImportedModules();
    TciModuleParameterListType tciGetModuleParameters (in TciModuleIdType moduleName);
    TciTestCaseIdListType tciGetTestCases ();
    TciParameterTypeListType tciGetTestCaseParameters (
        in TciTestCaseIdType testCaseId
    );
    TriPortIdListType tciGetTestCaseTSI (
        in TciTestCaseIdType testCaseId
    );
    void tciStartTestCase (
        in TciTestCaseIdType testCaseId,
        in TciParameterListType parameterList
    );
    void tciStopTestCase ();
    TriComponentIdType tciStartControl ();
    void tciStopControl ();
};

// *****
// Интерфейс администрирования тестом
// - Предоставляемый
// *****

interface TCI_TM_Provided {
    void tciTestCaseStarted (
        in TciTestCaseIdType testCaseId,
        in TciParameterListType parameterList,
        in TFloat timer
    );
    void tciTestCaseTerminated (
        in VerdictValue verdict,
        in TciParameterListType parameterList
    );
    void tciControlTerminated ();
    Value tciGetModulePar (
        in TciModuleParameterIdType parameterId
    );
    void tciLog (
        in TriComponentIdType testComponentId,
        in TString message
    );
    void tciError (in TString message);
};

```

```

};

// *****
// Интерфейс обработки компонента
// - Требуемый
// *****

interface TCI_CH_Required : TCI_CD_Required {
    void tciEnqueueMsgConnected (
        in TriPortIdType sender,
        in TriComponentIdType receiver,
        in Value receivedMessage
    );
    void tciEnqueueCallConnected (
        in TriPortIdType sender,
        in TriComponentIdType receiver,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList
    );
    void tciEnqueueReplyConnected (
        in TriPortIdType sender,
        in TriComponentIdType receiver,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList,
        in Value returnValue
    );
    void tciEnqueueRaiseConnected (
        in TriPortIdType sender,
        in TriComponentIdType receiver,
        in TriSignatureIdType signature,
        in Value except
    );
    TriComponentIdType tciCreateTestComponent (
        in TciTestComponentKindType kind,
        in Type componentType,
        in String name
    );
    void tciStartTestComponent (
        in TriComponentIdType comp,
        in TciBehaviourIdType behavior,
        in TciParameterListType parameterList
    );
    void tciStopTestComponent (
        in TriComponentIdType comp
    );
    void tciConnect (
        in TriPortIdType fromPort,
        in TriPortIdType toPort
    );
    void tciDisconnect (
        in TriPortIdType fromPort,
        in TriPortIdType toPort
    );
    void tciTestComponentTerminated (
        in TriComponentIdType comp,
        in VerdictValue verdict
    );
    TBoolean tciTestComponentRunning (
        in TriComponentIdType comp
    );
    TriComponentIdType tciGetMTC ();
    void tciMap (
        in TriPortIdType fromPort,
        in TriPortIdType toPort
    );
    void tciUnmap (
        in TriPortIdType fromPort,
        in TriPortIdType toPort
    );
    void tciExecuteTestCase (
        in TciTestCaseIdType testCaseId,
        in TriPortIdListType tsiPortList
    );
    TBoolean tciTestComponentDone (
        in TriComponentIdType comp
    );
    void tciReset ();
};

// *****

```

```

// Интерфейс обработки компонента
// - Предоставляемый
// *****

interface TCI_CH_Provided {
    void tciSendConnected (
        in TriPortIdType sender,
        in TriComponentIdType receiver,
        in Value sendMessage
    );
    void tciSendConnectedBC (
        in TriPortIdType sender,
        in Value sendMessage
    );
    void tciSendConnectedMC (
        in TriPortIdType sender,
        in TriComponentIdListType receivers,
        in Value sendMessage
    );

    void tciCallConnected (
        in TriPortIdType sender,
        in TriComponentIdType receiver,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList
    );
    void tciCallConnectedBC (
        in TriPortIdType sender,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList
    );
    void tciCallConnectedMC (
        in TriPortIdType sender,
        in TriComponentIdListType receivers,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList
    );

    void tciReplyConnected (
        in TriPortIdType sender,
        in TriComponentIdType receiver,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList,
        in Value returnValue
    );
    void tciReplyConnectedBC (
        in TriPortIdType sender,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList,
        in Value returnValue
    );
    void tciReplyConnectedMC (
        in TriPortIdType sender,
        in TriComponentIdListType receivers,
        in TriSignatureIdType signature,
        in TciParameterListType parameterList,
        in Value returnValue
    );

    void tciRaiseConnected (
        in TriPortIdType sender,
        in TriComponentIdType receiver,
        in TriSignatureIdType signature,
        in Value except
    );
    void tciRaiseConnectedBC (
        in TriPortIdType sender,
        in TriSignatureIdType signature,
        in Value except
    );
    void tciRaiseConnectedMC (
        in TriPortIdType sender,
        in TriComponentIdListType receivers,
        in TriSignatureIdType signature,
        in Value except
    );

    TriComponentIdType tciCreateTestComponentReq (
        in TciTestComponentKindType kind,
        in Type componentType,
        in String name
    );
}

```

```

    );
void tciStartTestComponentReq (
    in TriComponentIdType comp,
    in TciBehaviourIdType behavior,
    in TciParameterListType parameterList
);
void tciStopTestComponentReq (
    in TriComponentIdType comp
);
void tciConnectReq (
    in TriPortIdType fromPort,
    in TriPortIdType toPort
);
void tciDisconnectReq (
    in TriPortIdType fromPort,
    in TriPortIdType toPort
);
void tciTestComponentTerminatedReq (
    in TriComponentIdType comp,
    in VerdictValue verdict
);
TBoolean tciTestComponentRunningReq (
    in TriComponentIdType comp
);
TriComponentIdType tciGetMTCReq ();
void tciMapReq (
    in TriPortIdType fromPort,
    in TriPortIdType toPort
);
void tciUnmapReq (
    in TriPortIdType fromPort,
    in TriPortIdType toPort
);
void tciExecuteTestCaseReq (
    in TciTestCaseIdType testCaseId,
    in TriPortIdListType tsiPortList
);
void tciResetReq ();
TBoolean tciTestComponentDoneReq (
    in TriComponentIdType comp
);
};

// *****
// Интерфейс тестовой регистрации
// - Предоставляемый
// *****

interface TCI_TL_Provided {
    void tliTcExecute(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TciTestCaseIdType tcId,
        in TriParameterListType pars, in TriTimerDurationType dur
    );
    void tliTcStart(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TciTestCaseIdType tcId,
        in TriParameterListType pars, in TriTimerDurationType dur
    );
    void tliTcStop(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c
    );
    void tliTcStarted(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TciTestCaseIdType tcId,
        in TriParameterListType pars, in TriTimerDurationType dur
    );
    void tliTcTerminated(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TciTestCaseIdType tcId,
        in TriParameterListType pars, in VerdictValue outcome);

    void tliCtrlStart(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c
    );
    void tliCtrlStop(
        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c
    );
};

```

```

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

void tliMSend_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TriAddressType address, in TriStatusType encoderFailure,
    in TriMessageType msg, in TriStatusType transmissionFailure
);
void tliMSend_m_BC(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TriStatusType encoderFailure, in TriMessageType msg,
    in TriStatusType transmissionFailure
);
void tliMSend_m_MC(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TriAddressListType addresses, in TriStatusType encoderFailure,
    in TriMessageType msg, in TriStatusType transmissionFailure
);

void tliMSend_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TriComponentIdType to, in TriStatusType transmissionFailure
);
void tliMSend_c_BC(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TriStatusType transmissionFailure
);
void tliMSend_c_MC(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TriComponentIdListType toList, in TriStatusType transmissionFailure);

void tliMDetected_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in TriMessageType msg,
    in TriAddressType address
);
void tliMDetected_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TriComponentIdType from
);
void tliMMismatch_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TciValueTemplate msgTpl, in TciValueDifferenceList diffs,
    in TriAddressType address, in TciValueTemplate addressTpl
);
void tliMMismatch_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TciValueTemplate msgTpl, in TciValueDifferenceList diffs,
    in TriComponentIdType from, in TciNonValueTemplate fromTpl
);
void tliMReceive_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TciValueTemplate msgTpl, in TriAddressType address,
    in TciValueTemplate addressTpl
);
void tliMReceive_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port, in Value msgValue,
    in TciValueTemplate msgTpl, in TriComponentIdType from,
    in TciNonValueTemplate fromTpl
);

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

```

```

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

void tliPrCall_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in TriComponentIdType to, in TriStatusType transmissionFailure
    );
void tliPrCall_c_BC(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in TriStatusType transmissionFailure
    );
void tliPrCall_c_MC(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in TriComponentIdListType toList, in TriStatusType transmissionFailure
    );

void tliPrGetCallDetected_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TriParameterListType pars,
    in TriAddressType address
    );
void tliPrGetCallDetected_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in TriComponentIdType from
    );
void tliPrGetCallMismatch_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in TciValueTemplate parsTmpl, in TciValueDifferenceList diffs,
    in TriAddressType address, in TciValueTemplate addressTmpl
    );
void tliPrGetCallMismatch_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in TciValueTemplate parsTmpl, in TciValueDifferenceList diffs,
    in TriComponentIdType from, in TciNonValueTemplate fromTmpl
    );
void tliPrGetCall_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in TciValueTemplate parsTmpl, in TriAddressType address,
    in TciValueTemplate addressTmpl
    );
void tliPrGetCall_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in TciValueTemplate parsTmpl, in TriComponentIdType from,
    in TciNonValueTemplate fromTmpl
    );

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

```

```

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

void tliPrReply_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in Value replValue, in TriComponentIdType to,
    in TriStatusType transmissionFailure
);
void tliPrReply_c_BC(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in Value parsValue, in Value replValue,
    in TriStatusType transmissionFailure
);
void tliPrReply_c_MC(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in Value parsValue, in Value replValue,
    in TriComponentIdListType toList, in TriStatusType transmissionFailure
);

void tliPrGetReplyDetected_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TriParameterType repl,
    in TriAddressType address
);
void tliPrGetReplyDetected_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in Value replValue,
    in TriComponentIdType from
);
void tliPrGetReplyMismatch_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in Value replValue, in TciValueTemplate replyTmpl,
    in TciValueDifferenceList diffs, in TriAddressType address,
    in TciValueTemplate addressTmpl
);
void tliPrGetReplyMismatch_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in Value replValue, in TciValueTemplate replyTmpl,
    in TciValueDifferenceList diffs, in TriComponentIdType from,
    in TciNonValueTemplate fromTmpl
);
void tliPrGetReply_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in Value replValue, in TciValueTemplate replyTmpl,
    in TriAddressType address, in TciValueTemplate addressTmpl
);
void tliPrGetReply_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in Value replValue, in TciValueTemplate replyTmpl,
    in TriComponentIdType from, in TciNonValueTemplate fromTmpl
);

void tliPrRaise_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,

```

```

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

void tliPrRaise_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in Value excValue, in TriComponentIdType to,
    in TriStatusType transmissionFailure
);
void tliPrCatchDetected_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TriExceptionType exc,
    in TriAddressType address
);
void tliPrCatchDetected_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in Value excValue,
    in TriComponentIdType from
);
void tliPrCatchMismatch_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in Value excValue, in TciValueTemplate excTmpl,
    in TciValueDifferenceList diffs, in TriAddressType address,
    in TciValueTemplate addressTmpl
);
void tliPrCatchMismatch_c(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in Value excValue, in TciValueTemplate excTmpl,
    in TciValueDifferenceList diffs, in TriComponentIdType from,
    in TciNonValueTemplate fromTmpl
);
void tliPrCatch_m(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port,
    in TriSignatureIdType signature, in TciParameterListType parsValue,
    in Value excValue, in TciValueTemplate excTmpl,
    in TriAddressType address, in TciValueTemplate addressTmpl);

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

```

```

        in TString am, in TInteger ts, in TString src, in TInteger line,
        in TriComponentIdType c, in TriComponentIdType comp,
        in String name
    );
void tliCStart(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriComponentIdType comp,
    in TciBehaviourIdType name, in TciParameterListType parsValue
);
void tliCRunning(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriComponentIdType comp, in TBoolean status
);
void tliCALive(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c,
    in TriComponentIdType comp, in TBoolean status
);
void tliCStop(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriComponentIdType comp
);
void tliCKill(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriComponentIdType comp
);
void tliCDoneMismatch(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TciNonValueTemplate compTmpl
);
void tliCKilledMismatch(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TciNonValueTemplate compTmpl
);
void tliCDone(in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TciNonValueTemplate compTmpl
);
void tliCKilled(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TciNonValueTemplate compTmpl
);
void tliCTerminated(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in VerdictValue verdict
);
void tliPConnect(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1,
    in TriComponentIdType c2, in TriPortIdType port2
);
void tliPDisconnect(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1,
    in TriComponentIdType c2, in TriPortIdType port2
);
void tliPMap(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1,
    in TriComponentIdType c2, in TriPortIdType port2
);
void tliPUnmap(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriComponentIdType c1, in TriPortIdType port1,
    in TriComponentIdType c2, in TriPortIdType port2
);
void tliPClear(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port
);
void tliPStart(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port
);
void tliPStop(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port
);
void tliPHalt(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriPortIdType port
);

```

```

void tliEncode(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in Value val, in TriStatusType encoderFailure,
    in TriMessageType msg, in TString codec
);
void tliDecode(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriMessageType msg,
    in TriStatusType decoderFailure, in Value val, in TString codec
);
void tliTTimeoutDetected(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriTimerIdType timer
);
void tliTTimeoutMismatch(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TciNonValueTemplate timerTpl
);
void tliTTimeout(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TciNonValueTemplate timerTpl
);
void tliTStart(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriTimerIdType timer,
    in TriTimerDurationType dur
);
void tliTStop(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriTimerIdType timer
);
void tliTRead(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriTimerIdType timer,
    in TriTimerDurationType elapsed
);
void tliTRunning(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TriTimerIdType timer, in TBoolean status
);
void tliSEnter(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TString name, in TciParameterListType parsValue,
    in TString kind
);
void tliSLeave(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TString name, in Value returnValue,
    in TString kind
);
void tliVar(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TString name, in Value varValue
);
void tliModulePar(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TString name, in Value parValue
);
void tliGetVerdict(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in VerdictValue verdict
);
void tliSetVerdict(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in VerdictValue verdict
);
void tliLog(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TciValueList log
);
void tliAEnter(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c
);
void tliALeave(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c
);
void tliADefaults(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c
);

```

```

    );
void tliAActivate(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in TString name, in TciParameterListType pars,
    in Value ref
    );
void tliADeactivate(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c, in Value ref
    );
void tliANomatch(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c
    );
void tliARepeat(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c
    );
void tliAWait(
    in TString am, in TInteger ts, in TString src, in TInteger line,
    in TriComponentIdType c
    );
};
};
};

```

Приложение В

Отображение на языке XML для интерфейса TCI TL provided

В данном приложении приводится определение отображения для интерфейса регистрации в интерфейсе TCI при использовании определений схемы на расширяемом языке разметки (XML).

В.1 Схема для простых типов на языке XML интерфейса TCI-TL

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
  xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
  elementFormDefault="qualified">

  <!--Базовые определения -->
  <xsd:simpleType name="xpath">
    <!--эта строка должна соответствовать XPATH -->
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>

  <xsd:simpleType name="TBoolean">
    <xsd:restriction base="xsd:boolean"/>
  </xsd:simpleType>

  <xsd:simpleType name="TString">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>

  <xsd:simpleType name="TInteger">
    <xsd:restriction base="xsd:integer"/>
  </xsd:simpleType>

  <!-- Разное -->
  <xsd:simpleType name="TriTimerDurationType">
    <xsd:restriction base="xsd:float"/>
  </xsd:simpleType>

  <xsd:simpleType name="TciParameterPassingModeType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="in"/>
      <xsd:enumeration value="inout"/>
      <xsd:enumeration value="out"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="TriStatusType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="TRI_Ok"/>
      <xsd:enumeration value="TRI_Error"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="TciStatusType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="TCI_Ok"/>
      <xsd:enumeration value="TCI_Error"/>
    </xsd:restriction>
  </xsd:simpleType>

</xsd:schema>
```

В.2 Схема для типов на языке XML интерфейса TCI-TL

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
  xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
  xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
  xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
  elementFormDefault="qualified">

  <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values.xsd"
```

```

    schemaLocation="Values.xsd"/>
<xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes.xsd"
    schemaLocation="SimpleTypes.xsd"/>

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

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

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

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

<!-- СВЯЗЬ -->
<xsd:complexType name="TriMessageType">
    <xsd:attribute name="val" type="xsd:hexBinary"/>
</xsd:complexType>

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

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

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

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

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

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

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

```

```

</xsd:complexType>

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

<!-- Базовые определения -->
<xsd:complexType name="QualifiedName">
  <xsd:attribute name="moduleName" type="SimpleTypes:TString" use="required"/>
  <xsd:attribute name="baseName" type="SimpleTypes:TString" use="required"/>
</xsd:complexType>

<!-- основные абстрактные типы данных TCI -->
<xsd:complexType name="TciBehaviourIdType">
  <xsd:sequence>
    <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

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

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

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

<!-- основная структура идентификатора для тестовых компонентов, портов и таймера -->
<xsd:complexType name="Id">
  <xsd:sequence>
    <xsd:element name="name" type="SimpleTypes:TString" minOccurs="1"
maxOccurs="1"/>
    <xsd:element name="id" type="SimpleTypes:TInteger" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="type" type="SimpleTypes:TString" minOccurs="1"
maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```

В.3 Схема для значений на языке XML интерфейса TCI-TL

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
  xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
  xmlns:Templates="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates"
  xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
  xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
  elementFormDefault="qualified">

  <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates.xsd"
    schemaLocation="Templates.xsd"/>
  <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types.xsd"
    schemaLocation="Types.xsd"/>
  <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes.xsd"
    schemaLocation="SimpleTypes.xsd"/>

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

  <xsd:complexType name="Value" mixed="true">
    <xsd:choice>

```

```

<xsd:element name="integer" type="Values:IntegerValue"/>
<xsd:element name="float" type="Values:FloatValue"/>
<xsd:element name="boolean" type="Values:BooleanValue"/>
<xsd:element name="objid" type="Values:ObjidValue"/>

<xsd:element name="verdicttype" type="Values:VerdictValue"/>
<xsd:element name="bitstring" type="Values:BitstringValue"/>
<xsd:element name="hexstring" type="Values:HexstringValue"/>
<xsd:element name="octetstring" type="Values:OctetstringValue"/>
<xsd:element name="charstring" type="Values:CharstringValue"/>
<xsd:element name="universal_charstring"
type="Values:UniversalCharstringValue"/>
<xsd:element name="record" type="Values:RecordValue"/>
<xsd:element name="record_of" type="Values:RecordOfValue"/>
<xsd:element name="set" type="Values:SetValue"/>
<xsd:element name="set_of" type="Values:SetOfValue"/>
<xsd:element name="enumerated" type="Values:EnumeratedValue"/>
<xsd:element name="union" type="Values:UnionValue"/>
<xsd:element name="anytype" type="Values:AnytypeValue"/>
<xsd:element name="address" type="Values:AddressValue"/>
</xsd:choice>
<xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>

<! -- ОСНОВНЫЕ ЭЛЕМЕНТЫ СОБЫТИЯ -->
<xsd:complexType name="IntegerValue">
  <xsd:simpleContent>
    <xsd:extension base="SimpleTypes:TString">
      <xsd:attributeGroup ref="Values:ValueAtts"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

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

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

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

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

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

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

```

```

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

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

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

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

<xsd:complexType name="RecordOfValue">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="integer" type="Values:IntegerValue" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="float" type="Values:FloatValue" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="boolean" type="Values:BooleanValue" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="objid" type="Values:ObjidValue" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="bitstring" type="Values:BitstringValue"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="hexstring" type="Values:HexstringValue"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="octetstring" type="Values:OctetstringValue"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:choice>
</xsd:complexType>

```

```

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

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

<xsd:complexType name="SetOfValue">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="integer" type="Values:IntegerValue" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
      <xsd:element name="float" type="Values:FloatValue" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:choice>

```

```

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

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

<xsd:complexType name="UnionValue">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="integer" type="Values:IntegerValue"/>
    <xsd:element name="float" type="Values:FloatValue"/>
    <xsd:element name="boolean" type="Values:BooleanValue"/>
    <xsd:element name="objid" type="Values:ObjidValue"/>
    <xsd:element name="verdicttype" type="Values:VerdictValue"/>
    <xsd:element name="bitstring" type="Values:BitstringValue"/>
  </xsd:choice>

```

```

<xsd:element name="hexstring" type="Values:HexstringValue"/>
<xsd:element name="octetstring" type="Values:OctetstringValue"/>
<xsd:element name="charstring" type="Values:CharstringValue"/>
<xsd:element name="universal_charstring"
  type="Values:UniversalCharstringValue"/>
<xsd:element name="record" type="Values:RecordValue"/>
<xsd:element name="record_of" type="Values:RecordOfValue"/>
<xsd:element name="set" type="Values:SetValue"/>
<xsd:element name="set_of" type="Values:SetOfValue"/>
<xsd:element name="enumerated" type="Values:EnumeratedValue"/>
<xsd:element name="union" type="Values:UnionValue"/>
<xsd:element name="anytype" type="Values:AnytypeValue"/>
<xsd:element name="address" type="Values:AddressValue"/>
</xsd:choice>
<xsd:attributeGroup ref="Values:ValueAtts"/>
</xsd:complexType>

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

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

```

B.4 Схема для шаблонов на языке XML интерфейса TCI-TL

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates"
  xmlns:Templates="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates"
  xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
  xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
  xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
  elementFormDefault="qualified">

  <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values.xsd"
    schemaLocation="Values.xsd"/>
  <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types.xsd"
    schemaLocation="Types.xsd"/>

```

```

<xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes.xsd"
  schemaLocation="SimpleTypes.xsd"/>

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

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

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

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

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

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

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

```

```

        <xsd:attribute name="tpl" type="SimpleTypes:xpath" use="required"/>
</xsd:complexType>

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

<xsd:complexType name="IntegerTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:IntegerValue">
      <xsd:choice>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="null" type="xsd:string"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="FloatTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:FloatValue">
      <xsd:choice>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="null" type="xsd:string"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="BooleanTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:BooleanValue">
      <xsd:choice>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="null" type="xsd:string"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ObjidTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:ObjidValue">
      <xsd:choice>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="null" type="xsd:string"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="BitstringTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:BitstringValue">
      <xsd:choice>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="null" type="xsd:string"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

<xsd:complexType name="HexstringTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:BitstringValue">
      <xsd:choice>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="null" type="xsd:string"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="OctetstringTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:OctetstringValue">
      <xsd:choice>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="null" type="xsd:string"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="CharstringTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:CharstringValue">
      <xsd:choice>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="null" type="xsd:string"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="UniversalCharstringTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:UniversalCharstringValue">
      <xsd:choice>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="null" type="xsd:string"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="RecordTemplate">
  <xsd:complexContent>
    <xsd:extension base="Values:RecordValue">
      <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="integer" type="Templates:IntegerTemplate"/>
          <xsd:element name="float" type="Templates:FloatTemplate"/>
          <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
          <xsd:element name="objid" type="Templates:ObjidTemplate"/>
          <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
          <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
          <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
          <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
          <xsd:element name="universal_charstring"
            type="Templates:UniversalCharstringTemplate"/>
          <xsd:element name="record" type="Templates:RecordTemplate"/>
          <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
          <xsd:element name="set" type="Templates:SetTemplate"/>
          <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
          <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
          <xsd:element name="union" type="Templates:UnionTemplate"/>
          <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
          <xsd:element name="address" type="Templates:AddressTemplate"/>
          <xsd:element name="omit" type="Templates:omit"/>
          <xsd:element name="any" type="Templates:any"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
    </xsd:choice>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="RecordOfTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:RecordOfValue">
            <xsd:choice>
                <xsd:sequence>
                    <xsd:element name="integer" type="Templates:IntegerTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="float" type="Templates:FloatTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="boolean" type="Templates:BooleanTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="objid" type="Templates:ObjidTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="bitstring" type="Templates:BitstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="hexstring" type="Templates:HexstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="octetstring" type="Templates:OctetstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="charstring" type="Templates:CharstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="universal_charstring"
                        type="Templates:UniversalCharstringTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="record" type="Templates:RecordTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="record_of" type="Templates:RecordOfTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="set" type="Templates:SetTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="set_of" type="Templates:SetOfTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="union" type="Templates:UnionTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="anytype" type="Templates:AnytypeTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="address" type="Templates:AddressTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element name="omit" type="Templates:omit"/>
        <xsd:element name="any" type="Templates:any"/>
        <xsd:element name="anyoromit" type="Templates:anyoromit"/>
    <xsd:element name="templateDef" type="SimpleTypes:TString"/>
</xsd:choice>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SetTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:SetValue">
            <xsd:sequence>
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                    <xsd:element name="float" type="Templates:FloatTemplate"/>
                    <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                    <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                    <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                    <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                    <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                    <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                    <xsd:element name="universal_charstring"
                        type="Templates:UniversalCharstringTemplate"/>
                    <xsd:element name="record" type="Templates:RecordTemplate"/>
                    <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
                    <xsd:element name="set" type="Templates:SetTemplate"/>
                    <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
                    <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
                    <xsd:element name="union" type="Templates:UnionTemplate"/>
                    <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
                    <xsd:element name="address" type="Templates:AddressTemplate"/>
                    <xsd:element name="omit" type="Templates:omit"/>
                    <xsd:element name="any" type="Templates:any"/>
                    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                    <xsd:element name="templateDef" type="SimpleTypes:TString"/>
                </xsd:choice>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SetOfTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:SetOfValue">
            <xsd:choice>
                <xsd:sequence>
                    <xsd:element name="integer" type="Templates:IntegerTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="float" type="Templates:FloatTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="boolean" type="Templates:BooleanTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="objid" type="Templates:ObjidTemplate" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="bitstring" type="Templates:BitstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="hexstring" type="Templates:HexstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="octetstring" type="Templates:OctetstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="charstring" type="Templates:CharstringTemplate"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:sequence>
                    <xsd:element name="universal_charstring"
                        type="Templates:UniversalCharstringTemplate" minOccurs="0"
                </xsd:sequence>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
        <xsd:element name="record" type="Templates:RecordTemplate" minOccurs="0"
            maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
        <xsd:element name="record_of" type="Templates:RecordOfTemplate"
            minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
        <xsd:element name="set" type="Templates:SetTemplate" minOccurs="0"
            maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
        <xsd:element name="set_of" type="Templates:SetOfTemplate"
            minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
        <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"
            minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
        <xsd:element name="union" type="Templates:UnionTemplate" minOccurs="0"
            maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
        <xsd:element name="anytype" type="Templates:AnytypeTemplate" minOccurs="0"
            maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:sequence>
        <xsd:element name="address" type="Templates:AddressTemplate" minOccurs="0"
            maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:element name="omit" type="Templates:omit"/>
    <xsd:element name="any" type="Templates:any"/>
    <xsd:element name="anyoromit" type="Templates:anyoromit"/>
    <xsd:element name="templateDef" type="SimpleTypes:TString"/>
</xsd:choice>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="EnumeratedTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:EnumeratedValue">
            <xsd:choice>
                <xsd:element name="omit" type="Templates:omit"/>
                <xsd:element name="any" type="Templates:any"/>
                <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="UnionTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:UnionValue">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                <xsd:element name="float" type="Templates:FloatTemplate"/>
                <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                <xsd:element name="universal_charstring"
                    type="Templates:UniversalCharstringTemplate"/>
                <xsd:element name="record" type="Templates:RecordTemplate"/>
                <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
                <xsd:element name="set" type="Templates:SetTemplate"/>
                <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
                <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
                <xsd:element name="union" type="Templates:UnionTemplate"/>
                <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
                <xsd:element name="address" type="Templates:AddressTemplate"/>
                <xsd:element name="omit" type="Templates:omit"/>
                <xsd:element name="any" type="Templates:any"/>
                <xsd:element name="anyoromit" type="Templates:anyoromit"/>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element name="templateDef" type="SimpleTypes:TString"/>
    </xsd:choice>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="AnytypeTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:AnytypeValue">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                <xsd:element name="float" type="Templates:FloatTemplate"/>
                <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                <xsd:element name="universal_charstring"
                    type="Templates:UniversalCharstringTemplate"/>
                <xsd:element name="record" type="Templates:RecordTemplate"/>
                <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
                <xsd:element name="set" type="Templates:SetTemplate"/>
                <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
                <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
                <xsd:element name="union" type="Templates:UnionTemplate"/>
                <xsd:element name="address" type="Templates:AddressTemplate"/>
                <xsd:element name="omit" type="Templates:omit"/>
                <xsd:element name="any" type="Templates:any"/>
                <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="AddressTemplate">
    <xsd:complexContent>
        <xsd:extension base="Values:AnytypeValue">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="integer" type="Templates:IntegerTemplate"/>
                <xsd:element name="float" type="Templates:FloatTemplate"/>
                <xsd:element name="boolean" type="Templates:BooleanTemplate"/>
                <xsd:element name="objid" type="Templates:ObjidTemplate"/>
                <xsd:element name="bitstring" type="Templates:BitstringTemplate"/>
                <xsd:element name="hexstring" type="Templates:HexstringTemplate"/>
                <xsd:element name="octetstring" type="Templates:OctetstringTemplate"/>
                <xsd:element name="charstring" type="Templates:CharstringTemplate"/>
                <xsd:element name="universal_charstring"
                    type="Templates:UniversalCharstringTemplate"/>
                <xsd:element name="record" type="Templates:RecordTemplate"/>
                <xsd:element name="record_of" type="Templates:RecordOfTemplate"/>
                <xsd:element name="set" type="Templates:SetTemplate"/>
                <xsd:element name="set_of" type="Templates:SetOfTemplate"/>
                <xsd:element name="enumerated" type="Templates:EnumeratedTemplate"/>
                <xsd:element name="union" type="Templates:UnionTemplate"/>
                <xsd:element name="anytype" type="Templates:AnytypeTemplate"/>
                <xsd:element name="omit" type="Templates:omit"/>
                <xsd:element name="any" type="Templates:any"/>
                <xsd:element name="anyoromit" type="Templates:anyoromit"/>
                <xsd:element name="templateDef" type="SimpleTypes:TString"/>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

B.5 Схема для событий на языке XML интерфейса TCI-TL

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Events"
    xmlns:Events="http://uri.etsi.org/ttcn-3/3.0.0/tci/Events"
    xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
    xmlns:Templates="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates"
    xmlns:SimpleTypes="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes"
    xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values" elementFormDefault="qualified">

    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/SimpleTypes.xsd"
        schemaLocation="SimpleTypes.xsd"/>

```

```

<xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types.xsd"
schemaLocation="Types.xsd"/>
<xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values.xsd"
schemaLocation="Values.xsd"/>
<xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Templates.xsd"
schemaLocation="Templates.xsd"/>

<!-- общее определение для всех событий -->
<xsd:complexType name="Event" mixed="true">
  <xsd:sequence>
    <xsd:element name="am" type="SimpleTypes:TString"/>
  </xsd:sequence>
  <xsd:attribute name="ts" type="xsd:time" use="required"/>
  <xsd:attribute name="src" type="SimpleTypes:TString" use="optional"/>
  <xsd:attribute name="line" type="SimpleTypes:TInteger" use="optional"/>

  <!-- основная структура идентификатора для тестовых компонентов, портов и таймера -->
  <xsd:attribute name="name" type="SimpleTypes:TString" use="required"/>
  <xsd:attribute name="id" type="SimpleTypes:TInteger" use="required"/>
  <xsd:attribute name="type" type="SimpleTypes:TString" use="required"/>
</xsd:complexType>

<!-- это событие расширяется за счет всех событий конфигурации порта -->
<xsd:complexType name="PortConfiguration">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port1" type="Types:TriPortIdType" minOccurs="1"
maxOccurs="1"/>
        <xsd:element name="port2" type="Types:TriPortIdType" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!-- это событие расширяется за счет всех событий состояния порта -->
<xsd:complexType name="PortStatus">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!-- тестовые примеры -->
<xsd:complexType name="tliTcExecute">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="tcId" type="Types:TciTestCaseIdType"/>
        <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
        <xsd:element name="dur" type="Types:TriTimerDurationType" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTcStart">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="tcId" type="Types:TciTestCaseIdType"/>
        <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
        <xsd:element name="dur" type="Types:TriTimerDurationType" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTcStop">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event"/>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTcStarted">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">

```

```

        <xsd:sequence>
            <xsd:element name="tcId" type="Types:TciTestCaseIdType"/>
            <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
            <xsd:element name="dur" type="Types:TriTimerDurationType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTcTerminated">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="tcId" type="Types:TciTestCaseIdType"/>
                <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
                <xsd:element name="outcome" type="Values:VerdictValue"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<!-- управление -->
<xsd:complexType name="tliCtrlStart">
    <xsd:complexContent>
        <xsd:extension base="Events:Event"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCtrlStop">
    <xsd:complexContent>
        <xsd:extension base="Events:Event"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCtrlTerminated">
    <xsd:complexContent>
        <xsd:extension base="Events:Event"/>
    </xsd:complexContent>
</xsd:complexType>

<!-- асинхронная связь -->
<xsd:complexType name="tliMSend_m">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="msgValue" type="Values:Value"/>
                <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                <xsd:choice>
                    <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                    <xsd:sequence>
                        <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>
                        <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                    </xsd:sequence>
                </xsd:choice>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliMSend_m_BC">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="msgValue" type="Values:Value"/>
                <xsd:choice>
                    <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                    <xsd:sequence>
                        <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>
                        <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                    </xsd:sequence>
                </xsd:choice>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliMSend_m_MC">

```

```

    <xsd:complexContent mixed="true">
      <xsd:extension base="Events:Event">
        <xsd:sequence>
          <xsd:element name="port" type="Types:TriPortIdType"/>
          <xsd:element name="msgValue" type="Values:Value"/>
          <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>
          <xsd:choice>
            <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
            <xsd:sequence>
              <xsd:element name="msg" type="Types:TriMessageType" minOccurs="0"/>
              <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
            </xsd:sequence>
          </xsd:choice>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="tliMSend_c">
    <xsd:complexContent mixed="true">
      <xsd:extension base="Events:Event">
        <xsd:sequence>
          <xsd:element name="port" type="Types:TriPortIdType"/>
          <xsd:element name="msgValue" type="Values:Value"/>
          <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/>
          <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="tliMSend_c_BC">
    <xsd:complexContent mixed="true">
      <xsd:extension base="Events:Event">
        <xsd:sequence>
          <xsd:element name="port" type="Types:TriPortIdType"/>
          <xsd:element name="msgValue" type="Values:Value"/>
          <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="tliMSend_c_MC">
    <xsd:complexContent mixed="true">
      <xsd:extension base="Events:Event">
        <xsd:sequence>
          <xsd:element name="port" type="Types:TriPortIdType"/>
          <xsd:element name="msgValue" type="Values:Value"/>
          <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/>
          <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="tliMDetected_m">
    <xsd:complexContent mixed="true">
      <xsd:extension base="Events:Event">
        <xsd:sequence>
          <xsd:element name="port" type="Types:TriPortIdType"/>
          <xsd:element name="msgValue" type="Types:TriMessageType"/>
          <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="tliMDetected_c">
    <xsd:complexContent mixed="true">
      <xsd:extension base="Events:Event">
        <xsd:sequence>
          <xsd:element name="port" type="Types:TriPortIdType"/>
          <xsd:element name="msgValue" type="Types:TriMessageType"/>
          <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

```

    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliMMismatch_m">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="msgValue" type="Values:Value"/>
        <xsd:element name="msgTpl" type="Templates:TciValueTemplate"/>
        <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
        <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
        <xsd:element name="addressTpl" type="Templates:TciValueTemplate"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliMMismatch_c">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="msgValue" type="Values:Value"/>
        <xsd:element name="msgTpl" type="Templates:TciValueTemplate"/>
        <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
        <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
        <xsd:element name="fromTpl" type="Templates:TciNonValueTemplate"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliMReceive_m">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>
        <xsd:element name="msgTpl" type="Templates:TciValueTemplate"
minOccurs="0"/>
        <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
        <xsd:element name="addressTpl" type="Templates:TciValueTemplate"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliMReceive_c">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="msgValue" type="Values:Value" minOccurs="0"/>
        <xsd:element name="msgTpl" type="Templates:TciValueTemplate"
minOccurs="0"/>
        <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
        <xsd:element name="fromTpl" type="Templates:TciNonValueTemplate"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!-- синхронная связь -->
<xsd:complexType name="tliPrCall_m">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TriParameterListType"
minOccurs="0"/>
        <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
        <xsd:choice>
          <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"
minOccurs="0"/>

```

```

                <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

    <xsd:complexType name="tliPrCall_m_BC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"
minOccurs="0"/>
                </xsd:sequence>
                <xsd:choice>
                    <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"
minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:choice>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

    <xsd:complexType name="tliPrCall_m_MC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TriParameterListType"
minOccurs="0"/>
                </xsd:sequence>
                <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>
                <xsd:choice>
                    <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"
minOccurs="0"/>
                    <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
                </xsd:choice>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

    <xsd:complexType name="tliPrCall_c">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"
minOccurs="0"/>
                </xsd:sequence>
                <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/>
                <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

    <xsd:complexType name="tliPrCall_c_BC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">
                <xsd:sequence>
                    <xsd:element name="port" type="Types:TriPortIdType"/>
                    <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                    <xsd:element name="parsValue" type="Types:TciParameterListType"
minOccurs="0"/>
                </xsd:sequence>
                <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

    <xsd:complexType name="tliPrCall_c_MC">
        <xsd:complexContent mixed="true">
            <xsd:extension base="Events:Event">

```

```

        <xsd:sequence>
            <xsd:element name="port" type="Types:TriPortIdType"/>
            <xsd:element name="signature" type="Types:TriSignatureIdType"/>
            <xsd:element name="parsValue" type="Types:TciParameterListType"/>
minOccurs="0"/>
            <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/>
            <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"/>
minOccurs="0"/>
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetcallDetected_m">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="pars" type="Types:TriParameterListType"/>
                <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetcallDetected_c">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="pars" type="Types:TciParameterListType"/>
                <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetcallMismatch_m">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                <xsd:element name="pars" type="Types:TriParameterListType"/>
                <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetcallMismatch_c">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                <xsd:element name="pars" type="Types:TciParameterListType"/>
                <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetcall_m">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="from" type="Types:TriAddressType" minOccurs="0"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                <xsd:element name="pars" type="Types:TriParameterListType"/>
                <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetcall_c">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="from" type="Types:TriAddressType" minOccurs="0"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
        <xsd:element name="pars" type="Types:TciParameterListType"/>
        <xsd:element name="parsTmpl" type="Templates:TciValueTemplate"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrReply_m">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TriParameterListType"/>
        <xsd:element name="replValue" type="Values:Value"/>
        <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
        <xsd:choice>
          <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
          <xsd:sequence>
            <xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>
            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
          </xsd:sequence>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrReply_m_BC">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TriParameterListType"/>
        <xsd:element name="replValue" type="Values:Value"/>
        <xsd:choice>
          <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
          <xsd:sequence>
            <xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>
            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
          </xsd:sequence>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrReply_m_MC">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TriParameterListType"/>
        <xsd:element name="replValue" type="Values:Value"/>
        <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>
        <xsd:choice>
          <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
          <xsd:sequence>
            <xsd:element name="repl" type="Types:TriParameterType" minOccurs="0"/>
            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
          </xsd:sequence>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

</xsd:complexType>

<xsd:complexType name="tliPrReply_c">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TciParameterListType"/>
        <xsd:element name="replValue" type="Values:Value"/>
        <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/>
        <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrReply_c_BC">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TciParameterListType"/>
        <xsd:element name="replValue" type="Values:Value"/>
        <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrReply_c_MC">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TciParameterListType"/>
        <xsd:element name="replValue" type="Values:Value"/>
        <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/>
        <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetReplyDetected_m">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="replValue" type="Values:Value"/>
        <xsd:element name="address" type="Types:TriAddressType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetReplyDetected_c">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="replValue" type="Values:Value"/>
        <xsd:element name="from" type="Types:TriComponentIdType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetReplyMismatch_m">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>

```

```

        <xsd:element name="parsValue" type="Types:TriParameterListType"/>
        <xsd:element name="replValue" type="Values:Value"/>
        <xsd:element name="replTmpl" type="Values:Value"/>
        <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
        <xsd:element name="address" type="Types:TriAddressType"/>
        <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetReplyMismatch_c">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                <xsd:element name="replValue" type="Values:Value"/>
                <xsd:element name="replTmpl" type="Values:Value"/>
                <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
                <xsd:element name="from" type="Types:TriComponentIdType"/>
                <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetReply_m">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                <xsd:element name="replValue" type="Values:Value"/>
                <xsd:element name="replTmpl" type="Values:Value"/>
                <xsd:element name="address" type="Types:TriAddressType"/>
                <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrGetReply_c">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                <xsd:element name="replValue" type="Values:Value"/>
                <xsd:element name="replTmpl" type="Values:Value"/>
                <xsd:element name="from" type="Types:TriComponentIdType"/>
                <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrRaise_m">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="parsValue" type="Types:TriParameterListType"/>
                <xsd:element name="excValue" type="Values:Value"/>
                <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
                <xsd:choice>
                    <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                    <xsd:sequence>
                        <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>
                        <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
                    </xsd:sequence>
                </xsd:choice>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

<xsd:complexType name="tliPrRaise_m BC">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TriParameterListType"/>
        <xsd:element name="excValue" type="Values:Value"/>
        <xsd:choice>
          <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
          <xsd:sequence>
            <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>
            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
          </xsd:sequence>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrRaise_m MC">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TriParameterListType"/>
        <xsd:element name="excValue" type="Values:Value"/>
        <xsd:element name="addresses" type="Types:TriAddressListType" minOccurs="0"/>
        <xsd:choice>
          <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
          <xsd:sequence>
            <xsd:element name="exc" type="Types:TriExceptionType" minOccurs="0"/>
            <xsd:element name="transmission-failure"
type="SimpleTypes:TriStatusType" minOccurs="0"/>
          </xsd:sequence>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrRaise_c">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TciParameterListType"/>
        <xsd:element name="excValue" type="Values:Value"/>
        <xsd:element name="to" type="Types:TriComponentIdType" minOccurs="0"/>
        <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrRaise_c BC">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TciParameterListType"/>
        <xsd:element name="excValue" type="Values:Value"/>
        <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrRaise_c MC">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>

```

```

        <xsd:element name="parsValue" type="Types:TciParameterListType"/>
        <xsd:element name="excValue" type="Values:Value"/>
        <xsd:element name="toList" type="Types:TriComponentIdListType" minOccurs="0"/>
        <xsd:element name="transmission-failure" type="SimpleTypes:TriStatusType"
minOccurs="0"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatchDetected_m">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="exc" type="Types:TriExceptionType"/>
                <xsd:element name="address" type="Types:TriAddressType" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatchDetected_c">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="exc" type="Types:TriExceptionType"/>
                <xsd:element name="from" type="Types:TriComponentIdType" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatchMismatch_m">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                <xsd:element name="exc" type="Values:Value"/>
                <xsd:element name="excTmpl" type="Templates:TciValueTemplate"/>
                <xsd:element name="diffs" type="Templates:TciValueDifferenceList"/>
                <xsd:element name="address" type="Types:TriAddressType"/>
                <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatchMismatch_c">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="parsValue" type="Types:TciParameterListType"/>
                <xsd:element name="exc" type="Values:Value"/>
                <xsd:element name="excTmpl" type="Templates:TciValueTemplate"/>
                <xsd:element name="address" type="Types:TriAddressType"/>
                <xsd:element name="addressTmpl" type="Templates:TciValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatch_m">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="port" type="Types:TriPortIdType"/>
                <xsd:element name="signature" type="Types:TriSignatureIdType"/>
                <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
                <xsd:element name="exception" type="Values:Value"/>
                <xsd:element name="exceptionTmpl" type="Templates:TciValueTemplate"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

</xsd:complexType>

<xsd:complexType name="tliPrCatch_c">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="signatureTmpl" type="Templates:TciValueTemplate"/>
        <xsd:element name="exception" type="Values:Value"/>
        <xsd:element name="exceptionTmpl" type="Templates:TciValueTemplate"/>
        <xsd:element name="from" type="Types:TriComponentIdType"/>
        <xsd:element name="fromTmpl" type="Templates:TciNonValueTemplate"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatchTimeoutDetected">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPrCatchTimeout">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="port" type="Types:TriPortIdType"/>
        <xsd:element name="signature" type="Types:TriSignatureIdType"/>
        <xsd:element name="parsValue" type="Types:TriParameterListType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!-- КОМПОНЕНТЫ -->
<xsd:complexType name="tliCCreate">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="comp" type="Types:TriComponentIdType"/>
        <xsd:element name="name" type="SimpleTypes:TString"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCStart">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="comp" type="Types:TriComponentIdType"/>
        <xsd:element name="name" type="Types:TciBehaviourIdType"/>
        <xsd:element name="pars" type="Types:TciParameterListType" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCRunning">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>
        <xsd:element name="comp" type="Types:TriComponentIdType"/>
        <xsd:element name="status" type="SimpleTypes:TBoolean"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliCAlive">
  <xsd:complexContent mixed="true">
    <xsd:extension base="Events:Event">
      <xsd:sequence>

```

```

        <xsd:element name="comp" type="Types:TriComponentIdType"/>
        <xsd:element name="status" type="SimpleTypes:TBoolean"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tlicStop">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tlicKill">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tlicDoneMismatch">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="compTpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
            <xsd:attribute name="done" type="SimpleTypes:TBoolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tlicKilledMismatch">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="compTpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
            <xsd:attribute name="done" type="SimpleTypes:TBoolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tlicDone">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="compTpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
            <xsd:attribute name="done" type="SimpleTypes:TBoolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tlicKilled">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="compTpl" type="Templates:TciNonValueTemplate"/>
            </xsd:sequence>
            <xsd:attribute name="done" type="SimpleTypes:TBoolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tlicTerminated">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="comp" type="Types:TriComponentIdType"/>
                <xsd:element name="verdict" type="Values:VerdictValue" maxOccurs="1"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<!-- порты -->
<xsd:complexType name="tliPConnect">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPDisconnect">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPMap">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPUnmap">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPClear">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPStart">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPStop">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliPHalt">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:PortConfiguration"/>
    </xsd:complexContent>
</xsd:complexType>

<!-- кодек -->
<xsd:complexType name="tliEncode">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="val" type="Values:Value"/>
                <xsd:choice>
                    <xsd:element name="msg" type="Types:TriMessageType"/>
                    <xsd:element name="encoder-failure" type="SimpleTypes:TciStatusType"/>
                </xsd:choice>
            </xsd:sequence>
            <xsd:attribute name="codec" type="SimpleTypes:TString" use="optional"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliDecode" mixed="true">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:choice>
                    <xsd:element name="val" type="Values:Value"/>
                    <xsd:element name="decoder-failure" type="SimpleTypes:TciStatusType"/>
                </xsd:choice>
                <xsd:element name="msg" type="Types:TriMessageType"/>
            </xsd:sequence>
            <xsd:attribute name="codec" type="SimpleTypes:TString" use="optional"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<!-- таймеры -->
<xsd:complexType name="tliTTimeoutDetected">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1"
minOccurs="1"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTTimeoutMismatch">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1"
minOccurs="1"/>
                <xsd:element name="timerTpl" type="Templates:TciNonValueTemplate"
maxOccurs="1" minOccurs="1"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTTimeout">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="timer" type="Types:TriTimerIdType" maxOccurs="1"
minOccurs="1"/>
                <xsd:element name="timerTpl" type="Templates:TciNonValueTemplate"
maxOccurs="1" minOccurs="1"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTStart">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="timer" type="Types:TriTimerIdType"/>
                <xsd:element name="dur" type="Types:TriTimerDurationType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTStop">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="timer" type="Types:TriTimerIdType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTRead">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="timer" type="Types:TriTimerIdType"/>
                <xsd:element name="elapsed" type="Types:TriTimerDurationType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliTRunning">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="timer" type="Types:TriTimerIdType"/>
            </xsd:sequence>
            <xsd:attribute name="status" type="SimpleTypes:TBoolean"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<!-- КОНТЕКСТ -->
<xsd:complexType name="tliSEnter">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
                <xsd:element name="kind" type="SimpleTypes:TString"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliSLeave">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                <xsd:element name="return" type="Values:Value" minOccurs="0"/>
                <xsd:element name="kind" type="SimpleTypes:TString"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<!-- переменные и параметр модуля -->
<xsd:complexType name="tliVar">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                <xsd:element name="val" type="Values:Value" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliModuleParr">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                <xsd:element name="val" type="Values:Value" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<!-- вердикты -->
<xsd:complexType name="tliGetVerdict">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="verdict" type="Values:VerdictValue"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliSetVerdict">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="verdict" type="Values:VerdictValue"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<!-- регистрация -->
<xsd:complexType name="tliLog">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">

```

```

        <xsd:sequence>
            <xsd:element name="log" type="SimpleTypes:TString"/>
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<!-- alt -->
<xsd:complexType name="tliAEnter">
    <xsd:complexContent>
        <xsd:extension base="Events:Event"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliALeave">
    <xsd:complexContent>
        <xsd:extension base="Events:Event"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliADefaults">
    <xsd:complexContent>
        <xsd:extension base="Events:Event"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliAActivate">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="name" type="Types:QualifiedName" minOccurs="1"
maxOccurs="1"/>
                <xsd:element name="pars" type="Types:TriParameterListType" minOccurs="0"/>
                <xsd:element name="ref" type="Values:Value"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliADeactivate">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event">
            <xsd:sequence>
                <xsd:element name="ref" type="Values:Value"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliANomatch">
    <xsd:complexContent mixed="true">
        <xsd:extension base="Events:Event"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliARepeat">
    <xsd:complexContent>
        <xsd:extension base="Events:Event"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="tliAWait">
    <xsd:complexContent>
        <xsd:extension base="Events:Event"/>
    </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

В.6 Схема для регистрации на языке XML интерфейса TCI-TL

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/TLI"
xmlns:TLI="http://uri.etsi.org/ttcn-3/3.0.0/tci/TLI"
xmlns:Types="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types"
xmlns:Values="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values"
xmlns:Events="http://uri.etsi.org/ttcn-3/3.0.0/tci/Events" elementFormDefault="qualified">

    <xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Types.xsd"
schemaLocation="Types.xsd"/>

```

```

<xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Values.xsd"
  schemaLocation="Values.xsd"/>
<xsd:import namespace="http://uri.etsi.org/ttcn-3/3.0.0/tci/Events.xsd"
  schemaLocation="Events.xsd"/>

<xsd:element name="logfile" type="TLI:LogModule"/>
<xsd:complexType name="LogModule">
  <xsd:sequence>
    <xsd:element name="header" type="TLI:Header"/>
    <xsd:element name="body" type="TLI:Body"/>
    <xsd:element name="trailer" type="TLI:Trailer"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Header">
  <xsd:sequence>
    <!-- версия регистрации -->
    <xsd:element name="version" type="xsd:string"/>
    <!-- начало регистрации -->
    <xsd:element name="ts" type="xsd:time"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Trailer">
  <xsd:sequence/>
</xsd:complexType>
<xsd:complexType name="Body">
  <xsd:choice maxOccurs="unbounded">

    <!-- операции с тестовыми примерами -->
    <xsd:element name="tliTcExecute" type="Events:tliTcExecute"/>
    <xsd:element name="tliTcStart" type="Events:tliTcStart"/>
    <xsd:element name="tliTcStop" type="Events:tliTcStop"/>
    <xsd:element name="tliTcStarted" type="Events:tliTcStarted"/>
    <xsd:element name="tliTcTerminated" type="Events:tliTcTerminated"/>

    <!-- операции управления -->
    <xsd:element name="tliCtrlStart" type="Events:tliCtrlStart"/>
    <xsd:element name="tliCtrlStop" type="Events:tliCtrlStop"/>
    <xsd:element name="tliCtrlTerminated" type="Events:tliCtrlTerminated"/>

    <!-- асинхронная связь -->
    <xsd:element name="tliMSend_m" type="Events:tliMSend_m"/>
    <xsd:element name="tliMSend_c" type="Events:tliMSend_c"/>
    <xsd:element name="tliMDetected_m" type="Events:tliMDetected_m"/>
    <xsd:element name="tliMDetected_c" type="Events:tliMDetected_c"/>
    <xsd:element name="tliMMismatch_m" type="Events:tliMMismatch_m"/>
    <xsd:element name="tliMMismatch_c" type="Events:tliMMismatch_c"/>
    <xsd:element name="tliMReceive_m" type="Events:tliMReceive_m"/>
    <xsd:element name="tliMReceive_c" type="Events:tliMReceive_c"/>

    <!-- синхронная связь -->
    <xsd:element name="tliPrCall_m" type="Events:tliPrCall_m"/>
    <xsd:element name="tliPrCall_c" type="Events:tliPrCall_c"/>

    <xsd:element name="tliPrGetcallDetected_m" type="Events:tliPrGetcallDetected_m"/>
    <xsd:element name="tliPrGetcallDetected_c" type="Events:tliPrGetcallDetected_c"/>
    <xsd:element name="tliPrGetcallMismatch_m" type="Events:tliPrGetcallMismatch_m"/>
    <xsd:element name="tliPrGetcallMismatch_c" type="Events:tliPrGetcallMismatch_c"/>
    <xsd:element name="tliPrGetcall_m" type="Events:tliPrGetcall_m"/>
    <xsd:element name="tliPrGetcall_c" type="Events:tliPrGetcall_c"/>

    <xsd:element name="tliPrReply_m" type="Events:tliPrReply_m"/>
    <xsd:element name="tliPrReply_c" type="Events:tliPrReply_c"/>

    <xsd:element name="tliPrGetReplyDetected_m" type="Events:tliPrGetReplyDetected_m"/>
    <xsd:element name="tliPrGetReplyDetected_c" type="Events:tliPrGetReplyDetected_c"/>

    <xsd:element name="tliPrGetReplyMismatch_m" type="Events:tliPrGetReplyMismatch_m"/>
    <xsd:element name="tliPrGetReplyMismatch_c" type="Events:tliPrGetReplyMismatch_c"/>
    <xsd:element name="tliPrGetReply_m" type="Events:tliPrGetReply_m"/>
    <xsd:element name="tliPrGetReply_c" type="Events:tliPrGetReply_c"/>

    <xsd:element name="tliPrRaise_m" type="Events:tliPrRaise_m"/>
    <xsd:element name="tliPrRaise_c" type="Events:tliPrRaise_c"/>

    <xsd:element name="tliPrCatchDetected_m" type="Events:tliPrCatchDetected_m"/>
    <xsd:element name="tliPrCatchDetected_c" type="Events:tliPrCatchDetected_c"/>
    <xsd:element name="tliPrCatchMismatch_m" type="Events:tliPrCatchMismatch_m"/>
    <xsd:element name="tliPrCatchMismatch_c" type="Events:tliPrCatchMismatch_c"/>
    <xsd:element name="tliPrCatch_m" type="Events:tliPrCatch_m"/>
    <xsd:element name="tliPrCatch_c" type="Events:tliPrCatch_c"/>
  </xsd:choice>
</xsd:complexType>

```

```

<xsd:element name="tliPrCatchTimeout" type="Events:tliPrCatchTimeout"/>

<!-- КОМПОНЕНТЫ -->
<xsd:element name="tliCCreate" type="Events:tliCCreate"/>
<xsd:element name="tliCStart" type="Events:tliCStart"/>
<xsd:element name="tliCRunning" type="Events:tliCRunning"/>
<xsd:element name="tliCAlive" type="Events:tliCRunning"/>
<xsd:element name="tliCStop" type="Events:tliCStop"/>
<xsd:element name="tliCKill" type="Events:tliCStop"/>
<xsd:element name="tliCDoneMismatch" type="Events:tliCDone"/>
<xsd:element name="tliCDone" type="Events:tliCDone"/>
<xsd:element name="tliCKilledMismatch" type="Events:tliCDone"/>
<xsd:element name="tliCKilled" type="Events:tliCDone"/>
<xsd:element name="tliCTerminated" type="Events:tliCTerminated"/>

<!-- ПОРТЫ -->
<xsd:element name="tliPConnect" type="Events:tliPConnect"/>
<xsd:element name="tliPDisconnect" type="Events:tliPDisconnect"/>
<xsd:element name="tliPMap" type="Events:tliPMap"/>
<xsd:element name="tliPUnmap" type="Events:tliPUnmap"/>
<xsd:element name="tliPClear" type="Events:tliPClear"/>
<xsd:element name="tliPStart" type="Events:tliPStart"/>
<xsd:element name="tliPStop" type="Events:tliPStop"/>
<xsd:element name="tliPHalt" type="Events:tliPStop"/>

<!-- КОДЕК -->
<xsd:element name="tliDecode" type="Events:tliDecode"/>
<xsd:element name="tliEncode" type="Events:tliEncode"/>

<!-- ТАЙМЕРЫ -->
<xsd:element name="tliTTimeoutDetected" type="Events:tliTTimeoutDetected"/>
<xsd:element name="tliTTimeoutMismatch" type="Events:tliTTimeoutMismatch"/>
<xsd:element name="tliTTimeout" type="Events:tliTTimeout"/>
<xsd:element name="tliTStart" type="Events:tliTStart"/>
<xsd:element name="tliTStop" type="Events:tliTStop"/>
<xsd:element name="tliTRead" type="Events:tliTRead"/>
<xsd:element name="tliTRunning" type="Events:tliTRunning"/>

<!-- КОНТЕКСТЫ -->
<xsd:element name="tliSEnter" type="Events:tliSEnter"/>
<xsd:element name="tliSLeave" type="Events:tliSLeave"/>

<!-- ОПЕРАТОРЫ -->
<xsd:element name="tliVar" type="Events:tliVar"/>
<xsd:element name="tliGetVerdict" type="Events:tliGetVerdict"/>
<xsd:element name="tliSetVerdict" type="Events:tliSetVerdict"/>
<xsd:element name="tliLog" type="Events:tliLog"/>

<!-- alt -->
<xsd:element name="tliAEnter" type="Events:tliAEnter"/>
<xsd:element name="tliALeave" type="Events:tliALeave"/>
<xsd:element name="tliADefaults" type="Events:tliADefaults"/>
<xsd:element name="tliAActivate" type="Events:tliAActivate"/>
<xsd:element name="tliADeactivate" type="Events:tliADeactivate"/>
<xsd:element name="tliANomatch" type="Events:tliANomatch"/>
<xsd:element name="tliARepeat" type="Events:tliARepeat"/>
<xsd:element name="tliAwait" type="Events:tliAwait"/>

</xsd:choice>
</xsd:complexType>
</xsd:schema>

```

БИБЛИОГРАФИЯ

- INTOOL CGI/NPL038 (V2.2): *Generic Compiler/Interpreter interface; GCI Interface Specification Infrastructural Tools for Informational Technology and Telecommunications Conference Testing*, December 1996.
- Рекомендация МСЭ-Т Х.292 (2002), *Методология аттестационного тестирования ВОС и структура Рекомендаций о протоколах для применений МСЭ-Т – Древовидно-табличная комбинированная нотация (TTCN)*.
ISO/IEC 9646-3:1998, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 3: The Tree and Tabular combined Notation (TTCN)*.
- ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*.
- OMG CORBA v2.2: *The Common Object Request Broker: Architecture and Specification*, Section 3, February 1998.

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

| | |
|----------------|---|
| Серия А | Организация работы МСЭ-Т |
| Серия D | Общие принципы тарификации |
| Серия E | Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы |
| Серия F | Нетелефонные службы электросвязи |
| Серия G | Системы и среда передачи, цифровые системы и сети |
| Серия H | Аудиовизуальные и мультимедийные системы |
| Серия I | Цифровая сеть с интеграцией служб |
| Серия J | Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов |
| Серия K | Защита от помех |
| Серия L | Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений |
| Серия M | Управление электросвязью, включая СУЭ и техническое обслуживание сетей |
| Серия N | Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ |
| Серия O | Требования к измерительной аппаратуре |
| Серия P | Качество телефонной передачи, телефонные установки, сети местных линий |
| Серия Q | Коммутация и сигнализация |
| Серия R | Телеграфная передача |
| Серия S | Оконечное оборудование для телеграфных служб |
| Серия T | Оконечное оборудование для телематических служб |
| Серия U | Телеграфная коммутация |
| Серия V | Передача данных по телефонной сети |
| Серия X | Сети передачи данных, взаимосвязь открытых систем и безопасность |
| Серия Y | Глобальная информационная инфраструктура, аспекты протокола Интернет и сети последующих поколений |
| Серия Z | Языки и общие аспекты программного обеспечения для систем электросвязи |