

Стандартизированные языки спецификаций и описаний для современных телекоммуникационных архитектур

Б.С. Гольдштейн

За последние двадцать лет мировой рынок телекоммуникаций стал одним из наиболее быстро развивающихся на планете. Его ежегодный оборот оценивается примерно в сто млрд. долларов. С интенсивным ростом в мире услуг мобильной связи, с продолжающейся интеграцией речевой связи, передачи данных и видеосвязи кажутся безграничными возможности развития для операторов сети, для поставщиков услуг и для производителей телекоммуникационного оборудования.

Тем не менее имеются определенные трудности в реализации этих возможностей. Далеки от стабильности темпы развития индустрии телекоммуникаций. Далеки от совершенства удобства организации новых видов обслуживания и все более дорогими становятся изменения в инфраструктуре телекоммуникационных систем. Как город, переполненный лавинообразно возросшим автотранспортом, рынок телекоммуникаций наряду с механическим ростом нуждается в средствах увеличения гибкости и ликвидации заторов.

Для того, чтобы телекоммуникационная индустрия продолжала развиваться так же успешно, необходимо осознать и скоординировать задачи и потребности каждой из следующих групп: операторов телекоммуникационных сетей; производителей телекоммуникационного и информационного оборудования; поставщиков телекоммуникационных и информационных услуг и конечных пользователей информации - абонентов.

Теоретически этим группам следовало бы свободно и постоянно взаимодействовать друг с другом, чтобы обеспечивать необходимое выполнение функций и услуг по мере роста потребностей. Но реально эти сферы имеют множество различных систем управления, систем контроля, протоколов и концепций коммутации, все они разработаны для различных целей и используют различные технологии и международные стандарты. Интеллектуальные сети, ISDN, синхронные цифровые иерархии - все они являются частями рынка телекоммуникаций, но несмотря на это не составляют одного целого.

Как результат этого, около 75% работ по изменению услуг на сетях общего пользования имеют мало общего с реализацией самих услуг, а затрачиваются на то, чтобы заставить различные системы понимать друг друга и функционировать совместно в все усложняющейся и вечно меняющейся телекоммуникационной инфраструктуре. Как правило это занимает много лет и требует существенных капиталовложений.

Вот почему требуются решения, которые бы всецело основывались на новейших научных достижениях, но которые могли бы также работать с уже существующими системами и использовать уже существующие стандарты.

1. Постановка проблемы

Усилиями ведущих мировых научно-исследовательских институтов и университетов, входящих в международные организации в области телекоммуникаций (Международный союз электросвязи - МСЭ, в частности) разработаны и приняты ряд архитектур современной телекоммуникации: цифровая сеть с интегральным обслуживанием ISDN, широкополосная ISDN, интеллектуальная сеть IN, управление сетью связи TMN и др.

Назначение каждой такой архитектуры - служить основанием для разработки конкретных систем для конкретных нужд, а также стандартизовывать как процесс разработки, так и конечный продукт. Так например, архитектура TMN позволяет создавать новые функции технического обслуживания и управления сетью отдельно от самых элементов сети.

Для каждой такой архитектуры определяются основные архитектурные конструкции (элементы архитектуры) и стандартизованные средства взаимодействия и интерфейсы между ними.

Такие средства взаимодействия между элементами архитектуры телекоммуникационной сети в терминах протоколов систем сигнализации достаточно успешно разрабатывались и разрабатываются в Исследовательской комиссии 11 (ИК-11) Сектора стандартизации электросвязи МСЭ (ССЭ-МСЭ или ITU-T в английской аббревиатуре) на языке спецификаций и описаний SDL, разработанном Исследовательской комиссией 10 (ИК-10). Прекрасным примером такой разработки являются спецификации на SDL системы общеканальной межстанционной сигнализации ОКС-7.

Языковые средства этого недавнего периода до появления архитектур TMN и IN, когда вся логика обслуживания вызовов сосредотачивалась в коммутационных узлах и станциях, составляли язык спецификаций и описаний SDL (Рекомендации МСЭ серии Z.100), язык программирования CHILL (Рекомендации МСЭ серии Z.200) и язык общения "человек-машина" MML (Рекомендации МСЭ серии Z.300).

Современные телекоммуникационные архитектуры вызвали необходимость в дополнительных языках их спецификаций и описаний: ASN.1 (Abstract Syntax Notation One) для протоколов модели Взаимодействия открытых систем (ВОС или OSI в английской аббревиатуре), TTCN (Tree and Tabular Combined Notation) для создания тестовых сценариев при тестировании конформности в рамках телекоммуникационной архитектуры, GDMO для информационных моделей в рамках архитектуры TMN и др. Проблемы стандартизации, развития и совместного использования SDL и этих языков для спецификаций и описаний новых телекоммуникационных архитектур составляют предмет настоящей статьи.

Но прежде чем перейти к более глубокому рассмотрению этой чрезвычайно важной задачи стандартизации методов и средств разработки перспективных

телекоммуникационных систем, полезно определить, какая стандартизация в этой статье рассматриваться *не будет*.

А именно не будет рассматриваться используемая российскими НИИКБ система ГОСТов ЕСКД, традиционно сопровождавшая и сопровождающая все НИИОКР в областях телекоммуникации и вычислительной техники вплоть до присвоения большинству из них литеры О1 “посмертно” (последние достижения ЕС ЭВМ и СМ ЭВМ - примеры этому) и породившая целый ряд трудно вообразимых силлогизмов типа “калькодержатель” (насилие не только над языком, но и над здравым смыслом).

Не будет также рассматриваться и другая крайность, связанная с противопоставлением устаревшим стандартам отрицания любой методологии проектирования как таковой. Действительно сегодняшние конструкторы фирм MITEL, INTEL и т.д., позволяют собрать 16-тиномерную мини-АТС выпускнику ВУЗа связи, а язык C++ давно уже является стандартом де факто. Но распространение этого подхода на более серьезные проекты в области телекоммуникаций обречено на провал.

2. Анализ текущего состояния

Интенсивное взаимопроникновение информационных (компьютерных) и телекоммуникационных технологий (столь бурно развивающееся, что уже сегодня невозможно однозначно ответить на вопрос: является ли ИНТЕРНЕТ телекоммуникационной сетью общего пользования или нет?) существенно меняет сложившиеся представления о протоколах сигнализации все более и более преобразуя эти протоколы в чисто программные интерфейсы, строящиеся в терминах идеологии открытых распределенных процессов (ODP).

При этом используются различные промышленные стандарты де юре (ANSI C, например) и де факто (X/Open, например). Интересно отметить, что зарубежная телекоммуникационная промышленность традиционно ориентировалась на стандарты де юре, а зарубежная же компьютерная промышленность - на стандарты де факто. Единодушная техническая политика отечественных предприятий связи и вычислительной техники по этому вопросу уже упоминалась выше.

К необходимости единодушия (правда, не такого) приводит и наблюдающаяся тенденция к интеграции различных телекоммуникационных архитектур. Соответственно возрастает необходимость единообразия нотаций, описывающих различные архитектуры. Впрочем, уже сегодня ни один язык ни в одной архитектуре не используется изолированно. Так например, TTCN используется совместно с ANS.1, т.к. само тестирование конформности предполагает структуру PDU (Protocol Data Unit), написанную на ANS.1. По совместному использованию SDL и ANS.1 уже принята Рекомендация Z.105, а по MSC и SDL - Рекомендация Z.120 МСЭ.

Таким образом, для описаний современных телекоммуникационных архитектур в рамках МСЭ используются следующие языки: SDL, MSC, ASN.1, TTCN и GDMO. Этот перечень может быть дополнен языком IDL (Interface Definition Language),

разрабатываемым OMG (Object Management Group) и ISO, языком ODL (Object Definition Language) из TINA-C, который является расширением IDL и поддерживает современные концепции объектов с разнообразными интерфейсами, групповых объектов, потоковых интерфейсов и описаний QoS (Quality of Service).

Более того, и сам перечень, и каждый язык в нем не перестает развиваться и дополняться. Идеальным вариантом было бы при создании каждой новой архитектуры, или хотя бы - в начале проекта, направленного на создание новой архитектуры, заранее проанализировать, какие протоколы сигнализации и интерфейсы потребуется специфицировать в рамках этой архитектуры и, соответственно, подготовить адекватные языковые средства. Но это вряд ли реалистично, т.к. для определения интерфейсов уже нужно зафиксировать какие-то конкретные языковые нотации.

Существенно также, что перспективные проекты, например, TINA-C, уже не связываются с какими-либо конкретными архитектурами типа TMN или IN. Протоколы взаимодействия в этих проектах будут в основном выражаться в терминах Прикладных программных интерфейсов (API - Application Programm Interface).

Математические основы для вышеупомянутых стандартных средств спецификаций и описаний телекоммуникационных систем составляют следующие общие модели:

- Теория конечных автоматов (расширенных конечных автоматов, машин сообщений);
- Сети Петри;
- Алгебраические модели абстрактных типов;
- Теория множеств;
- Логика предикатов;
- Временная логика.

3. Методология использования современных средств SDL-92/ASN.1/MSC

На рис.1 представлена последовательность использования стандартов ITU (Исследовательская комиссия 10) для описания и/или спецификации телекоммуникационных систем. Эта последовательность состоит из 4-х базовых элементов: текстовые описания, представляющие телекоммуникационную систему с помощью неструктурированной прозы, сценарии обмена сообщениями на языке MSC, описывающие информационные потоки между блоками телекоммуникационной системы, диаграммы SDL, специфицирующие режимы поведения телекоммуникационной системы, коды ASN.1, определяющие синтаксис данных и сообщений.

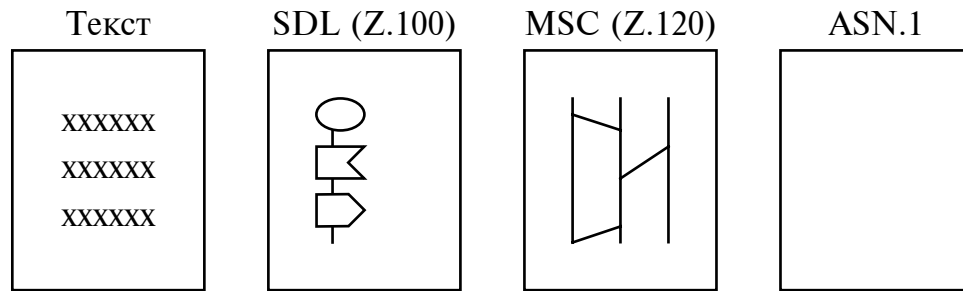


Рис.1. Стандарты ИТУ для описаний и спецификаций телекоммуникационных систем

В первую очередь необходимо подчеркнуть, что в данной статье SDL однозначно трактуется в соответствии с Рекомендациями серии Z.100 Белой книги ИТУ-T, т.е. непосредственно SDL согласно Рек. Z.100 совместно с ASN.1 согласно Рек. Z.105 и совместно с MSC согласно Рек. Z.120.

Текущее состояние SDL-92 отражено в предыдущей статье автора [1] в журнале “Электросвязь”, продолжением которой под известным латинским девизом “Quo vadis?” (куда ведешь?) в сущности и является данная работа.

На рис.2 представлена методология проектирования на базе SDL+, причем явно видна преемственность данной методологии методологическим принципом проектирования программного обеспечения систем коммутации, изложенном автором в работе [2]. Однако, внешние изображения терминологии в данной статье максимально унифицированы с разрабатываемым МККТТ методом руководством по проектированию с использованием SDL, MSC и ASN.1 [3].

Операция формулирования и сбора требований, представленная на рис.2, не прекращается с началом следующей операции. Некоторые требования могут быть переопределены в процессе проектирования после достижения более глубокого понимания той или иной проблемы. Более того, другие операции могут порождать вопросы, которые в свою очередь могут приводить к формулированию новых требований или к изменению требований, ранее сформулированных. Все это показано линиями обратной связи на рис.2.

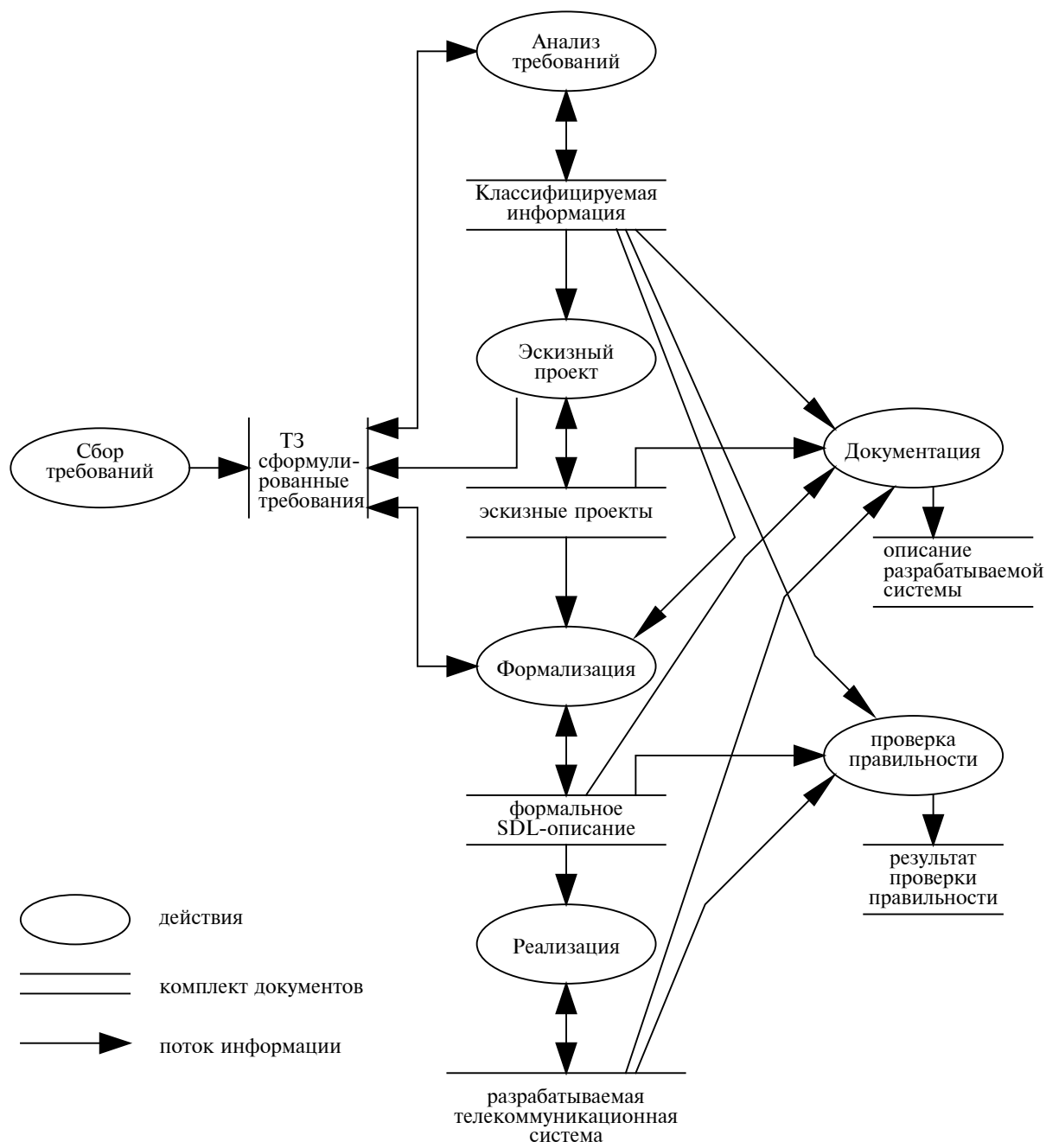


Рис.2. Структурная схема процесса проектирования

Как отмечено в [2], формулирование требований обычно осуществляется на естественном языке и сопровождается неформальными диаграммами (структурными схемами и пр.). Следующая операция архитектурного проектирования, фактически, конвертирует собранные требования в формальные структуры с использованием формального синтаксиса.

Фаза проектирования архитектуры (как это названо в [2]) или операция анализа (как это названо в [3]), также представленная на рис.2, включает уже объектно-ориентированный подход.

Объектно-ориентированный метод хорошо поддерживается и на остальных фазах проектирования методологии на рис.2. Эта методология может быть использована в

комбинации с другими методологиями. Например, методология Рекомендации I.130 [4] формулирует 3 модели: модель с пользовательской точки зрения, модель, показывающая организацию функций сети, и третья модель, представляющая сигнализацию и организацию услуг для поддержки модели пользователя.

4. Графические нотации GDMO

Языковые средства описания управляемых объектов GDMO (Guidelines for the Definition of Managed Objects) определены Рекомендацией X.722 для спецификации управляемых объектов в TMN (Telecommunications Management Network). В силу ограниченного объема статьи и с учетом появления у российских операторов различных протокол-тестеров для ISDN, ОКС-7 и др. решено опустить описание GDMO, отослав интересующегося читателя к Рекомендации X.722, увеличив за счет этого объем описания TTCN в следующем подразделе.

5. TTCN - язык современных протокол-тестеров

Тестовые сценарии представляют собой последовательности событий, описывающих режимы функционирования тестирующей системы и тестируемой протокольной реализации с достаточной степенью детализации для вынесения окончательного вердикта о соответствии или несоответствии. Для абстрактного описания таких тестовых последовательностей в ISO был разработан язык комбинированных древовидных и табличных нотаций (Tree and Tabular Combined Notation - TTCN).

Назначение языка TTCN - описание режимов функционирования и обмена сигналами между тестируемой протокольной реализацией и тестирующей системой. Протокол может быть представлен в форме древовидного графа, отображающего реакции на те или иные входные (в частности - тестовые) сигналы. Как следует из названия язык TTCN использует табличные представления таких деревьев для описания динамики поведения протоколов, а также дополнительные статические таблицы для записи самих тестовых сценариев.

Тестовая архитектура - это описание среды, в которой тестируется данная протокольная реализация. Тестовая архитектура состоит из:

- тестера,
- тестируемой протокольной реализации,
- тестового контекста,
- точек контроля и наблюдения,
- точек доступа к реализации.

Тестер представляет собой тестовый комплект; выполняющий тесты и наблюдающий за результатами.

TTCN базируется на концепции верхнего и нижнего тестеров (см. рис. 3).

Набор тестирующих компонент, взаимодействующих с тестируемой системой (IUT) в точках контроля и наблюдения (PCO) через интерфейс нижнего уровня, называется нижним тестером (LT - lower tester). Набор тестирующих компонент, взаимодействующих с тестируемой системой (IUT) в точках контроля и наблюдения (PCO) через интерфейс верхнего уровня, называется верхним тестером (UT - upper tester).

Система должна содержать по крайней мере одну из тестирующих компонент. Эта компонента будет являться мастер-компонентой (MTC master test component), ответственной за координацию и управление ходом теста и за вынесение окончательного вердикта.

Связь между тестирующими компонентами каждого из тестеров осуществляется через точки координации (CP - coordination points). Координация между верхним и нижним тестером осуществляется посредством процедур координации тестирования (TCP - test coordination procedures).



PDU: Блок Данных Протокола
ASP: Элемент Абстрактных Услуг

Рис. 3. Логическая структура тестирования комфортности

Нижний тестер является более сложным, чем верхний, вследствие необходимости выполнения им функций контроля и наблюдения за блоками данных протокола (PDUs - protocol data units). Блоки данных протокола являются частью абстрактных примитивов (ASP), которые нижний тестер посылает и принимает во время выполнения теста. Фактически в любой момент времени нижний тестер, исполняя какой-то тест, реализует определенную часть соответствующую протокола.

Для проведения тестирования конкретной системы необходимо специфицировать последовательность взаимодействий, или тестовых событий, которые следует подвергнуть наблюдению и контролю в этой системе.

Последовательность таких событий, полностью специфицирующих цель проведения теста, называется тестом (test case). Набор тестов для определенного протокола называется тестовым комплектом (test suite).

Абстрактные тесты содержат всю информацию, необходимую для полной спецификации цели проведения теста в терминах блоков данных протокола, который данная система должна реализовывать в процессе функционирования.

Абстрактные тесты не содержат информации, специфичной для конкретной системы. Однако сама нотация как таковая не является абстрактной; определение TTCN достаточно точно, как в части синтаксиса, так и в части семантики операций, и в настоящее время TTCN приближается к языку программирования.

Определение графической формы нотации (TTCN.GR), в которой вся информация представлена в виде таблиц, составляет основную часть стандарта ISO/IEC 9646-3. Существует также формат нижнего уровня, исполняемый процессором (TTCN.MP).

На рис.3 показана логическая структура тестирования комфортности в соответствии с семиуровневой моделью взаимодействия открытых систем (ВОС), согласно которой требуются спецификации тестов в терминах абстрактных примитивов ASP уровня (N-1), а также в терминах абстрактных примитивов ASP уровня N и блоков данных протокола уровня N. Для того, чтобы удовлетворять таким требованиям, язык TTCN позволяет:

- декларировать типы абстрактных примитивов и блоков данных протокола;
- декларировать точки контроля и наблюдения;
- специфицировать реальные абстрактные примитивы и блоки данных протокола;
- специфицировать различные варианты поведения системы.

Ориентированные на SDL методы спецификации протоколов используют для описания их поведения диаграммы состояний. Конкретные тесты определяются на основании таких диаграмм. Однако, в связи с тем что тестирование соответствия (конформности) в основном ориентировано на наблюдение и контроль последовательности взаимодействий в точке интерфейса между уровнями модели взаимодействия открытых систем (в точке доступа к услуге), то целесообразно специфицировать поведение тестируемой системы в виде дерева, имеющего ветви для

всех возможных вариантов последовательностей взаимодействий, которые могут существовать между двумя данными состояниями протокола.

В TTCN такое дерево взаимодействий называется деревом поведения. Структура дерева представляется посредством увеличивающихся уровней отступов для показа продвижения по дереву относительно времени.

Поведение тестируемой системы (например, прием или посылка абстрактных примитивов) описывается при помощи описаний TTCN. Описания бывают трех типов:

- события,
- действия,
- квалификаторы.

События. Некоторые описания TTCN могут оказаться успешными или неуспешными в зависимости от наступления тех или иных событий. Существуют два типа событий: входные события и таймерные события. Пример входных событий - приход абстрактного примитива в определенной точке контроля и наблюдения. Таймерное событие представляет собой истечение таймера, специфицированного протоколом. Для событий в TTCN используются следующие описания:

```
RECEIVE  
OTHERWISE  
TIMEOUT
```

Действия. Некоторые описания всегда будут успешными. Такие описания называются действиями, которые исполняются тестовой системой. Предполагается, что они всегда исполняются успешно. Для действий в TTCN используются следующие описания:

```
SEND  
IMPLICIT_SEND  
ASSIGNMENT_LIST  
TIMER_OPERATION  
GOTO
```

Квалификаторы. Строки описаний могут включать описания квалификаторов, то есть булевские выражения. Никакие события не могут совпасть и никакие действия не будут исполнены, пока значение квалификатора не станет равным TRUE.

Набор линий описаний, находящихся на одном уровне отступа и принадлежащих одной ветви дерева, называется набором альтернатив линий описания или альтернативами.

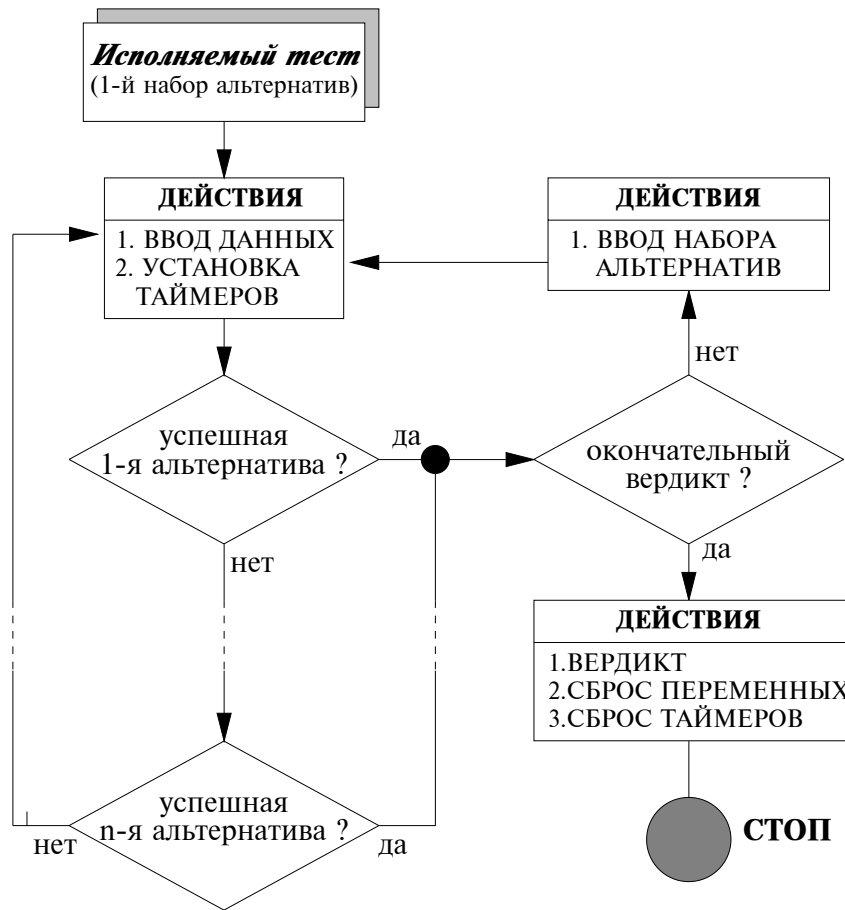


Рис. 4. Цикл исполнения дерева поведения теста

Цикл исполнения теста представлен на рис.4. Исполнение начинается с корня дерева. Первый набор альтернатив циклически повторяется и при этом каждая альтернатива анализируется в том порядке, в котором она появляется в описании теста. Цикл повторяется пока линия поведения не будет успешно выполнена (совпадет). В случае успешного выполнения начинается проверка следующей альтернативы и т.д.

Выполнение приостанавливается при достижении конца ветви или всего дерева. Вынесение окончательного вердикта также может остановить выполнение дерева.

TTCN был разработан с привязкой к абстрактному синтаксису ASN.1 (ISO/IEC 8824:1990). Однако не существует явной связи между типами, используемыми в TTCN и в ASN.1. Это позволяет конструировать типы данных, абстрактные примитивы ASP и блоки данных протокола PDU без использования ASN.1, если разработчик теста не желает этого (например, для протоколов низкого уровня, для спецификации которых обычно не используется ASN.1).

Предопределенные типы Предопределенные (встроенные) типы в TTCN, в основном, являются подмножеством типов ASN.1. Предопределенные типы бывают простые и структурированные (используется одна таблица на каждое определение).

HEXSTRING

Встроенный тип, отсутствующий в ASN.1

BOOLEAN

INTEGER	<i>Встроенные типы, используемые как в ISO/IEC 9646-3, так и в ISO/IEC 8824</i>
BITSTRING	
OCTETSTRING	
Character String	
ENUMERATED	<i>Встроенные типы, определенные в ISO/IEC 8824 и разрешенные к использованию в TTCN без явного декларирования</i>
OBJECT IDENTIFIER	
REAL	
NULL	
...other ASN.1 built-in types	

Рис. 5. Встроенные типы

Простые определяемые пользователем типы Являются подмножеством встроенных типов и образуются посредством ограничения последнего за счет специфицирования:

- перечня значений,
- диапазонов,
- ограничения длин.

Структурированные типы Основное назначение структурированных типов обеспечить возможности структурирования абстрактных примитивов ASP и единиц данных протокола PDU.

TTCN поддерживает асинхронную модель связи. Связь между тестовыми компонентами TC и тестируемой системой IUT обеспечивается через точки контроля и наблюдения (PCOs - points of control and observation). Связь между самими тестовыми компонентами осуществляется через координационные точки (CPs - coordination points). Для описания модели связи используется система с очередями со следующими свойствами:

- каждая точка PCO/CP имеет две бесконечные очереди FIFO: одна очередь для SEND и одна очередь для RECEIVE,
- ровно два объекта должно быть подсоединено к одной точке PCO или CP,
- очередь SEND одного объекта является очередью RECEIVE другого объекта, и наоборот.

SEND позволяет создателю теста описать необходимость передачи ASP определенного типа через данную точку PCO. Описание SEND обозначается следующим образом:

PCO_Identifier ! ASP_Identifier

Обычно блоки данных протокола PDU встроены в ASP, однако имеется возможность использовать описание SEND конкретно для PDU:

PCO_Identifier ! PDU_Identifier

Описание RECEIVE позволяет создателю теста описать необходимость приема абстрактного примитива ASP определенного типа в данной точке контроля и наблюдения PCO. Описание RECEIVE обозначается

PCO_Identifier ? ASP_Identifier

Аналогично для PDU

PCO_Identifier ? PDU_Identifier

ASP задаются в соответствии со стандартным описанием услуги, предоставляемой данным уровнем модели BOC. PDU описываются определениями, заданными в спецификации конкретного протокола. В случае необходимости использования нестандартных PDU они должны быть определены соответствующей таблицей.

6. Техника объектного моделирования ОМТ

Метод ОМТ был разработан Джеймсом Рунбаугом в Риме 1991 г. ОМТ включает в себя три аспекта системного анализа: объектное моделирование, динамическое моделирование и функциональное моделирование.

Разработка ПО поддерживается объектной системой обозначений, используемой для объектного моделирования в системном анализе. Т. о. ОМТ гарантирует постоянную связь между анализом и проектированием.

Однако ОМТ, как признается самими авторами, не очень подходит для архитектурного проектирования, тестирования и внедрения. ОМТ особенно удобна для разработок информационных систем, но она не предполагает специальных конструкций для разработок в реальном времени.

Модель объекта ОМТ включает два вида диаграмм:

- Диаграммы класса: они основаны на хорошо известной системе обозначений взаимоотношений логических объектов, расширенной следующими объектно-ориентированными концепциями: операции и наследование свойств. Диаграммы класса состоят из *классов* (блоков) с *атрибутами* (вторые части блока класса) и операциями (третьи части блока класса), связей (линий) и составных связей (линий с ромбом) между классами, а также иерархий наследования свойств (линий с треугольником).

- Диаграммы экземпляров: они представляют моментальные снимки системы при использовании ее в своем окружении. Диаграммы экземпляров состоят из экземпляров значимого класса, называемых *объекты* (блоки с закругленными углами) и экземпляров связей, называемых *звенья* (линии).

Динамическая модель ОМТ составляется из двух видов диаграмм:

- Следы Событий описывают сценарии течения событий между экземплярами. Следы событий состоят из *экземпляров класса* (вертикальные полосы) и *значимых событий* (горизонтальные стрелки)

- Диаграммы перехода состояний: они основываются на системе обозначений диаграммы состояний Харела (которая является расширением традиционной системы

обозначений для конечного автомата, расширенной уточнением состояния и совпадением внутреннего состояния). Диаграммы перехода состояний состоят из *состояний* (блоки с закругленными углами и переходов (стрелки). Состояния включают в себя *входные и выходные воздействия, действия* диаграммы состояния переходов для сложных состояний. Переходы включают *приемы* значимых событий, условия, действия и *посылки* событий. Существует одна диаграмма перехода состояний на параллельный класс.

Функциональная модель состоит из схем информационных потоков, которые основываются на широко известной системе обозначений структурного анализа. Схемы информационных потоков состоят из *процессов* (кружки) и *потоков данных* (стрелки).

Прослеживаются следующие связи между объектной, динамической и функциональной моделями. В системном анализе объектная модель является центральной моделью ОМТ. Динамическая модель улучшает объектную модель тем, что определяет: когда устанавливаются и удаляются классы, когда вызываются операции с классами, когда имеется доступ к атрибутам и когда создаются, используются и удаляются связи. Функциональная модель предлагает новый взгляд на услуги, обеспечиваемые объектной моделью, путем объединения единичных операций с классами в более крупные процессы или, наоборот, путем детализации сложных операций с классами разбиением на более простые процессы.

Анализ требований согласно [2] (т.н. "системный анализ" в методе ОМТ) использует все диаграммы объектного, динамического и функционального моделирования. Усилия, затрачиваемые на каждое из трех моделирований, зависят от вида разрабатываемой системы. Информационные системы требуют уделить особое внимание объектному моделированию. Высоко-интерактивные системы - динамическому моделированию. Большие системы, состоящие из различных подсистем, модулей и блоков, могут быть первоначально описаны функциональными моделями.

Проектирование ПО [2], называемое в методе ОМТ "объектным проектированием", использует только диаграммы классов. Первоначальная модель анализа объекта улучшается путем использования других моделей анализа и решений, найденных при архитектурном проектировании

Использование ОМТ для разработки систем реального времени особенно существенно в контексте данной работы. Функциональная модель ОМТ слабо помогает при разработке программного обеспечения систем реального времени. Эта модель менее точна, чем динамическая модель для определения динамики системы, а объектное моделирование предоставляет наилучший способ описания системы в своем окружении при ее повторном использовании (объекты лучше подходят для повторного использования, чем иерархические процессы).

Обозначения, предлагаемые ОМТ в отношении динамического моделирования, т.е., следы событий и подмножества диаграмм состояний Харела, к сожалению, не являются формальными. Например, не определяются механизмы для конкретизации понятий класса, для процедуры прохождения событий (очередь/взаимодействие между

параллельными процессами) или для многопоточной обработки. Более того, действия и функции описываются обычным текстом.

7. Информационная сетевая телекоммуникационная архитектура TINA

Консорциум из 40 компаний - лидеров на телекоммуникационном и компьютерном рынках: Alcatel, AT&T, Bellcore, BT, Cable and Wireless, CSELT, Deutsche Telecom, Digital Equipment Corp., ERICSSON, ETRI, EURESCOM, FRANCE TELECOM, Fujitsu, GPT, Hewlett-Packard, Hitachi, IBM, ISIS Distributed Systems, KDD, Korea Telecom, MCI, NEC, Nokia, Northern Telecom, NTT, Oki, PTT Telecom Netherlands, Portugal Telecom, Samsung Electronics, SIEMENS, Stentor, Swiss Telecom PTT, TELECOM Italia, TeleDanmark, Telefonica, Telenor, Telia, Telstra, UNISYS, объединившихся вокруг пятилетнего проекта, разрабатывает новый вид архитектуры программного обеспечения, которая поможет вырваться из пут технического и коммерческого застоя и относится к долгосрочному будущему развитию телекоммуникаций. Это информационная сетевая телекоммуникационная архитектура, которая более известна как объединенная команда *TINA*.

TINA работает следующим образом.

Первое: TINA логически отделяет приложения высокого уровня и физическую инфраструктуру от потребности связывать их друг с другом. Кроме того более стабильные аспекты управления и контроля отделяются от динамично развивающихся технологий и коммерческих приложений. Таким образом, разрабатывается новая архитектура программного обеспечения.

Второе: TINA объединяет все функции управления и контроля в унифицированную логическую архитектуру программного обеспечения, поддерживаемую единой концепцией объединенной обработки данных. Это означает, что вместо существующих отдельных систем, размещенных в конкретных географических точках, эти функции могут осуществляться в любой точке сети.

Это достигается посредством объектно-ориентированной программной технологии, где каждая функция представлена как основной объект программного обеспечения со своим собственным интегрированным управлением и встроенными в него контрольными механизмами. Эти объекты должны реализовываться исключительно через сетевую инфраструктуру, включающую резервируемые "помещения пользователя", со всеми коммуникациями между ними и другими функциями поддержки. Все это управляется структурой процессоров, известных как среда распределенной обработки или DPE.

Внутренние действия этих объектов скрыты от внешнего мира - все они представлены разработчиками как ряд согласованных интерфейсов, которые позволяют подсоединяться либо к ним самим, либо к внешним функциям программного

обеспечения, что и является ключевым требованием, обеспечивающим гибкость всей архитектуры.

Не менее существенного обстоятельство, что существующие технологии и протоколы адаптируются к TINA и также могут подключаться посредством интерфейсов TINA.

Используя эти основные концепцию легко, быстро и экономично можно разрешить создавшиеся проблемы. Всемирные сети из компонентов TINA будут полностью понятны всем. Производители оборудования и программного обеспечения могут продолжать работать со средствами и языками, которые они предпочитают, зная, что они на достаточно длительный срок соответствуют спецификациям интерфейса TINA и будут взаимодействовать с другими элементами среды TINA. Это также облегчает задачу и для новых участников рынка: поставщиков услуг “сверху” или производителей элементов обслуживания “снизу”. В каждом случае они могут разрабатывать продукцию и услуги по своему желанию, а для уверенности во *взаимодействии* им необходима только гарантия, что их разработки сопоставимы с соответствующими требованиями к интерфейсу TINA.

TINA основывается на идее, что будущее телекоммуникаций зависит не только от индивидуальных технических или основанных на стандартах решениях, но также на единственном универсальном основополагающем решении архитектуры программного обеспечения.

С 1997 г, когда проект завершится, индустрия телекоммуникаций будет иметь в своем распоряжении новую, открытую архитектуру программного обеспечения с полным набором спецификаций любой степени сложности на глобальном уровне и с использованием коммуникационных технологий завтрашнего дня.

8. Перечень английских сокращений

ANSI C	-
API	- Application Programm Interface
ASN.1	- Abstract Syntax Notation One
ASP	- Abstract service primitive
ATS	- Abstract Test Suite
B-AN	- Broadband Access Network
BOC	-
CORBA	- Common Object Request Architecture
CP	- Coordination Points
DPE	-
Estelle	- Extended State Transition Language
ETSI TC MTS	- Methods for Testing and Specification
FPN	- Freephone (supplementary service)
GDMO	- Guidelines for the Design of Managed Objects

HMI	- Human-Machine Interface
IDL	- Interface Definition Language
IN	- Intelligent Network
ISDN	- Integrated Services Digital Network
ISO	-
ISO/IEC	-
ITU	- International Telecommunication Union
ITU-T	- ITU Telecommunications Standardization Sector
IUT	- Implementation Under Tests
LOTOS	- Language of Temporal Ordering Specification
LT	- Lower Tester
MML	-
MSC	-
MTC	- Master Test Component
NIT	- Network Integration Testing
ODL	- Object Definition Language
ODP	-
OMG	- Object Management Group
OMT	-
OOA	- Object Oriented Analysis
OSI	-
PCO	- Points of Control and Observation
PDU	- Protocol Data Unit
PICS	- Protocol Implementation Conformance Statement (proforma specification)
PIXIT	- Protocol Implementation eXtra Information for Testing (proforma specification)
Q3VAD	- Q3 Value-Added Data
QoS	- Quality of Services
SCCP	- Signalling Connection Control Part
SDL	- Specification and Description Language
SN	- Service Node
TCP	- Test Coordination Procedures
TMN	- Telecommunication Management Network
TSS&TP	- Test Suite Structure and Test Purpose
TTCN	- Tree and Tabular Combined Notation
UT	- Upper Tester
UUS	- User-to-User Signalling
VPN	- Virtual Private Network
X/OPEN	-

Литература

1. Фаергеманд О., Сарма А., Гольдштейн Б. SDL-92: анализ современного состояния// Электросвязь. - 199 . - №
2. Гольдштейн Б.С. Технологические аспекты проектирования программного обеспечения цифровых систем коммутации// Электросвязь. - 1988. - № 10
3. Варакин Л.Е., Осипов В.Г., Филюшин Ю.И. Концепция построения интеллектуальной сети России..Электросвязь.-1994.-№2.
4. Anders Olsen, Ove Faergemand, Birger Moeller Pedersen, J. R. W. Smith and Rick Reed, "Systems Engineering Using SDL-92", North Holland, 1994