

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
МИНИСТЕРСТВО СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М.А. БОНЧ-БРУЕВИЧА

А.А. Атцик, А.Б. Гольдштейн, М.А. Феноменов

ЭКСПЛУАТАЦИОННОЕ УПРАВЛЕНИЕ
ИНФОКОММУНИКАЦИЯМИ

Учебное пособие для практических занятий
и лабораторных работ по дисциплинам
«Системы управления инфокоммуникациями»
и «Бизнес-процессы Операторов связи»

Рекомендовано УМО по образованию в области инфокоммуникаций в качестве учебного пособия для подготовки бакалавров/магистров техники и технологии по направлениям 210700 «Инфокоммуникационные технологии и системы связи» и 230100 «Программная инженерия»

Санкт-Петербург

2013

УДК 621.391:004.7(075.8)

ББК 3889я73

Г 63

Рецензенты:

кафедра автоматической электросвязи, зав. кафедрой профессор А.П. Пшеничников (МТУСИ), доктор технических наук, профессор Н.А. Соколов (ЛО ЦНИИС)

*Утверждено
редакционно-издательским советом университета
в качестве учебного пособия*

Г 63 А.А. Атцик, А.Б. Гольдштейн. ЭКСПЛУАТАЦИОННОЕ УПРАВЛЕНИЕ ИНФОКОММУНИКАЦИЯМИ: учебное пособие для практических занятий и лабораторных работ по дисциплинам «Системы управления инфокоммуникациями» и «Бизнес-процессы Операторов связи»: учебное пособие / ГОУВПО СПбГУТ. СПб, 2013 – 80 с.

Учебное пособие содержит теоретический материал по истории развития систем поддержки эксплуатации сетей связи. Описывает методологию NGOSS (NewGenerationOperationsSystemsandSoftware) построения OSS/BSS инфраструктуры. В пособии также приведена краткая характеристика программно-аппаратного OSS комплекса АРГУС, используемого в лабораторном курсе.

УДК 621.391:004.7(075.8)

ББК 3889я73

© Государственное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича», 2013

Оглавление

Перечень сокращений	4
1 Введение в эксплуатационное управление	7
2 Историческое развитие систем эксплуатационного управления.....	8
3 Сеть управления телекоммуникациями TMN	15
4 Концепция NGOSS.....	18
4.1 Методология и жизненный цикл	21
4.2 Модель бизнес-процессов eTOM.....	23
4.3 Модель общей информации и данных SID	29
4.3.1 Основы языка UML и его использование в SID	29
4.3.2 Концепция SID	34
4.4 Технологически-нейтральная архитектура TNA	39
4.5 Карта приложений TAM	40
5 Интерфейсы NGOSS для интеграции компонентов OSS	44
5.1 Причины и подходы к интеграции компонентов OSS	44
5.2 Интерфейсы взаимодействия MTNM, MTOSI, OSS/J.....	44
5.2.1 OSS/J	44
5.2.2 MTNM.....	45
5.2.3 MTOSI	51
6 Решения OSS АРГУС	52
6.1 О продуктах АРГУС.....	52
6.2 АРГУС - Технический Учет	54
6.3 Абонентский Отдел	55
6.4 Комплексная Техническая Поддержка.....	58
7 Литература	59

Перечень сокращений

ABE - AggregatedBusinessEntities

ADSL - AsymmetricDigitalSubscriberLine

ATM - AsynchronousTransferMode

BML - businessmanagementlayer

BSS - BusinessSupportSystems

CCV - commoncommunicationvehicle

CLI - CommandLineInterface

CORBA - CommonObjectRequestBrokerArchitecture

CTP - ConnectionTerminationPoint

DSLAM - DigitalSubscriberLine Access Multiplexer

EMS - elementmanagementsystem

EML - elementmanagementlayer

ERP - Enterprise Resource Planning

eTOM - enhancedTelecomOperationsMap

HRM - Humanresourcemanagement

HTTP - HyperTextTransferProtocol

IP - Internetprotocol

IPTV - IP television

ITU-T- InternationalTelecommunicationUnion - Telecommunicationsector

IT- InformationTechnologies

JMS-javamessageservice

ME- ManagedElement

MPLS- multiprotocollabelswitching

MRP- MaterialRequirementPlanning

MTOSI- multi-technologyoperationssysteminterface

MTNM- Multi-TechnologyNetworkManagement

NEL -networkelementlayer

NGN-NextGenerationNetwork
NGOSS -NewGenerationOperationsSystemsandSoftware
NML -networkmanagementlayer
NMS- networkmanagementsystem
OS-OperationSystem
OSS- OperationsSupportSystem
OSS/J- OperationsSupportSystemJava
PDH-PliesiochronousDigitalHierarchy
PTP-PhysicalTerminationPoint
QoS-QualityofService
RT&A-Resource testing&activation
SCM- Serviceconfigurationmanagement
SDH- SynchronousDigitalHierarchy
SID- SharedInformationandDatamodel
SLA- ServiceLevelAgreement
SML -servicesmanagementlayer
SNC-Subnetworkconnection
SNMP- simplenetworkmanagementprotocol
SOAP- SimpleObject Access Protocol
SONET- Synchronousopticalnetworking
TAM- TelecomApplicationsMap
TL- TopologicalLink
TMF -TeleManagementForum
TMN- TelecommunicationsManagementNetwork
TNA- TechnologyNeutralArchitecture
TOM- TelecomOperationsMap
TP-TerminationPoint
UML-UnifiedModelingLanguage

VoIP-Voiceover IP

WFM-Workforcemanagement

xDSL- digitalsubscriberline

АБС- Агрегированная Бизнес Сущность

АО-Абонентский отдел

АТС- Автоматическая телефонная станция

КТП-комплексная техническая поддержка

МСЭ-Т - Международный союз электросвязи

ПО- программное обеспечение

СИП- Стратегия, Инфраструктура и Продукт

СОЗ-Система обработки заказов

ТУ-технический учёт

УП- Управления Предприятием

1 Введение в эксплуатационное управление

Задача эксплуатационного управления для Оператора связи состоит в том, чтобы обеспечить основные (то есть приносящие доход) процессы – это процессы предоставления услуг связи. Состав функций эксплуатационной поддержки меняется в зависимости от портфеля услуг Оператора связи и технологий предоставления этих услуг. Например, для услуги IP-телевидения такими функциями являются обеспечение заданных характеристик скорости переключения каналов, количества кадров в секунду, а для традиционной телефонии – характеристик соотношения сигнал/шум, сопротивления и емкости абонентской линии.

Разнообразие, качество и стоимость телекоммуникационных услуг определяют доход Оператора связи. С технической стороны вопроса сетевая и ИТ-инфраструктура Оператора постоянно модернизируются – растет число используемых телекоммуникационных технологий предоставления услуг связи, благодаря чему создаются условия для быстрой разработки новых типов услуг. Эта тенденция, с одной стороны, дает конкурентные преимущества Оператору связи и способствует привлечению новых клиентов. Однако разнообразие предоставляемых услуг повышает интенсивность клиентских обращений в эксплуатационные службы Оператора, а новые технологии предъявляют новые требования к компетенции обслуживающего персонала. Эти два фактора приводят к повышению стоимости содержания традиционных эксплуатационных служб Оператора и делают актуальным внедрение автоматизированных систем поддержки эксплуатации (OperationsSupportSystem, **OSS**).

Современные системы **OSS** позволяют автоматизировать такие процессы, как:

- учет (инвентаризацию) знаний об инфраструктуре и об услугах Оператора связи,
- взаимодействие с абонентами,
- продажа услуг,
- регистрация и обработка заказов услуг,
- поиск и устранение неисправностей,
- мониторинг качества предоставления услуг
- и многие другие.

При разработке и внедрении **OSS**-систем возникает ряд трудностей. Во-первых, сеть Оператора связи в большинстве случаев состоит из разнородного оборудования – устаревшего и современного, разных производителей, с использованием различных технологий. Вторая проблема заключается в отсутствии формализации деятельности эксплуатационных служб Оператора. Основная же трудность связана с необходимостью интеграции компонентов **OSS** в единый, управляющий всей телекоммуникационной сетью комплекс. Последнее в большей степени связано с тем, что работа приложений **OSS** взаимосвязана. Приложения систем поддержки эксплуатации совместно используют и изменяют информацию об инфраструктуре, услугах, клиентах Оператора. Результаты выполнения работы одного приложения используются в работе другого. Поэтому при внедрении нового компонента **OSS** всегда возникает необходимость его интеграции с другими, уже существующими приложениями.

Так, например, приложение, автоматизирующее службу технической поддержки Оператора связи должно быть интегрировано с приложением технического учёта сети, чтобы сотрудник технической поддержки мог видеть схемы подключения того или иного абонента.

Приложение, автоматизирующее и организующее на сетиподключение абоненту новых услуг, должно быть интегрировано с биллинговой системой, чтобы изменять правила тарификации услуг, предоставляемых абонентам.

Отсутствие стандартных подходов к организации процессов эксплуатационного управления, учета информации об инфраструктуре, способа доступа к функциям программных приложений лишает разработчиков OSS-систем возможности быть заранее готовыми к внедрению своих продуктов у любых Операторов. В каждом случае требуется глубокое изучение сетевых технологий, моделей предоставления услуг и уже существующей инфраструктуры управления услугами.

Решение этих и других сопутствующих вопросов будет подробно рассматриваться в данном учебном пособии.

2 Историческое развитие систем эксплуатационного управления

В эволюции эксплуатационного управления сетями связи можно выделить несколько этапов, между которыми изменялись технологии и методы организации управления телекоммуникационными услугами и ресурсами.

Первые эксплуатационные процессы создавались, исходя из необходимости поддержания сети связи в рабочем состоянии. Идея *предоставления услуг* в этих процессах являлась вторичной. Вместе с установкой оборудования связи у Оператора вынужденно сформировались процессы, обеспечивающие функционирование сети. Исторически первыми у традиционного Оператора фиксированной сети связи появились процессы *развития сети, подключения новых абонентов и процессы обеспечения бесперебойной работы сети*.

Первоначально процессы эксплуатационной поддержки выполнялись подразделениями Оператора связи, в которых регистрировались, хранились и обрабатывались заказы на подключение к сети, жалобы на работу сети, наряды на устранение выявленных неисправностей. В этих службах использовались преимущественно ручные операции (выполняемые людьми), что и дало название первому этапу развития – *ручная организация эксплуатационной деятельности* (см. рис. 2.1).

При таком построении процессов эксплуатации за каждым сетевым элементом закрепляется техник; Оператор регистрирует заказы и жалобы, актуализирует абонентские данные; диспетчер планирует работу техников, назначает наряды. Образно, данный способ организации эксплуатационной деятельности можно обозначить, как «управление ресурсами». По отдельно взятому ресурсу всегда можно определить ответственного за его конфигурацию и состояние. При необходимости изменения конфигурации ресурса необходимо уведомить закрепленное за ним ответственное лицо.

В первой половине двадцатого века сети связи развивались равномерно, процессы эксплуатационной поддержки не отличались особой сложностью. Этот факт объясняется относительно небольшой емкостью сети (в частности, из-за высокой стоимости подключения услуг связи) и небольшим количеством телекоммуникационных технологий организации связи, работа которых представлялась наглядной.

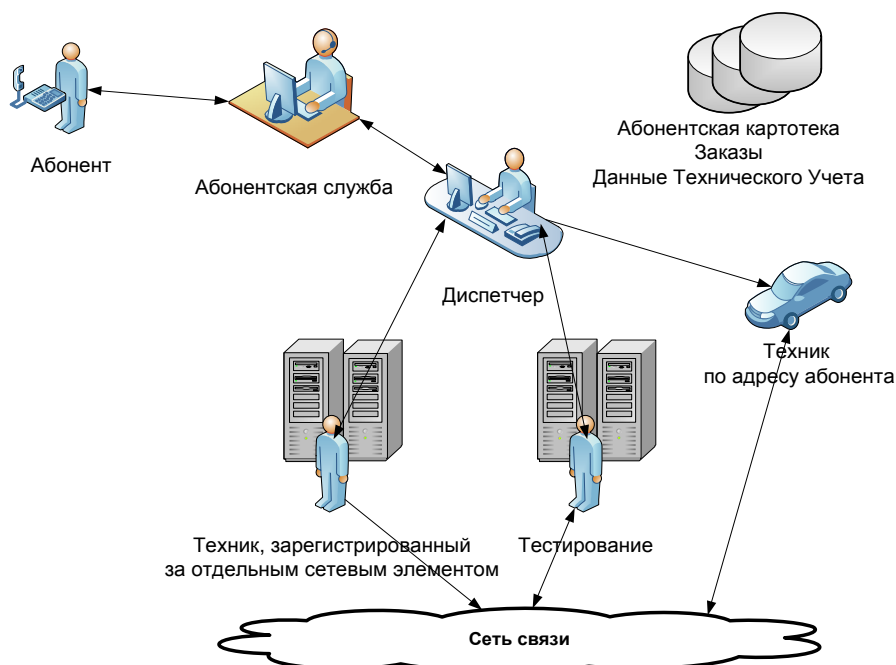


Рис. 2.1. Ручная СОЗ (система обработки заказов)

Вместе с научно-техническим прогрессом и распространением услуг телефонной связи стоимость подключения неуклонно снижалась, количество желающих подключиться росло. Рост номерной емкости сети привел к расширению номенклатуры телекоммуникационных устройств, росту объемов информации. Для поддержания приемлемого качества обслуживания абонентов Оператору приходилось содержать весьма большой штат инженеров, монтажников и других специалистов, что существенно увеличило эксплуатационные расходы. Все это, в совокупности с желанием максимизировать прибыль, привело к необходимости оптимизации ручной эксплуатационной деятельности для условий предоставления базовой услуги - телефонии. Начался поиск инструментов и методов эффективного эксплуатационного управления. Ручной способ ведения эксплуатационной деятельности уступил место второму этапу развития – *централизованной* организации (см. рис. 2.2), в основе которой лежит централизация управления рабочей силой.

За счет использования поставляемых вместе с сетевым оборудованием систем управления стало возможным выполнять конфигурирование сетевых элементов удаленно, силами меньшего числа инженеров и техников. Таким образом, часть эксплуатационных процессов выполнялась централизованно, но большинство задач все же требовало личного присутствия специалистов. Единственное отличие этого варианта от ручного состоит в том, что часть техников находятся в центральном офисе, а их терминалы подключены к системе управления сетевым оборудованием посредством внутренней сети передачи данных. При такой организации технический персонал

имеет возможность удаленно получать доступ к сетевому оборудованию вместо того, чтобы физически находиться в том же месте, что и оборудование. В результате меньшее количество сотрудников могут выполнить больший объем работ.

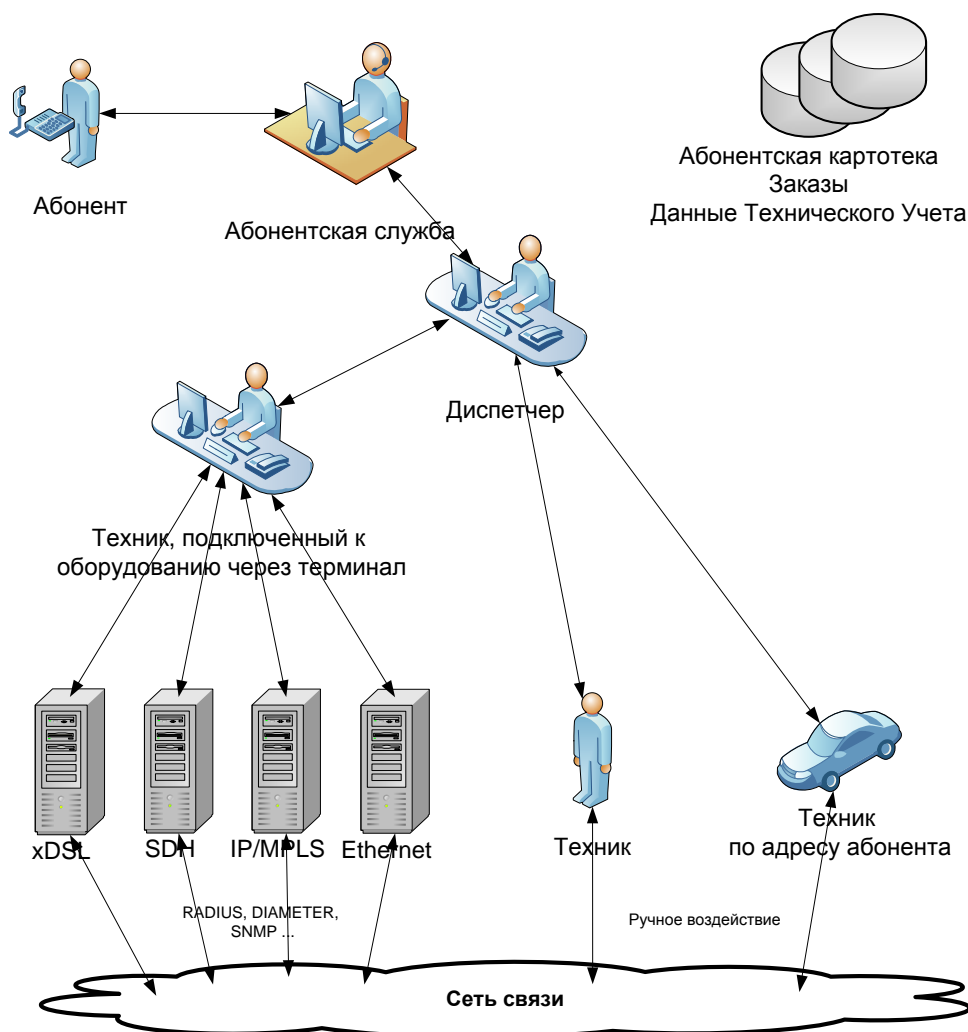


Рис. 2.2. Централизованная система обработки заказов

Централизованная организация повысила эффективность эксплуатационной деятельности – то есть замедлила рост количества обслуживающего персонала относительно роста количества абонентов и услуг. Но на фоне скачкообразного развития технологий – появления Операторов мобильной связи, роста емкости сети и спектра предлагаемых услуг – требовалось все больше персонала для эксплуатации и обслуживания сети. Поэтому Операторы охотно пошли по пути механизации эксплуатационных операций, определив название следующего, третьего этапа эволюции организации эксплуатационных процессов – *механизированный*.

До механизированного этапа организации процессов эксплуатационной поддержки автоматические средства использовались персоналом только как инструменты, которые позволяли ускорить выполнение рутинной части работы, однако логикой использования этих средств управляли люди. Данное обстоятельство не позволяло вести оперативный контроль и управление всеми заказами, так как информация об открытых заказах не была собрана в одном месте, а была распределена между ответственными: диспетчерами, инженерами, техниками. Следующий этап развития организации эксплуатационных процессов был стимулирован

расширением спектра услуг связи, которые Оператор связи предоставлял своим клиентам с внедрением новых технологий. Помимо телефонии и небольшого количества дополнительных услуг, появились услуги доступа в Интернет, IP-телевидения и многие другие. Механизированная организация эксплуатационной деятельности – это первая попытка автоматизировать логику выполнения решаемых задач с централизованным контролем и отслеживанием их состояния.

Внедрение программного обеспечения для управления процессами обработки заказов и жалоб на услуги обеспечивает координацию всех критически важных видов работ, как показано на рис.2.3. В механизированной среде техники являются посредниками между системой управления заказами и конфигурацией телекоммуникационной сети. После регистрации заказа или жалобы автоматизированная система **OSS** берет на себя координацию и отслеживание всех работ, выполняемых персоналом Оператора. Такой принцип управления можно обозначить как «управление услугами». При переходе к пакетным сетям, все меньше ресурсов является индивидуальными, и конфигурирование сетевого элемента происходит, исходя из требований целой группы заказов на услуги связи, в реализации которых данный сетевой элемент участвует.

Именно подход механизированной организации эксплуатационных процессов закрепился у многих традиционных Операторов связи, как наиболее приемлемый с экономической и технической точки зрения. Во-первых, стоимость такой системы и количество занятого персонала в эксплуатации невелики. Во-вторых, до тех пор, пока интенсивность регистрации заказов не превышает некоторых пороговых значений, такая схема обработки заказов позволяет быстро их обслуживать. Малая интенсивность регистрации заказов у традиционных Операторов связи объяснялась низким уровнем конкуренции и небольшим количеством предоставляемых услуг (преимущественно единственной услуги – телефонии). В таких условиях потребность клиентов в частом конфигурировании услуг оставалась относительно низкой (от одного абонента заказы регистрировались раз в квартал, год и даже реже).

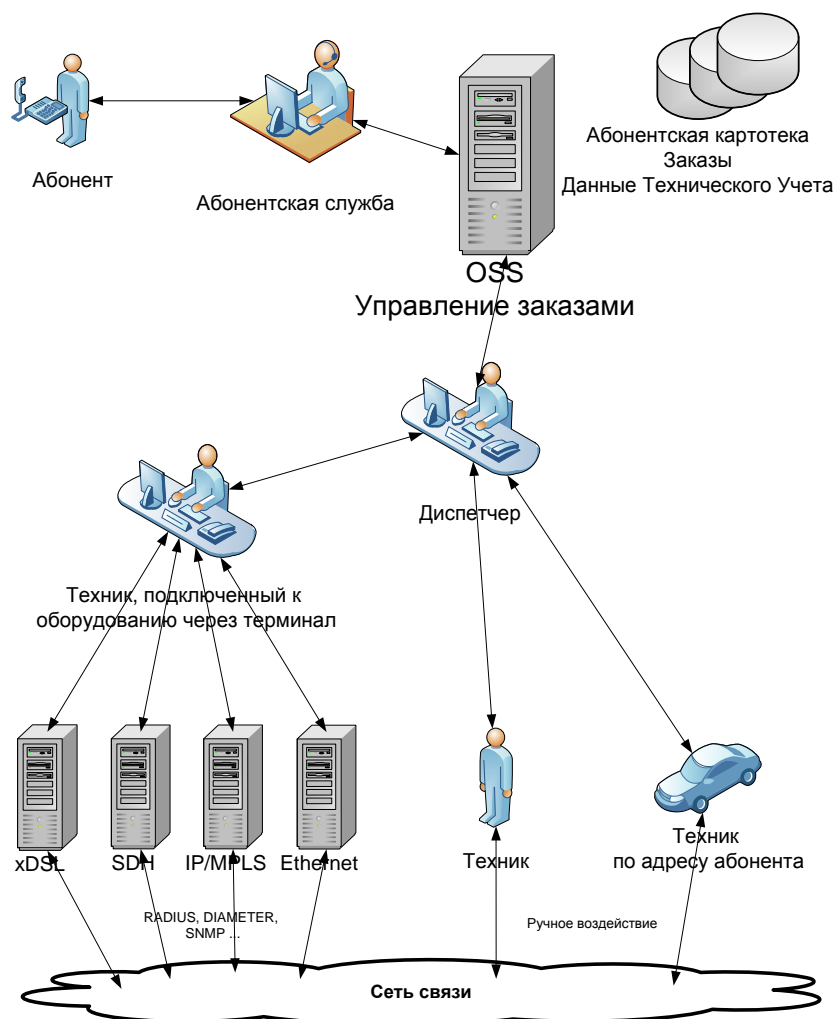


Рис. 2.3. Механизированная система обработки заказов

Следующий эволюционный этап развития организации эксплуатационных процессов наблюдается при построении Операторами сетей связи следующего поколения - **NGN**. Ориентированная на услуги, эта концепция изменила среду и требования к системам эксплуатационной поддержки.

Во-первых, акцент на услуги подразумевает резкий рост количества типов услуг и их модификаций (десятки, сотни типов услуг). Услуги **NGN** Оператор связи постоянно развивает: создает новые, модифицирует существующие. Это провоцирует рост интенсивности обращений клиентов в систему эксплуатационной поддержки, что требует пересмотра подходов к построению ее архитектуры.

Во-вторых, растет количество технологий, которые участвуют в реализации услуг связи. Это повышает требования к компетенции персонала, занятого в обработке заказов на услуги связи, в случае механизированной организации процессов эксплуатационной деятельности.

В-третьих, происходит изменение бизнес-модели предоставления услуг связи, которое заключается в том, что конечную услугу связи пользователю предоставляют несколько компаний совместно. Например, одна компания владеет сетью доступа, вторая – магистральной сетью, а третья – контентом (телевидение, радио, игры). В такой среде эксплуатационные процессы распределяются между несколькими заинтересованными сторонами. Это

требует регламентирования и согласования их взаимных обязательств для каждой участвующей в реализации стороны, так как пользователь будет оценивать услугу в целом.

Все эти перечисленные свойства сетей следующего поколения сделали актуальной разработку нового поколения архитектуры эксплуатационного поколения – **NGOSS** (New Generation Operations Systems and Software) в рамках некоммерческого объединения Операторов, производителей оборудования и интеграторов систем **OSS–TeleManagement Forum (TMF)**.

Обобщенный вид следующего поколения организации процессов эксплуатационной поддержки – автоматизированной – по аналогии с предыдущими этапами представлен на рис. 2.4.

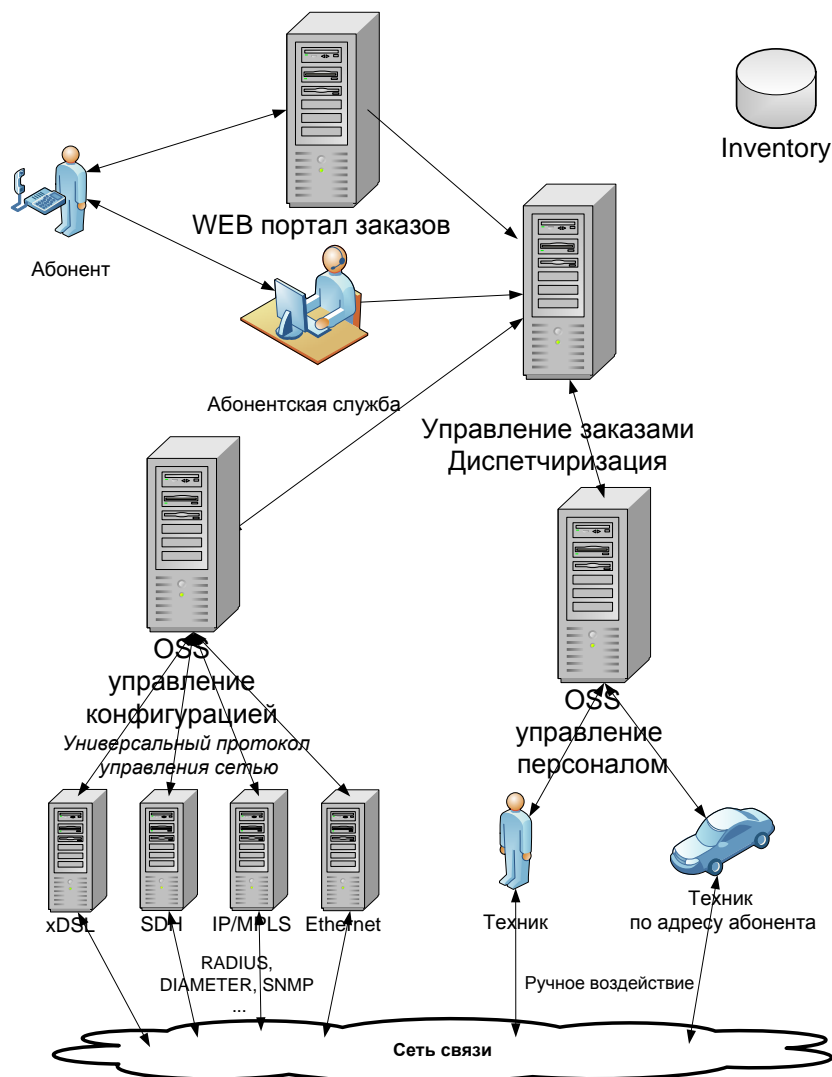


Рис. 2.4. Автоматизированная система обработки заказов

Автоматизированный подход заключается в сведении к минимуму человеческого участия в процессах эксплуатационной поддержки. Надо отметить, что в отличие от технологических процессов предоставления услуг, некоторые операции обработки заказов на услуги не могут быть автоматизированы в принципе (монтаж кабеля, установка оборудования). Такие эксплуатационные операции в сети **NGN** занимают лишь малую долю и выполняются, как правило, один раз для каждого клиента или группы клиентов на этапе подключения к сети связи.

Направленность систем **OSS** на сквозную автоматизацию (т.е. исключение ручных операций) определила основные сложности построения таких систем, которые заключаются в унификации доступа к компонентам ПО и создании универсального набора программных интерфейсов управления сетевыми элементами. Такие задачи широко обсуждаются, и в рамках архитектуры **NGOSS** предложены и апробированы подходы к построению компонентов системы эксплуатационного управления.

Особенности автоматизированной реализации эксплуатационных процессов обеспечивают Оператору снижение эксплуатационных издержек и повышение конкурентных преимуществ. Среди плюсов сквозной автоматизации процессов можно выделить следующие:

- Снижается время реализации услуг (измеряется в секундах, а не в часах и днях).
- Упрощаются процессы отслеживания, контроля и управления работой системы эксплуатационных служб.
- Снижается влияние человеческого фактора, так как ручной труд замещается автоматическими системами управления.
- Уменьшается численность персонала, занятого в процессах эксплуатационной деятельности, и снижаются требования к его профессиональным знаниям и навыкам.

Причины перехода к автоматизированной эксплуатационной деятельности

Потребность Оператора связи в автоматизации эксплуатационной деятельности диктуется разными требованиями, к числу которых относится как повышение качества обслуживания клиентов (повышение скорости обслуживания, достоверности принимаемых решений в процессе обслуживания), так и экономические соображения: снижение эксплуатационных издержек, т.е. снижение затрат на заработную плату персонала, непосредственно занятого в обслуживании клиентов.

Политика традиционных Операторов связи не акцентирует особого внимания на управлении количеством персонала и действует по принципу «наращивать персонал пропорционально поступающей на него нагрузке». Такая политика объясняется особенностями предоставления услуг традиционными Операторами.

Традиционный Оператор связи предоставляет услуги, которые базируются на нескольких технологиях: ТФОП с двухпроводной медной абонентской линией на “последней миле”, передаче данных на базе xDSL, магистральных сетях SDH, ATM, IP/MPLS. При этом для подключения и модификации услуг необходимо взаимодействие только с доменом ресурсов xDSL. Магистральная сеть управляется отдельно, независимо от предоставляемых услуг, она является общим ресурсом для передачи данных. За счет функционального деления задач в процессах предоставления услуг снижаются общие требования к компетенции персонала эксплуатационных служб. Так, одна эксплуатационная группа занимается обслуживанием сети доступа, другая - магистральной сетью. Из сложившегося ресурс-ориентированного подхода, можно сделать несколько выводов. Во-первых, имеет место низкая потребность клиентов в обращении к Операторским службам, что объясняется небольшим числом типов предоставляемых услуг. Основными мотивами обращений будут являться: «подключиться к сети», «модифицировать тарифный план», «пожаловаться на работу услуги» и прочие, которые появляются у клиентов достаточно редко (при надлежащем качестве предоставления услуг). Во-вторых, при обслуживании заказов и жалоб, а также для выдачи справок перед обслуживающим персоналом ставятся требования к компетенции в

определенной технологии, в данном случае это xDSL и сервер доступа, регулирующие доступ в Интернет. Другими словами, у традиционных Операторов связи наблюдается низкая нагрузка на эксплуатационные службы и невысокие требования к компетенции персонала этих служб (компетенция в узкой области). Таким образом, становится понятна актуальность упомянутого выше принципа увеличения обслуживающего штата пропорционально поступающей на него нагрузке.

Это правило перестает работать при эксплуатации сетей **NGN**. Основным отличием концепции сетей **NGN** от предшествующих является ориентированность на услуги. На базе сети **NGN** Оператор формирует целый портфель продуктов и услуг, в процессе предоставления которых используется множество инфокоммуникационных технологий. К примеру, для реализации услуги **IPTV** могут понадобиться ресурсы серверов приложений **IPTV**, каналы **MPLS**, порты **DSLAM**.

Для реализации такого рода услуг необходимо одновременно управлять множеством ресурсов разных технологий. При этом, помимо того, что каждая из услуг может базироваться на множестве технологий, она имеет различные конфигурации (например, **IPTV** с поддержкой **QoS** или по принципу *besteffort*). Наличие разных конфигураций услуг учащает обращение клиентов в Операторскую службу.

Второй особенностью эксплуатации сетей **NGN** являются повышенные требования к специалистам, которые заняты в обслуживании клиентов. Требования к знанию множества технологий разных уровней (например, **IPTV**, **IP**, **MPLS**, **Ethernet**, **xDSL**) повышают ценность и стоимость каждого сотрудника, который занят настройкой услуг.

Таким образом, в условиях эксплуатации сетей **NGN** наблюдается рост клиентских обращений, то есть рост нагрузки на службы Оператора, но также имеет место повышение требований к компетенции специалистов. Оба фактора приводят к росту эксплуатационных издержек. Одним из эффективных путей сокращения эксплуатационных затрат Оператора в такой ситуации является внедрение автоматизированных систем эксплуатационной поддержки.

3 Сеть управления телекоммуникациями **TMN**

В предыдущем параграфе были рассмотрены основные факторы, которые делают очень важным и актуальным построение систем автоматизации эксплуатационной деятельности Оператора связи. В этом параграфе мы раскроем основные вопросы, возникающие в процессе разработки и внедрения таких систем, и подходы к их решению.

Первые системы для автоматизации эксплуатационных задач разрабатывались и поставлялись вместе с телекоммуникационным оборудованием от производителя. Эти программные средства позволяли планировать запуск различных профилактических работ (например, тестирование по расписанию), обеспечивали мониторинг и предоставляли пользователю (персоналу) упрощенные средства конфигурирования оборудования, то есть избавляли от необходимости использовать низкоуровневые интерфейсы конфигурирования (**CLI**). Однако область применения подобных решений была ограничена, как правило, одной серией оборудования от одного производителя. Узость применения подобных решений создавала основные проблемы, так как не позволяла

организовать централизованное управление гетерогенной сетью, т.е. мультитехнологической сетью, построенной на основе оборудования разных производителей. Попытки сторонних организаций создать универсальный продукт, который смог бы взять на себя задачи централизованного управления сетью и управлять всем разнородным парком телекоммуникационного оборудования Оператора требовали несоизмеримых трудозатрат, связанных с поддержкой множества проприетарных протоколов управления и их версий. Разработать конечное решение становилось возможным только под индивидуальные требования конкретного Оператора связи, в которых нужно было поддерживать обозначенный набор протоколов и оборудования. Таким образом, в условиях отсутствия стандартных подходов к построению подобных систем их разработка влекла за собой большие вложения и риски.

Первой попыткой стандартизации подхода к построению систем управления в области телекоммуникаций была концепция «Сеть управления телекоммуникациями» (**TMN**, Telecommunications Management Network), которая была разработана и поддерживается **МСЭ-Т (ITU-T)**. В рамках этой концепции было рассмотрено множество аспектов построения систем управления телекоммуникационными сетями, таких как функциональная, логическая и физическая архитектуры, информационная модель, уровни эксплуатационного управления и многое другое. Традиционный взгляд на задачи управления представлен в логической модели управления **TMN**, изображенной на рисунке 4.1.

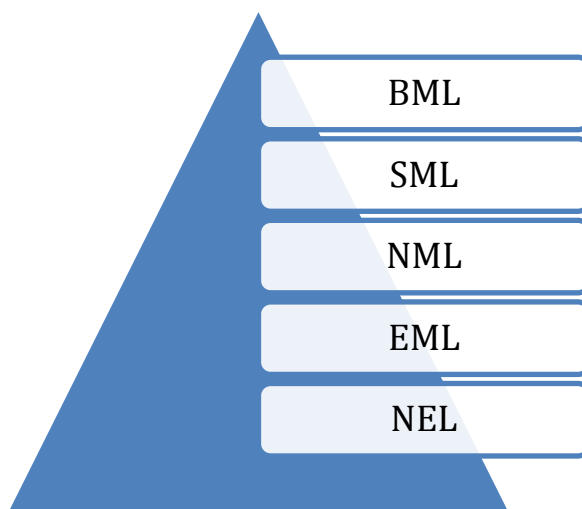


Рис. 4.1. Логическая модель управления TMN

В этой модели заложены следующие принципы:

- Функции управления реализует не одна система, а совокупность систем, взаимодействующих друг с другом.
- Декомпозиция задач управления по уровням (начиная снизу):

NEL – NetworkElementLayer – уровень сетевых элементов – на этом уровне располагаются отдельные сетевые элементы: коммутаторы, маршрутизаторы, мультиплексоры и т.п.

EML – ElementManagementLayer – уровень управления сетевыми элементами – на этом уровне находятся программные средства, которые позволяют удаленно управлять работой сетевых элементов. Как правило, такими системами производители снабжают свое сетевое оборудование.

Основными функциями этих систем являются конфигурирование и мониторинг сетевых элементов.

NML – NetworkManagementLayer – уровень управления сетью – на этом уровне располагаются системы, которые занимаются централизованным управлением сетью и оперируют понятиями, лишенными специфики отдельных технологий, например, такими, как «узел сети», «точка доступа к сети», «соединение», «задержка передачи пакетов от точки к точке», «джиттер» и т.п. Функции таких систем аналогичны функциям систем уровня EML, но реализованы в масштабах всей сети Оператора независимо от многообразия сетевого оборудования и его производителей.

SML – ServicesManagementLayer – уровень управления услугами – на этом уровне находятся системы, управляющие конфигурацией услуг при помощи вызова функций системы управления сетью. Данный уровень особенно важен, когда у Оператора связи появляются услуги, каждая из которых предъявляет разные требования к сети. Примерами таких услуг будут VoIP, IPTV, Videoconference и прочие. У каждой из них будут свои требования к проценту потерь пакетов, среднему времени задержки пакетов, джиттеру пакетов. Для каждой услуги будут разными процессы подключения/модификации, устранения неисправностей, оценки **QoS**.

BML – BusinessManagementLayer – уровень управления бизнесом – на этом уровне располагаются системы управления бизнесом предприятия. Они управляют кадрами, финансами, целями предприятия, стратегией и прочими аспектами, связанными с бизнесом.

Обозначенный подход, предполагающий разделение систем по функциональным уровням, сохраняется и сегодня. Комплексное решение для Оператора связи собирается из отдельных частей, каждая из которых имеет узкую область применения. Общая логическая архитектура позволяет разделить функции по системам и предупредить их дублирование или пробелы в конечном решении.

После того как был обозначен набор систем управления для конкретных решений Оператора связи, для организации совместной работы систем необходима их интеграция в единое решение. Интеграция предполагает, что системы будут совместно использовать общее хранилище информации о клиентах, продуктах, услугах и инфраструктуре Оператора связи, а также будут использовать функции друг друга, предупреждая тем самым их дублирование.

Как было отмечено выше, задача интеграции разных систем друг с другом требует больших трудозатрат. Основную сложность вызывает раскрытие формата данных в приложениях (то есть, взаимная корреляция разных информационных моделей) и организация единообразного доступа к внутренним функциям всех приложений. Одно из решений этой проблемы было предложено **ITU-T**. Для связей систем друг с другом был разработан универсальный интерфейс **Q3**. Все разрабатываемые системы в рамках концепции **TMN** должны поддерживать этот интерфейс, как основу взаимодействия между системами разных уровней. Возможно, подход с использованием всего одного универсального интерфейса, а также его сложность и были причиной отсутствия полноценных реализаций концепции **TMN**. Реализации интерфейса **Q3** у разных производителей часто различались, что приводило к несовместимости систем друг с другом. Подход **ITU-T**, исходящий из нужд сетевых элементов, оказался сложен в реализации и ограниченно востребован в бизнесе и практике.

Отказ производителей автоматизированных систем поддержки эксплуатационного управления от регламентированных протоколов и строгого соблюдения функциональной архитектуры привели к

появлению альтернативного направления построения систем поддержки эксплуатации OperationsSupportSystems (**OSS**), в основе которого лежала возможность самостоятельного выбора применяемых протоколов и функций, реализуемых в решении. Под этим понимается, что разработчики вправе самостоятельно разрабатывать архитектуру, функции, которые она выполняет, и протоколы взаимодействия между компонентами внутри системы и вне ее. Чтобы такой подход не приводил к проблемам при интеграции систем от разных разработчиков, необходимо решить ряд задач, связанных с разработкой общего подхода, или стиля проектирования и реализации систем поддержки эксплуатации, который обеспечил бы их совместимость. В существовании подобного «стиля» ведения разработки заинтересованы практически все участники телекоммуникационного рынка, начиная от производителей телекоммуникационного оборудования и заканчивая Операторами связи. Инициатива разработки подобного «стиля» принадлежит Форуму управления телекоммуникациями – TeleManagementForum (**TMF**). Одна из основных разработок Форума TMF – концепция **NGOSS**, в рамках которой предполагается развитие универсальной методологии разработки программного обеспечения.

4 Концепция **NGOSS**

Перед специалистами форума TMF стояла непростая задача: предложить универсальный способ разработки систем **OSS**, с помощью которого можно было бы достичь следующих целей:

- Обеспечить высокую скорость разработки новых услуг связи и вывод их на рынок. Сделать этот процесс менее затратным.
- Оптимизировать внутреннюю деятельность Операторов связи и помочь им на пути к становлению “leanoperator” – экономными Операторами, то есть Операторами, максимально сократившими свои внутренние издержки на ведение бизнеса.
- Обеспечить в **OSS**-системах сквозную автоматизацию бизнес-процессов.
- Гарантировать заданное качество обслуживания на неоднородных, комплексных сетях;
- Управлять мультивендорными (от многих производителей) и мультитехнологическими сетями связи.
- Минимизировать затраты на интеграцию **OSS**-компонентов разных производителей.

В рамках форума в работу были вовлечены представители ведущих компаний-разработчиков, Операторов связи, системных интеграторов, производителей оборудования и других заинтересованных сторон.

В процессе работы был проведен поиск и анализ лиц, ролей, целей и процессов, возникающих при разработке систем класса **OSS**. Целью анализа была универсальная методология разработки систем, возможности которых удовлетворяли бы перечисленным выше требованиям. Подход к разработке систем **OSS** нового поколения назвали **NGOSS** (NewGenerationOperationsSystemsandSoftware). В него включили две основные составляющие:

- методология и жизненный цикл разработки;
- инструменты разработки.

Методология и жизненный цикл **NGOSS** определяют лиц, заинтересованных в построении систем поддержки эксплуатации, их цели, процессы, в которых они участвуют, фазы разработки, информацию, которая используется в каждой из фаз, информацию, которая получается на выходе фаз и прочее.

Другими словами, методология и жизненный цикл **NGOSS** определяют порядок действий каждой стороны для того, чтобы построить систему класса **OSS** с такими качествами, которые устроили бы все заинтересованные стороны.

В ходе изучения информации, используемой в процессе разработки систем **OSS**, особое внимание уделялось тому, каким способом она представляется. Существенная разница в принципах описания информации о процессах, объектах управления и учета, приложениях приводила к неоднозначности в используемых терминах и к непониманию между сторонами, участвующими в разработке и внедрении систем **OSS**. Поэтому, при разработке подхода **NGOSS**, внимание было уделено разработке универсальных способов, или «инструментов» описания информации, которые использовались бы в процессе разработки. По мере развития и использования этих инструментов изменялась и методология **NGOSS**, которая определяет: как, в какой последовательности, кто и при помощи какого инструмента выполняет отведенную ему задачу в процессе разработки.

Вернемся к отправной точке методологии. Прежде чем пытаться оптимизировать какую-либо деятельность, необходимо определить, в чем именно эта деятельность заключается. Для этого нужно выявить список целей бизнеса в телекоммуникациях и бизнес-процессы, из которых данная деятельность состоит.

Под бизнес-процессом понимается последовательность взаимосвязанных действий, нацеленных на достижение одной или нескольких бизнес-целей.

Первоначально специалисты форума ограничивались только эксплуатационной деятельностью Оператора связи, и первая их попытка формализовать бизнес-процессы была выражена в карте процессов для Оператора связи – TelecomOperationsMap (**TOM**). Карта процессов **TOM** упорядочивала деятельность Оператора в рамках быстрых и рутинных процессов эксплуатации (например, смена тарифного плана или подключение новой услуги), однако с ее помощью не всегда получалось выделить непрерывные последовательности процессов, так как цели эксплуатации иногда лежат в области бизнеса, стратегии или управления предприятием (например, ввод на рынок новых услуг, управление персоналом и пр.). Потребность в сквозном описании бизнес-процесса, т.е. учитывающем все стадии бизнес-процесса, начиная от его инициирования и заканчивая получением его результата, привела к расширению карты процессов, в результате чего появилась расширенная карта процессов для Оператора связи – enhancedTelecomOperationsMap (**eTOM**). Это первый инструмент, определяемый в концепции **NGOSS**. С помощью карты **eTOM** Оператор может структурировать свои бизнес-процессы, определить дублирующиеся и недостающие, если таковые имеются. Данный инструмент бизнес-ориентирован, и поэтому с его помощью руководство Предприятия связи может построить стратегию или выделить процессы, которые надо автоматизировать с помощью информационной системы (т.е. поставить высокоуровневые требования Разработчику), но с помощью **eTOM** нельзя решить какие-либо более детальные и технологически-ориентированные вопросы.

Для того, чтобы обозначить объекты, участвующие в процессах карты **eTOM**, а также взаимосвязи между этими объектами, и определить, какой информацией оперируют бизнес-процессы, строится *информационная модель*.

Информационная модель — модель объекта, представленная в виде информации, описывающей существенные для данного рассмотрения параметры и переменные величины объекта, связи между ними, входы и выходы объекта.

При помощи модели можно обозначить правила формирования возможных выходных состояний объекта при изменении входных величин. Почему именно информационная модель? Для описания информации можно было бы использовать список объектов – глоссарий, либо сделать текстовое описание. В глоссарии крайне сложно задавать отношения объектов между собой, а текстовое описание может быть непоследовательным, хаотичным и содержать перекрывающиеся друг друга терминологии. Именно поэтому была выбрана информационная модель (или как её еще иногда называют – модель данных). Однако в области информационного моделирования наблюдалась та же ситуация, что и с бизнес-процессами: подходов к организации информационной модели было множество. Из-за отличия подходов к определению информации в продуктах различных разработчиков серьезно усложнялся процесс их интеграции друг с другом. В сложившихся условиях для сопряжения двух различных информационных моделей было необходимо разобраться в обеих моделях, выбрать эталонную, спроецировать вторую модель информации на эталонную и только после этого разработать конвертор, который позволит унифицировать язык общения между двумя системами. Появление третьей системы в данном случае обяжет разработчиков создавать еще один конвертор, для которого потребуется аналогичная процедура. Чтобы этого избежать, TeleManagementForum принял решение о разработке новой эталонной информационной модели. Несмотря на то, что уже существовали эталонные информационные модели (от **ITU-T**, **CommonInformationModel**), они достаточно редко использовались в программных продуктах. Это было связано с тем, что, несмотря на их сложность, уровень проработки моделей был низким. В итоге, при адаптации эталонных моделей к конкретному внедрению требовалась их существенная доработка, а это означало потери при раскрытии форматов данных в процессах интеграции.

Форум TMF сделал серьезный и очень ответственный шаг в этом направлении. Он предпринял попытку разработки новой эталонной модели данных, в которой задействованы все объекты и вся информация, которые могут участвовать в бизнес-процессах, определенных в модели **eTOM**, то есть специфических для отрасли телекоммуникаций. Модель получила название **SharedInformationandDatamodel (SID)** и стала следующим инструментом, определенным в методологии **NGOSS**. В процессе создания **SID** был произведен анализ основных информационных моделей, выбраны участки для заимствования и участки для дальнейшей проработки. Подчеркнем еще раз то, что использование общей модели данных становится особенно важным, когда дело доходит до интеграции продуктов от различных производителей. Возможность взаимодействия двух систем появляется только в том случае, если они оперируют одними и теми же понятиями. В противном случае возникают существенные проблемы, и требуется создание специального ПО, приводящего форматы данных к единому виду.

Следующий вопрос, решенный форумом, касался принципов построения архитектуры программного обеспечения, так как именно в архитектуре должна закладываться идея всеобщей интеграции систем различных производителей между собой. В результате появилась технологически нейтральная архитектура (**TechnologyNeutralArchitecture, TNA**), в основу которой

были заложены принципы сервис-ориентированной архитектуры. Именно в **TNA** определяются основные идеи, позволяющие при необходимости интегрировать различные по функциональному назначению системы, даже если их разработкой занимались разные производители.

В последние годы наблюдался значительный рост рынка программного обеспечения, направленного на автоматизацию эксплуатационных бизнес-процессов. Хаотическое расширение рынка потребовало стандартизации в данной области. Для этого, во-первых, требуется определить, каким образом классифицировать решения, а также стандартизировать их названия. Во-вторых, требуется определить, какие функции закладываются в каждое конкретное решение, чтобы можно было определить стандартные интерфейсы и в дальнейшем заметно упростить интеграцию. Для решения этих задач была разработана карта **TAM** – карта приложений в области телекоммуникаций. Разбиение приложений в карте **TAM** на определенные функциональные области базируется на бизнес-процессах, определенных в модели **еТОМ**.

Теперь рассмотрим подробнее жизненный цикл и методологию **NGOSS**, а после и каждый из вышеопределенных инструментов, чтобы можно было понять, как все эти инструменты используются совместно.

4.1 Методология и жизненный цикл

Методология и жизненный цикл (**Мижц**) описывают процессы разработки системы **OSS**, которые начинаются с постановки требований к системе **OSS** и заканчиваются ее развертыванием, коммерческой эксплуатацией и технической поддержкой.

Разработчики **NGOSS** представили жизненный цикл в виде четырех ракурсов, проекций, взглядов на разработку решения, отказавшись от введения жестко заданной последовательности шагов. Рассмотрим эту декомпозицию процесса разработки последовательно (см. рис. 5.1).

Первично имеем единую область задач и процессов, подлежащую декомпозиции.

Все процессы разделяются на две части – логическую (технологически независимую) и физическую (технологически зависимую); это горизонтальное разбиение (стрелка 1, рис 5.1.). Такая декомпозиция отделяет логическую составляющую разработки от конкретных физических реализаций и таким образом обеспечивает возможность повторного использования однажды построенной логической архитектуры. Отделив логику от непосредственной подготовки инфраструктуры и внедрения физического решения, мы упрощаем развитие и развертывание нескольких эквивалентных экземпляров реализации одного решения.

Следующим этапом декомпозиции является вертикальное разбиение процессов разработки – выделение плоскостей интересов Разработчика и Заказчика (стрелка 2, рис 5.1). Такое разделение процессов разработки обосновано тем, что у каждой стороны есть конкретные цели в физической и в логической плоскостях разработки. Для Заказчика в логической плоскости интерес представляют бизнес-требования к разработке системы (бизнес-процессы, бизнес-сущности, модели данных и пр.). В то же время, его волнуют вопросы развертывания решения, эксплуатации и технического обслуживания – уже в физической плоскости. С другой стороны, Разработчик в логической плоскости уделяет внимание моделированию решения, опираясь на требования Заказчика, а в физической плоскости – реализации данных моделей с использованием определенных технологий.

Таким образом, получается четыре квадранта (стрелка 3, рис 5.1) – четыре подхода, четыре проекции, четыре точки зрения на процесс разработки, каждая из которых представляет аспекты определения, проектирования, реализации и работы решения **NGOSS**.

Ø Проекция бизнеса

На этом этапе осуществляется постановка требований Заказчиком – на достаточно абстрактном уровне описывается, какой набор функций должно реализовать решение, которое желает в результате получить Заказчик. Основное внимание уделяется бизнесу и его составляющим: процессам, сущностям и взаимодействию между ними. **eTOM** и **SID** используются совместно, для того чтобы выявить и описать бизнес-процессы и информацию, которая должна взаимодействовать с данными процессами для успешного выполнения бизнес-требований. Описание происходит с помощью высокоуровневых, технологически независимых терминов.

Ø Проекция системной разработки

После постановки требований Разработчик принимается за создание решения – прежде всего строится архитектура системы, не зависящая от будущей физической реализации. Основное внимание уделяется объектам, их поведению и взаимодействию между ними. К этому этапу в первую очередь относится моделирование системных процессов и информации в технологически нейтральной манере. Для создания желаемого решения **OSS/BSS** используются и **SID**, и **eTOM**, и **TNA**.

Ø Проекция реализации

Непосредственно физическая реализация решения – технологически нейтральная система, разработанная на предыдущем этапе, привязывается к физической структуре. Основное внимание уделяется тому, как построить инфраструктуру (hardware), программное обеспечение под нее (software) и встроенные программные средства, которые подходили бы для спроектированной системы. Здесь используется специфический архитектурный стиль **NGOSS**, чтобы перейти от технологически нейтральных спецификаций системы к выбранной целевой архитектуре.

Ø Проекция внедрения

На этом этапе Разработчик передает окончательную версию решения Заказчику. Заказчик проводит тестирование и сверяет полученные результаты работы решения с требованиями, поставленными на начальном этапе. Основной акцент делается на мониторинг работы системы: необходимо убедиться, что ее поведение адекватно. Если требуется внести исправления или дополнить набор функций, реализуемых решением, процесс разработки вновь переходит на первый этап – постановку требований в Проекции бизнеса. Именно таким образом и формируется замкнутый жизненный цикл разработки решения согласно концепции **NGOSS**.

Ключ к пониманию жизненного цикла заключается в том, что процессы анализа бизнес-требований, выяснения системных требований, моделирования решения, подготовки инфраструктуры и внедрения решения составляют единый жизненный цикл – этот цикл может начинаться с бизнес-целей/бизнес-требований и заканчиваться эксплуатацией развернутого решения, используемого для поддержания и автоматизации заданных требований. Хотя продвижение по фазам жизненного цикла идет последовательно, следует учитывать, что через жизненный цикл можно двигаться разными путями. Этапы могут пересекаться, например, работа

над реализацией системных требований может проходить в то же самое время, что и анализ бизнес-требований.

Жизненный цикл **NGOSS** использует все инструменты, разработанные **TMF** и являющиеся частью программы **NGOSS**. Он использует карту бизнес-процессов **eTOM**, общую информационную модель **SID**, принципы построения архитектуры **TNA** для того, чтобы предоставить цельный бизнес-ориентированный подход к развертыванию **OSS** решения, который включает в себя определение требований, построение архитектуры системы, подготовку инфраструктуры и внедрение решения.

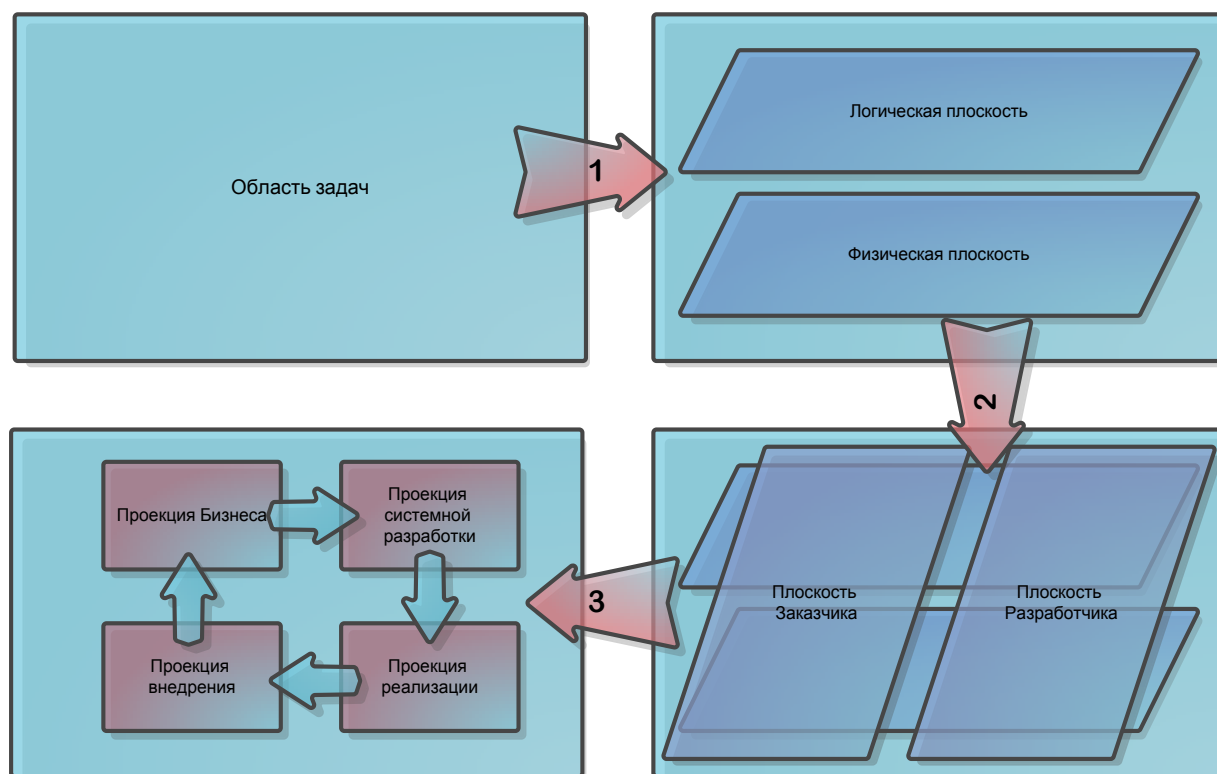


Рис. 5.1. Формирование жизненного цикла NGOSS

4.2 Модель бизнес-процессов eTOM

eTOM (enhancedTelecomOperationsMap) – расширенная карта бизнес-процессов Оператора связи – служит для определения бизнес-процессов внутри компании Оператора связи, а также может использоваться для определения процессов и точек взаимодействия с другими сторонами. Модель **eTOM** позволяет рассматривать деятельность Оператора с точки зрения бизнеса – выделить структуру/компоненты/взаимодействие бизнес-процессов – и, таким образом, позволяет поставить требования для разработчика **OSS**-решений с технологически нейтральной стороны. Можно сказать, что **eTOM** является «языком общения» Оператора и Поставщика **OSS**-решений на этапе общего описания требований к системе **OSS**.

Одним из условий разработки модели **eTOM**, равно как и других компонентов **NGOSS**, было удовлетворение потребностей максимального числа участников рынка телекоммуникаций и независимость от особенностей организационной структуры предприятия, используемых технологий и предоставляемых услуг. Именно поэтому модель обладает высоким уровнем

абстракции, но, в то же время, какие-либо ограничения её использования отсутствуют. Любой Оператор связи может самостоятельно определить свои бизнес-процессы на требуемом уровне детализации. Для достижения обозначенных условий, в модели **eTOM** предусмотрены уровни декомпозиции. Вместе с уровнем изменяется глубина детализации процессов. Всего было специфицировано четыре уровня декомпозиции: с нулевого по третий. Моделью **eTOM** подразумевается больше уровней декомпозиции, но уровни ниже третьего являются технологически-зависимыми и определяются уже при построении реальных процессов и систем поддержки эксплуатации. Нулевой уровень декомпозиции процессов **eTOM** представлен на рисунке 5.2. Он является самым абстрактным и представляет собой совокупность процессов, рассматриваемых на самом высоком уровне, анализ которых производится на уровне управления топ-менеджерами компании.

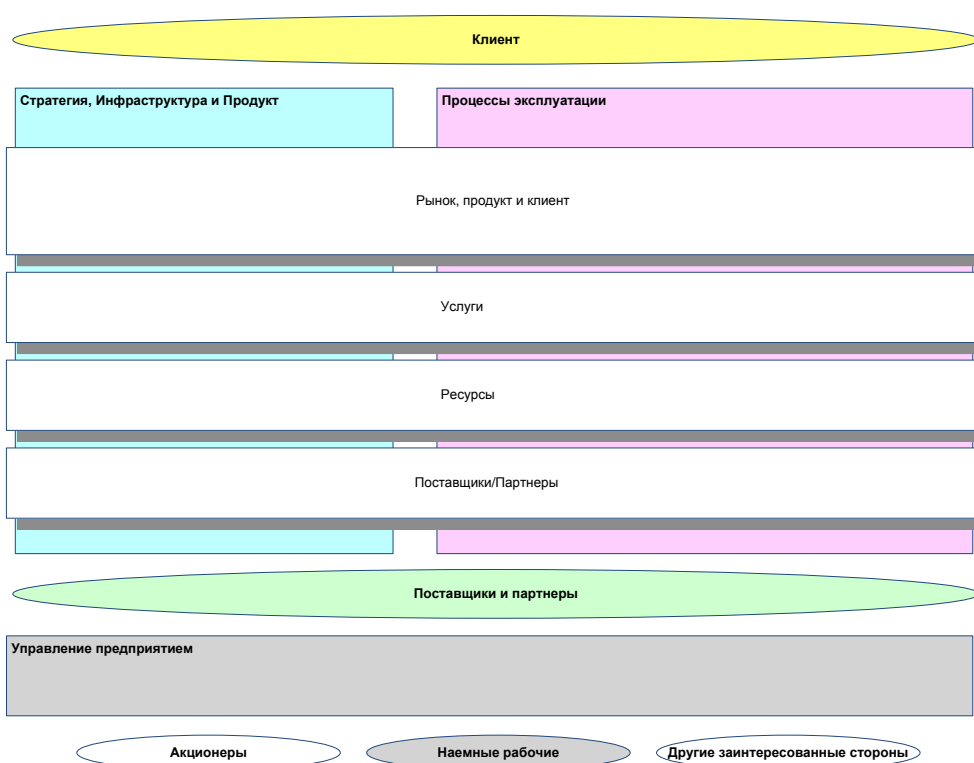


Рис. 5.2. Декомпозиция процессов eTOM уровня 0

На нулевом уровне декомпозиции определяются три крупных блока, в которые объединяются бизнес-процессы:



- Процессы эксплуатации – основная область для разработки **OSS**-решений. Объединяет процессы, связанные с технической эксплуатацией и взаимоотношениями с клиентами и поставщиками/партнерами.

- Стратегия, Инфраструктура и Продукт – область процессов, связанных с планированием стратегии и управлением жизненным циклом продукта и инфраструктуры.
- Управление предприятием – включает в себя процессы, отвечающие за поддержку управления предприятием и бизнесом и не зависящие от сферы деятельности компании.

Такое группирование обуславливается длиной жизненного цикла процессов, находящихся в той или иной области. Процессы, обладающие наиболее коротким жизненным циклом, собраны в группу процессов эксплуатации. Для демонстрации масштаба времени можно привести примеры процессов: «обслужить клиента», «выставить счет», «изменить тарифный план» и другие. Реализация таких процессов может занимать секунды, минуты, за редким исключением дни. Процессы построения стратегии и управления жизненным циклом инфраструктуры и продукта должны быть отделены от ежедневных, ежеминутных и ежесекундных процессов эксплуатации, так как они обладают большей длиной жизненного цикла. Именно этим и обуславливается появление второй группы – Стратегия, Инфраструктура и Продукт. Процессы из этой группы предоставляют также базу для управления и поддержки маркетинга, продажи и каналов поставок; именно они направляют и осуществляют поддержку процессов эксплуатации. Примерами таких процессов могут стать «Разработать новый продукт для выпуска его на рынок», «Принять решение об использовании новой инфраструктуры» и прочее. Реализация таких процессов занимает недели, месяцы, возможно годы. Последней выделенной группой стали процессы, затрагивающие Управление предприятием – они сосредоточены на постановке и достижении стратегических бизнес-целей компании, а также на поддержании двух вышеуказанных групп процессов. Обычно такие процессы рассматривают как некие корпоративные функции: управление рисками предприятия, управление финансами и активами, управление персоналом и пр.

На этом уровне декомпозиции также определяются все заинтересованные лица, имеющие отношение к бизнесу компании:



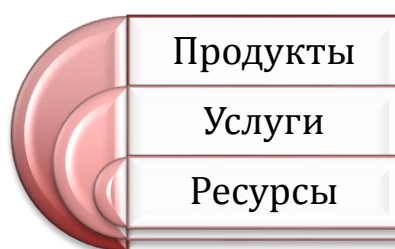
- Клиенты – те, кому предприятие продает свои продукты. Эти физические или юридические лица являются основой бизнеса любого Оператора связи.
- Поставщики/Партнеры – те, кто предоставляет предприятию свои ресурсы и/или напрямую или косвенно поддерживают бизнес компании.
- Наемные рабочие – это персонал, работающий на предприятии. С помощью работников предприятие может вести свой бизнес.
- Акционеры – владельцы предприятия, те, кто обладает акциями предприятия.
- Другие заинтересованные лица – другие участники сектора, которых может интересовать деятельность предприятия. Это могут быть государственные организации, СМИ, конкуренты.

Остановимся подробнее на модели Продукт-Услуга-Ресурс, которая используется в функциональном разделении процессов на горизонтальные уровни.

Продукт – это товар, который продает Оператор связи своим клиентам. Продукт состоит из набора услуг.

Услуга – это единичный(атомарный, т.е. неделимый) элемент, из которого строится предложение клиенту.

Т.е. в составе продукта должна быть как минимум одна услуга. Услуга, в свою очередь, базируется на ресурсах, или потребляет ресурсы, такие как, например: пропускная способность маршрутизатора, порт стойки DSLAM и т.д. Например, продукт «Телевидение IPTV» может базироваться на услугах: предоставление доступа к сети через ADSL, доступ к IPTV-серверу, и прочее. При этом, услуга «доступ через ADSL» занимает порт стойки DSLAM и пропускную способность входящего канала 2Мбит/с.



Такой подход к организации связей между ресурсами Оператора и его клиентами лежит в выделении горизонтальных функциональных областей. Для этого выделяются четыре слоя:



- Рынок, Продукт и Клиент – слой включает в себя процессы, связанные с управлением продажей, маркетингом, интерфейсом с клиентом, обработкой клиентских жалоб, управлением **SLA** и биллингом.
- Услуги – слой, включающий в себя процессы, связанные с разработкой, предоставлением, конфигурацией услуг, анализом их качества и оплатой их использования.
- Ресурсы – включает в себя процессы, связанные с разработкой и предоставлением инфраструктуры ресурсов, управлением, конфигурацией и проблемами, возникающими в ресурсах. Следует сразу отметить, что под ресурсами понимаются не только физические, но и логические ресурсы, т.е., например, не только количество портов у маршрутизатора, но и пропускная способность, обеспечиваемая каждым из них.
- Поставщики/Партнеры – слой, отвечающий за взаимодействие с поставщиками и партнерами. Это относится и к управлению каналами поставок, и к интерфейсу взаимодействия.

Структура **eTOM** создавалась таким образом, чтобы можно было определить и упорядочить все процессы Оператора связи. Иерархическая классификация процессов карты **eTOM** позволяет более детально рассматривать каждый из них.

Следующий за нулевым уровнем, первый уровень декомпозиции увеличивает детализацию процессов, проходящих на предприятии. В нем, помимо горизонтальных слоев, затрагивающих группы процессов эксплуатации и группы процессов Стратегии, Инфраструктуры и Продукта, появляется разбиение, отражающее вертикальные сквозные бизнес-процессы.

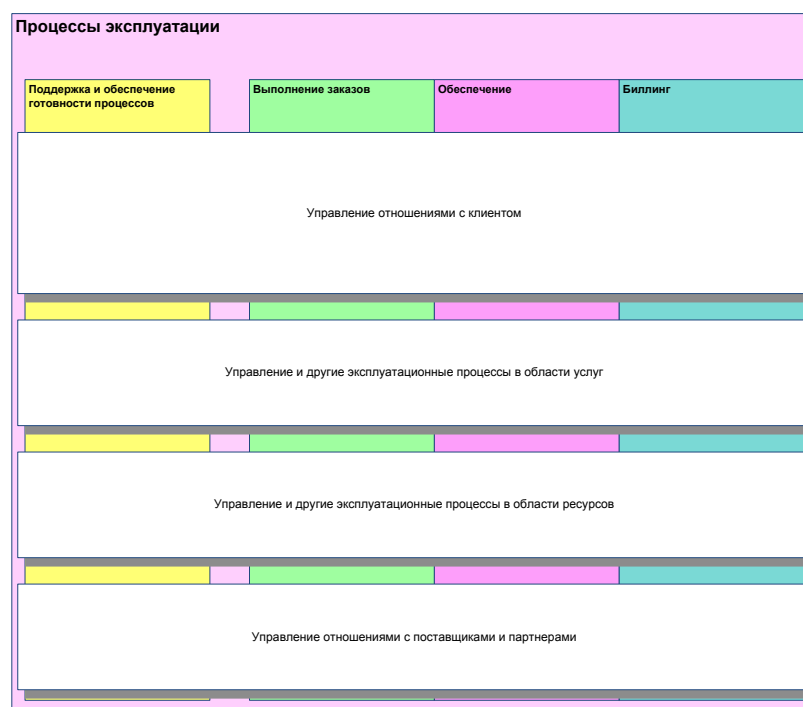


Рис. 5.3. Декомпозиция процессов блока «Процессы эксплуатации» до 1 уровня

В контексте данного материала мы не будем рассматривать декомпозицию первого уровня процессов Стратегии, Инфраструктуры и Продукта (**СИП/SIP**), а также процессов Управления Предприятием (**УП**), так как они имеют менее важное значение с точки зрения эксплуатационного управления, нежели группа процессов эксплуатации, которую рассмотрим более подробно. Стоит пояснить, что в рамках реализации процессов в области управления предприятием и стратегии лишь ставятся цели, которые в дальнейшем реализуются с помощью процессов эксплуатации. Так, например, управление жизненным циклом, входящее в состав области процессов **СИП**, включает в себя задачи, определяющие прохождение процессов в области **Эксплуатации**. Лишь отметим, что процессы вышеупомянутых областей **СИП** и **УП** хоть и проработаны в **еТОМ** вплоть до 3-го уровня декомпозиции, но все же относятся к другой предметной области, нежели к разработке **OSS**-решений (эти процессы ближе к таким системам как ERP, MRP, SCM, HRM и пр.).

Итак, перейдем к рассмотрению первого уровня декомпозиции Процессов Эксплуатации. Как было сказано ранее, с увеличением индекса уровня декомпозиции увеличивается детализация отраженных на карте процессов, и все четыре функциональных слоя, определенных на нулевом уровне, при рассмотрении на первом уровне декомпозиции приобретают более характерные для процессов эксплуатации черты; возникают слои:



- Управление отношениями с клиентами – собирает всю информацию о клиентах, об их потребностях; определяет все процессы, необходимые для поддержания имеющихся и выявления новых возможностей взаимодействия с клиентами. Так, например, в данном слое могут быть определены процессы управления аккаунтом через клиентский портал.
- Управление и другие эксплуатационные процессы в области услуг – фокусируются на предоставлении услуг и управлении ими, отдельно акцентируется внимание на качестве предоставляемых услуг. Так, в данном слое могут быть определены процессы контроля качества предоставляемых клиенту услуг, например, чтобы скорость подключения канала ADSL была не ниже указанной в договоре.
- Управление и другие эксплуатационные процессы в области ресурсов – учитываются все аспекты взаимодействия с ресурсами (инсталляция, конфигурация, активация, тестирование и т.д.) для обеспечения необходимого качества предоставляемых услуг.
- Управление отношениями с поставщиками и партнерами – объединяет все процессы, связанные с взаимодействием с поставщиками и партнерами. Так, в пределах этого слоя могут быть определены процессы взаимодействия с поставщиками оборудования, с которыми у Оператора заключен контракт. Или, например, могут быть оговорены процессы взаимодействия с партнерской компанией, которая предоставляет услуги дальней связи некоторому Оператору местной связи.

Как уже отмечалось, на первом уровне декомпозиции появляется вертикальное разбиение:



- Выполнение заказов (Fulfillment) – процессы этой группы отвечают за своевременное и правильное выполнение заказов и обеспечение клиентов продуктами согласно произведенным заказам. Здесь также находятся процессы, отвечающие за информирование клиента о продвижении его заказа и сведения об удовлетворенности клиента взаимодействием с компанией Оператора связи.
- Обеспечение качества (Assurance) - эта группа обеспечивает непрерывное наблюдение за качеством услуг и техническое обслуживание с целью предоставления клиентам услуг требуемого качества. Процессы этой группы бизнес-процессов гарантируют соблюдение соглашения об уровне обслуживания (SLA) и сообщают клиентам о работе услуги. Эта группа также обрабатывает жалобы клиентов, информирует их о возможных опасностях вследствие возникших проблем у Оператора и следит за тем, чтобы необходимые меры устранения возникшей проблемы были приняты вовремя.
- Биллинг (Billing) – эта группа процессов отвечает за предоставление клиентам счетов на оплату, получение и обработку счетов об оплате, обработку запросов от клиентов о состоянии счета и пр.
- Поддержка и Обеспечение Готовности (OperationsSupportandReadiness)– в этой группе собраны все процессы, направленные на обеспечение эффективного протекания

процессов групп «Выполнение заказов», «Обеспечение» и «Биллинг». Эти процессы, хоть и связаны непосредственно с предоставлением услуг в реальном времени, но, тем не менее, не выходят прямо на клиента, а остаются «на заднем плане». Таким процессом может быть, например, управление доступностью и готовностью клиентского интерфейса.

Последняя группа – Поддержка и Обеспечение Готовности – в силу своей специфики отделена от трех других, изображаемых в виде единого блока, который называется **FAB** (Fulfillment, Assurance, Billing).

Следует отметить, что даже если не рассматривать **eTOM** как одно из вспомогательных средств для разработки **OSS** решений, с точки зрения людей, причастных бизнесу телекоммуникаций, карта все равно остается чрезвычайно полезным инструментом, который можно использовать для такого важного этапа разработки стратегии, как реструктуризация бизнес-процессов компании.

4.3 Модель общей информации и данных SID

4.3.1 Основы языка UML и его использование в SID

В данном разделе мы рассмотрим, каким образом язык **UML** (Unified Modeling Language) используется для описания модели данных **SID**.

Для многих программистов существует только мысленный процесс моделирования программного обеспечения, имеющий место на фазе реализации проекта. Для крупных проектов в сотни и тысячи человеко-часов отсутствие системного подхода к описанию данных абсолютно неприемлемо. Назначение **UML** – дать основу для этапа моделирования. **UML** – это язык, имеющий конструкции и символы, которые служат для визуализации, специфицирования, конструирования и документирования ПО. То обстоятельство, что **UML** базируется на открытых стандартах, позволяет создавать диаграммы, понятные всем лицам и организациям, с которыми приходится взаимодействовать при разработке.

Следует отметить, что задачей данного раздела является дать только базовое описание **UML** на том уровне, который требуется для общего понимания **SID** и умения им воспользоваться. Для желающих более подробно ознакомиться с языком **UML** в конце методического пособия будет представлен список литературы.

Сущности

Сущность – это общее формальное описание группы объектов, обладающих одинаковым набором характеристик (атрибутов).

Она представляет собой набор экземпляров одинакового типа (например, Сущность Маршрутизатор представляет собой именно идею маршрутизатора, а не какой-либо конкретный

маршрутизатор). Имя Сущности, как правило, представляет собой имя существительное в единственном числе, например Кабель, Клиент, АТС и пр.

Сущность может отображать как нечто физическое (например, мобильный телефон), так и идею (например, принадлежность какого-то объекта более крупной группе).

В **SID** Сущность изображается в виде прямоугольника, содержащего три секции (см. рис. 5.4). Иногда используется упрощенное отображение Сущности – просто прямоугольник.

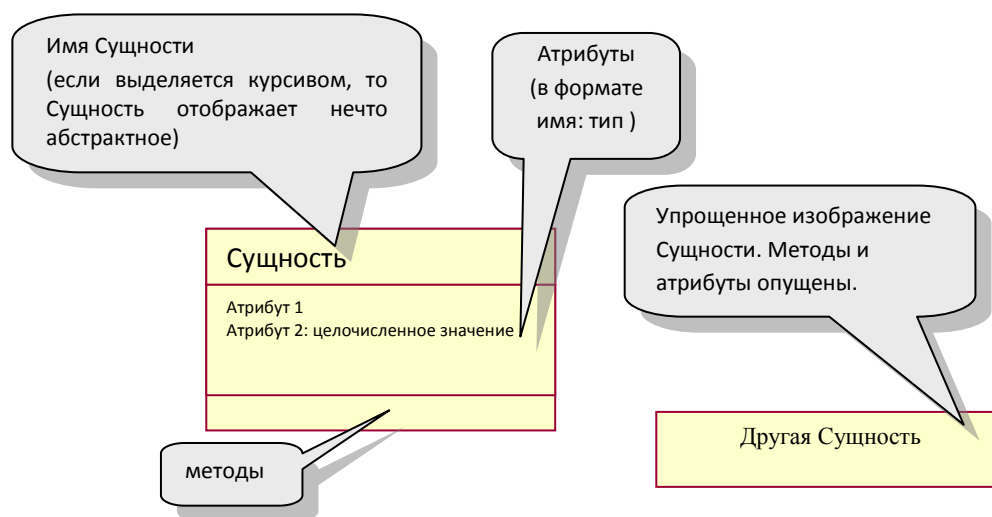


Рис. 5.4. Сущность

Атрибуты представляют собой характеристики Сущности (например, сущность Маршрутизатор может содержать в себе такие атрибуты как количество портов и дату введения в эксплуатацию). У атрибута также может быть определен тип (например, булевого(логического) типа (true или false), целочисленное значение, строка и пр.).

Сложные атрибуты могут быть выделены в отдельные Сущности; так, например, в **SID**. Существуют такие Сущности как Person и PersonName.

Ассоциации

Между Сущностями неизбежно существуют связи и отношения, иначе вместо информационной модели получится простой набор объектов и элементов. Такие связи называются Ассоциациями.

Ассоциации изображаются в виде сплошной линии, соединяющей две Сущности. Например, с помощью ассоциации можно показать зависимость «Клиент *делает заказ на* Продукт».

В случае, когда у Ассоциации есть собственные отдельные свойства, они отображаются в отдельной Сущности, которая соединяется штрихованной линией с этой Ассоциацией.

Имена Ассоциаций помогают нам понять, что именно они отображают. Имя, как правило, является глаголом, например «Тракт E1 *содержит* в себе временные интервалы».

На каждом из окончаний Ассоциации есть также пометки, отражающие количество экземпляров Сущности, которые могут быть задействованы в данном отношении.

Экземпляром Сущности называют конкретный объект из целого классаему подобных. Так, если определить Сущность «Маршрутизатор», то конкретный четырехпортовый маршрутизатор Cisco с поддержкой VPN будет являться экземпляром Сущности «Маршрутизатор».

Ассоциация может быть дополнена Ролью, которую Сущность играет по отношению к другим Сущностям.

Чтобы лучше понять назначение Роли, рассмотрим, для примера, Сущность «Человек». В зависимости от того, с кем Человек общается, набор его свойств изменяется: он может быть Отцом, Директором, Покупателем, Налогоплательщиком – это его Роли.

Итак, каждой Ассоциации на одном из ее концов может быть назначена Роль. У одной сущности, таким образом, может быть несколько Ролей.

Роль часто обладает Кратностью. Кратность показывает, сколько объектов может быть связано посредством одного экземпляра Ассоциации. Для понимания назначения Роли, обладающей кратностью, можно привести следующий пример: «У родителя один или более детей, у каждого ребенка двое родителей». Кратность указывается на концах ассоциации.

В заключение рассмотрения Ассоциаций введем понятие Самообъединения.

Самообъединение – это ассоциация, которая соединяет класс Сущности с самим собой.

Примером Самообъединения может служить шасси в стойке, когда обе эти сущности входят в класс Holder, и этому классу разрешено содержать в себе другие сущности класса Holder.



Рис. 5.5. Ассоциация

Агрегация

Агрегация – это частный случай Ассоциации, отражающий более тесную взаимосвязь, чем обычная Ассоциация. Агрегация отмечает отношение «Целое»-«Часть» и обозначается ромбом на том конце Ассоциации, где находится Сущность «Целое», группирующая другие части.

Агрегацию можно соотнести с бизнес-правилом «X содержит в себе один или более Y, каждый Y является частью X».

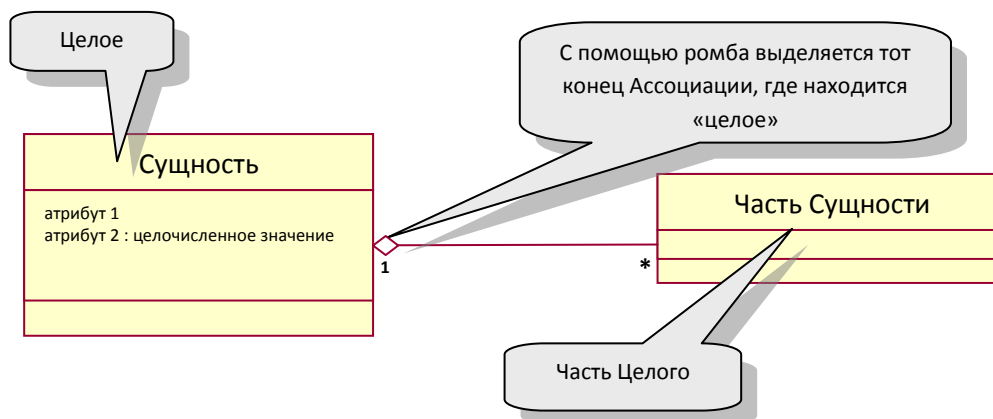


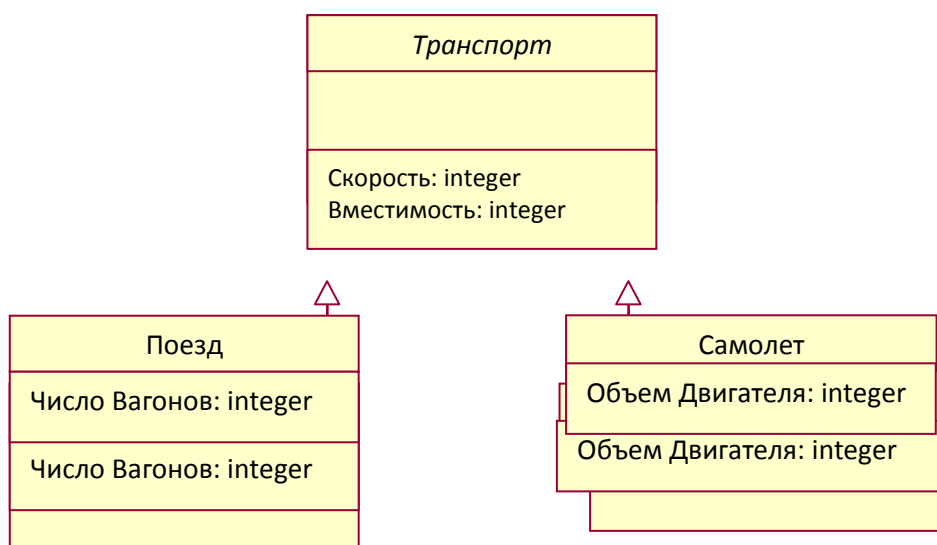
Рис. 5.6. Агрегация

Наследование

Наследование соотносит общие Сущности с более частными вариантами (специализациями). Так, например, можно сказать, что «Мобильный телефон» – это частный вариант Сущности «Телефон».

В **SID** часто родительские Сущности являются абстрактными (в этом случае имя сущности выделяется *курсивом*). Абстрактная сущность неполна, т.е. нельзя представить конкретный экземпляр этой Сущности – она существует с целью объединения других Сущностей и выделения у них общих признаков.

Разберем пример для иллюстрации понятия Абстрактной Сущности. Пусть имеется две Сущности – «Поезд» и «Самолет». Их можно рассматривать по отдельности, не объединяя никакими связями, и каждой из них присвоить свой набор атрибутов. Очевидно, что некоторые атрибуты могут повторяться – например, «Максимально развиваемая скорость», «Количество пассажирских мест». Поэтому целесообразно выделить для них родительскую Сущность «Транспорт» с атрибутами «Скорость» и «Количество мест».



Таким образом, нельзя создать экземпляр Сущности «Транспорт», т.к. он является абстрактным, неполным, и содержит только те атрибуты, которые являются общими для Поезда и Самолета, экземпляр которых создать можно. Сущности «Поезд» и «Самолет», помимо собственных атрибутов, обладают наследованными (скорость, количество мест).

Атрибуты родительской Сущности присутствуют у сущностей-потомков, но на UML-диаграммах они повторно не отмечаются.

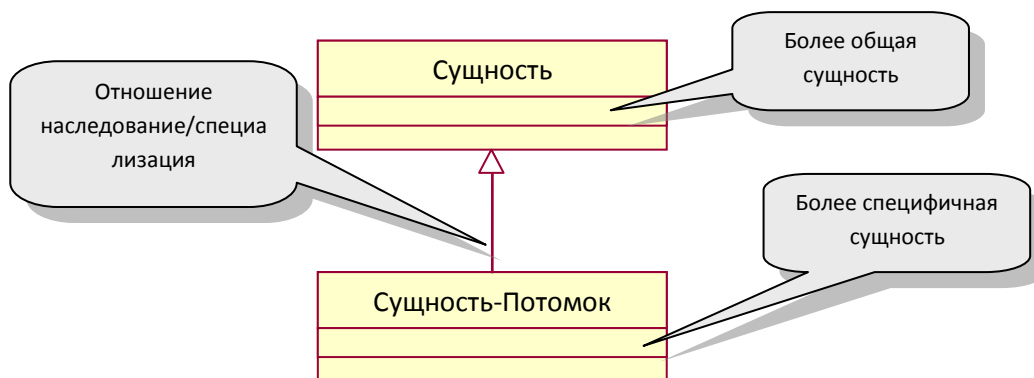


Рис. 5.7. Наследование

4.3.2 Концепция SID

SID представляет из себя общую информационную модель для участников рынка телекоммуникаций, независимую от платформы, языка и протокола реализации конкретной системы класса **OSS/BSS**. Для описания объектов и взаимосвязей между объектами, которые фигурируют в бизнес-процессах, необходим общий принцип определения и структурирования информации. Таким принципом является общая информационная модель.

Данные в **SID** иерархически структурированы. На самом верхнем уровне находится такое понятие как домен. Домен – это область управления. Домены соотносятся с концепцией **eTOM** на уровне 0. Всего было определено 8 доменов управления, семь из которых соотносятся с областями **eTOM** и содержат соответствующие этим областям управления объекты, а восьмой домен содержит общие объекты, которые используются во всех остальных областях управления.

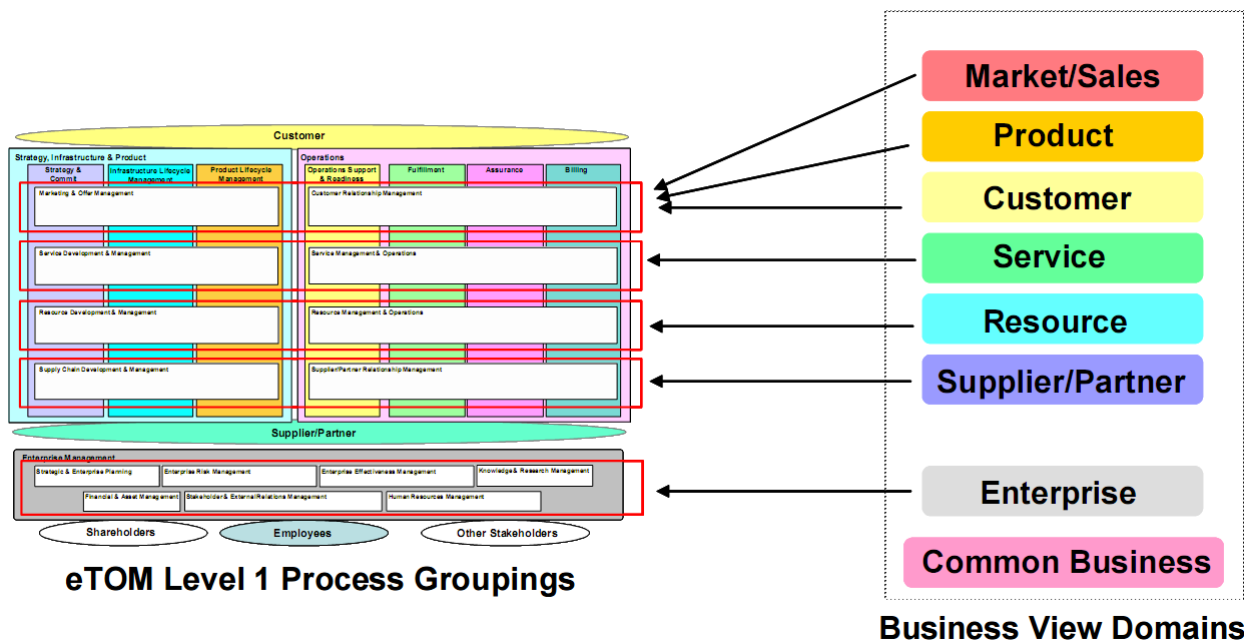


Рис. 5.8. Домены SID

Итак, 8 доменов **SID**:

- Рынок/Продажи;
- Продукт;
- Клиент;
- Услуга;
- Ресурс;
- Поставщик/Партнер;
- Предприятие;
- Общие Бизнес-Сущности.

Как видно из рисунка, домен Общих Бизнес-Сущностей не связан ни с одной из областей процессов **eTOM**. Это объясняется тем, что Сущности этого домена используются во множестве процессов **eTOM** и их нельзя отнести ни к одной конкретной области (например, такие Сущности как Местоположение (Location), Сторона (Party), Соглашение (Agreement) и др.). Остальные домены содержат в себе объекты, участвующие в бизнес-процессах соответствующих областей **eTOM**.

Следующим уровнем иерархии группирования объектов в модели данных **SID** является такое понятие как Агрегированная Бизнес Сущность – **АБС** (Aggregated Business Entities, **ABE**). **АБС** является детально определенным набором нескольких сущностей. В пределах каждого домена собраны несколько **АБС**, соотнесенных с конкретной областью управления.

Каким образом определялось, какие должны быть **АБС** в пределах каждого из доменов? Для этого была разработана отдельная методика категоризации **АБС**. Все категории **АБС** (независимо от домена) разбиты на две группы, каждая из которых в свою очередь разбита на отдельные категории:

- Managed Entity **ABE** (Управляемые сущности)
 - Strategy and plan (стратегия и план);
 - Managed Entity (управляемая сущность);
 - Managed Entity Specification (Спецификация управляемой сущности).
- Management Entity **ABE** (Управляющая сущность)
 - Interaction (взаимодействие) – взаимодействие с управляемой сущностью;
 - Configuration (конфигурация) – внутренняя структура управляемой сущности;
 - Performance (производительность) – измерение качества управляемой сущности;
 - Test (тестирование) – подразумевает опрос управляемой сущности с целью определения её состояния;
 - Trouble (неисправность, неполадка) – проблема, ассоциированная с управляемой сущностью. Сюда входят аварийная сигнализация (alarms), простои (outages) и аварии (faults);
 - Financial (финансы) – стоимость управляемой сущности;
 - Usage (срок использования) – период времени, в течение которого используется управляемая сущность.

Согласно данному принципу категоризации были определены следующие **АБС** (для доменов Рынок, Предприятие и Общие Бизнес-Сущности **АБС** категории еще не присвоены).

	ABE Category									
	Managed Entity ABE			Management Entity ABE						
	<i>Strategy and Plan</i>	<i>Managed Entity</i>	<i>Managed Entity Specification</i>	<i>Interaction</i>	<i>Configuration</i>	<i>Performance</i>	<i>Test</i>	<i>Trouble</i>	<i>Financial</i>	<i>Usage</i>
Customer		Customer	Customer SLA	Customer Interaction, Customer Bill Inquiry	Customer Order	Customer Statistic,		Customer Problem	Customer Bill Collection, Applied Customer Billing Rate	Customer Bill
Product	Strategic Product Portfolio Plan	Product	Product Specification		Product Offering	Product Performance			Product Price (Product Offering Level 2)	Product Usage Statistic
Service	Service Strategy and Plan	Service, Service Application	Service Specification		Service Configuration	Service Performance	Service Test	Service Trouble		Service Usage
Resource	Resource Strategy and Plan	Resource	Resource Specification	Resource Communication	Resource Configuration, Resource Topology	Resource Performance	Resource Test	Resource Trouble		Resource Usage
Supplier/ Partner	Supplier / Partner Plan	Supplier / Partner, Supplier Partner Product	S/P SLA	Supplier / Partner Interaction, S/P Bill Inquiry	S/P Order	Supplier / Partner Performance, S/P Statistic		S/P Problem	S/P Payment	

Рис. 5.9. Категоризация доменов SID

Пропуски в таблице подразумевают одно из двух: либо домен не имеет точного соответствия категориям модели, либо **АБС** для данной категории еще не определены.

Давайте рассмотрим конкретную **АБС** из домена Ресурс, с одноименным названием (Resource), а если быть точнее, то одну из его составляющих – Физический Ресурс.

В первую очередь определим ключевые сущности, необходимые для четкого представления любого физического ресурса, вне зависимости от технологий, на которых он базируется, или от компании производителя:

1. Оборудование - например, маршрутизаторы или коммутаторы (на данный момент модель физического ресурса в рамках модели **SID** нацелена в основном на оборудование,

построенное по блочному принципу, когда любое сложное оборудование можно представить как состоящее из определенного набора функциональных блоков).

2. Компоненты оборудования – например, плата (материнская, сетевая и т.д.) или физический порт.
3. Контейнеры оборудования – подразумевается, что существует оборудование, назначением которого является содержать в себе другое оборудование. Примером такого оборудования может быть блок или стойка.
4. Расположение оборудования – например, маршрутизатор № 32423423 находится в стойке AS899, в серверной № 2, по адресу Московский пр., д. 6.
5. Расположение различных физических элементов внутри оборудования – например, необходима возможность различать физические порты на разных сетевых картах в пределах одного маршрутизатора.
6. Вспомогательное оборудование – оборудование, необходимое для корректной работы основного оборудования, но не участвующее в реализации первичных функций основного оборудования. Например, для корректной работы и функционирования маршрутизатору требуется источник питания, но непосредственно в реализации основной функции маршрутизатора (прием и перенаправление пакетов) источник питания никоим образом не участвует. Следует отметить, что в представленной на рис. 5.10 упрощенной версии модели физического ресурса данные объекты не показаны.

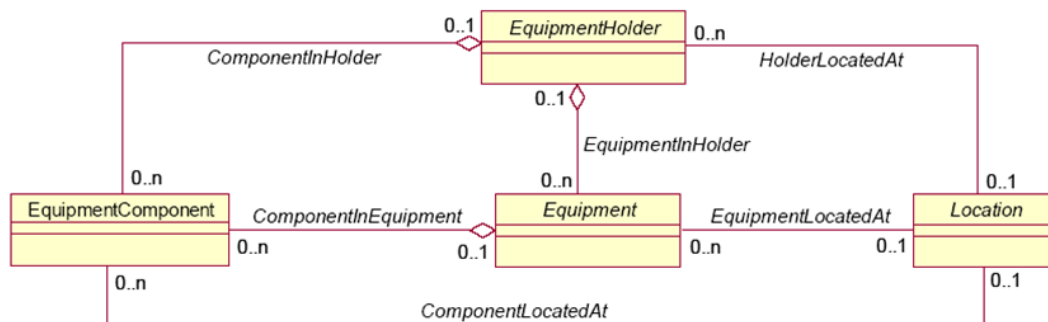


Рис. 5.10. Модель физического ресурса

С точки зрения описания какого-либо физического ресурса у нас должна быть возможность рассматривать его и как единое целое, и в то же время представлять, как оно сконструировано, из каких частей состоит. Например, у Маршрутизатора нас могут интересовать: число и тип плат, установленных на нем, какой разъем/слот они занимают, наличие резервного питания и другие не менее важные характеристики. Однако в контексте конфигурации сети все эти особенности конкретного маршрутизатора не представляют интереса, и важной остается лишь функция, которую он выполняет в сети – прием и перенаправление пакетов. Именно по этой причине введены две сущности – Equipment (оборудование) и EquipmentComponent (компонент оборудования) – и между ними существует зависимость в виде агрегации ComponentInEquipment. Значения на концах такой агрегации означают:

- От 0 до n со стороны сущности EquipmentComponent: «n» означает, что в состав оборудования может входить неопределенное число компонентов; «0» означает, что оборудование может быть цельным и вообще не состоять из компонентов;

- От 0 до 1 со стороны агрегирующей сущности Equipment: «1» отражает, что определенные компоненты могут входить в состав одного (и только одного) оборудования; «0» означает, что компоненты могут существовать отдельно от оборудования (например, резервные сетевые платы, еще не задействованные и находящиеся на складе).

Как было сказано выше, помимо оборудования и его составляющих также существует понятие «контейнер», то есть оборудование, которое содержит в себе другое оборудование (например, стойка, блок, штатив и т.д.). Такие объекты в модели **SID** представлены с помощью сущности EquipmentHolder. Как видно из рисунка, возможность нахождения оборудования в конкретном маршрутизаторе отображается с помощью агрегации EquipmentInHolder:

- от 0 до n со стороны сущности Equipment: «n» означает, что в одном контейнере может находиться неопределенное количество оборудования; «0» значит, что контейнер не содержит никакого оборудования (то есть в стойке может находиться либо n маршрутизаторов, либо ни одного, т.е. стойка пуста);
- от 0 до 1 со стороны сущности EquipmentHolder: «0» означает, что оборудование может существовать вне контейнера (например, маршрутизатор еще не успели установить в стойку); «1» значит, что оборудование содержится в данном контейнере.

Кроме этого, сам контейнер может состоять из компонентов. Например, у стойки может быть свой независимый блок питания, который будет являться её компонентом. Эта зависимость показана с помощью агрегации ComponentInHolder, которая аналогична двум рассмотренным выше.

Расположение различных физических элементов внутри других задается с помощью трех агрегаций: ComponentInEquipment, EquipmentInHolder и ComponentInHolder. Все остальные вопросы по месторасположению остаются на откуп сущности Location, которая входит в состав другого домена (Общие Бизнес Сущности) и нами рассматриваться не будет.

Одним из главных признаков хорошей модели является повышенная гибкость и такое свойство, как «reuse» («повторное использование») однажды разработанных шаблонов, с внесением в них минимальных изменений. Должны быть также предусмотрены механизмы быстрого расширения модели. В **SID** эти задачи решаются с помощью понятий Роль и Спецификация.

Спецификация – это способ определить инвариантные характеристики и поведение управляемой сущности. Так, например, если маршрутизаторы Cisco обладают какими-либо определенными характеристиками, отсутствующими у маршрутизаторов других производителей, то есть смысл выделить для них отдельную Спецификацию, в которую и будут помещены данные характеристики.

С помощью Роли мы можем выделить ряд функций, которые могут реализовывать устройства (например, Маршрутизатор, Коммутатор и т.д.).

В настоящей главе были рассмотрены лишь основополагающие принципы **SID**, более детальная информация может быть найдена в соответствующих рекомендациях TMForum, часть которых есть в открытом доступе на сайте www.tmforum.org.

4.4 Технологически-нейтральная архитектура TNA

Существует ряд требований к архитектуре системы, создающейся согласно концепции **NGOSS**. Все эти требования собраны вместе и представляют собой единую архитектуру для разработки систем управления. Далее мы рассмотрим основные положения технологически-нейтральной архитектуры (TechnologyNeutralArchitecture, **TNA**).

В **TNA** предусматривается компонентный подход к архитектуре. Любая система рассматривается как набор компонентов. Компоненты – это составляющие программного обеспечения, каждый из которых реализует определенные функции. Доступ к функциям компонента ПО предоставляется с помощью «Сервисов» (Services) или, как их еще называют, интерфейсов. Интерфейс – это описание и точка доступа к функциональным возможностям компонента (в частности, к атрибутам и действиям, которые можно запросить), к которым могут обратиться другие компоненты.

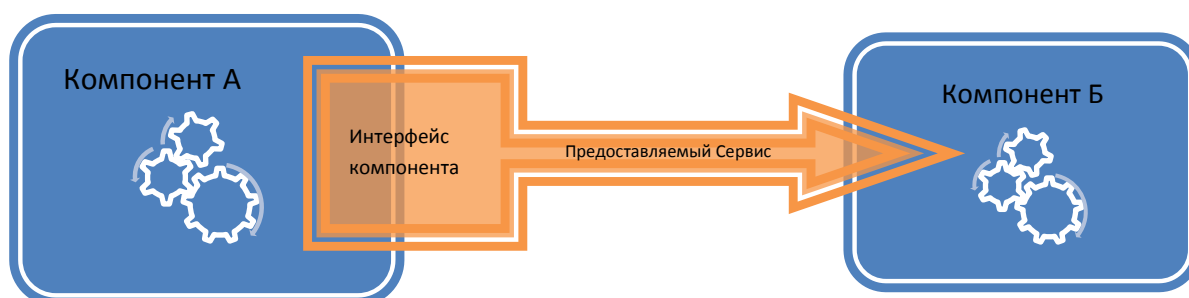


Рис. 5.11. Компонентный подход

Для описания предоставляемых компонентом Сервисов требуются:

- метаданные, необходимые для описания интерфейса;
- метаданные, необходимые для описания возможных действий над Сервисом;
- для каждого действия описываются результаты, которые могут возникнуть после реализации этого действия.

Метаданные, в общем случае, — это данные, характеризующие или поясняющие другие данные.

Для описания возможного поведения Сервисов необходимы:

- предусловия, при которых данное действие можно выполнить (т.е. набор условий, которые должны быть реализованы, чтобы действие могло быть произведено);
- постусловия, описывающие все состояния компонента, возможные после выполнения действия.

Концепция **NGOSS** предполагает следующие ограничения:

- Архитектура **NGOSS** не должна определять интерфейсы и сервисы внутри компонента;
- Она должна определять поведение компонента в целом и то, каким образом компонент делает свои сервисы доступными (через какие интерфейсы).

Существует ряд причин, обуславливающих переход от цельных, монолитных решений с широкими возможностями функциональной применимости, к специализированным модульным решениям, каждый компонент которых отвечает за свою узкую область функциональности и может входить в состав большой системы. К этим причинам относятся:

- Поддержка компонентов различных вендоров.

Это может заметно оптимизировать затраты при создании единого решения из различных компонентов, с возможностью использовать компоненты разных вендоров.

- Повторное использование компонентов.

Уже имеющиеся компоненты могут быть повторно использованы в различных ситуациях. Таким образом можно достичь того, что при разработке новой системы будет закупаться не цельное решение, а лишь будут докупаться нужные новые компоненты, тогда как некоторые можно выбрать из уже имеющихся в наличии.

- Взаимодействие с уже существующими системами.

Путем добавления специальных интерфейсов медиации (медиатор – посредник) можно наладить взаимодействие новых, добавляемых компонентов с уже существующими.

- Повышенная гибкость системы.

Когда сборка общего решения ведется на основе нескольких компонентов, у Оператора есть возможность выбрать лучшие решения для каждого типа задач, таким образом оптимизировать свои затраты и получить систему **OSS** именно с тем функционалом, который ему требуется.

4.5 Карта приложений TAM

Бизнес-процессов и информации, получаемых из **SID** и **eTOM**, теоретически вполне достаточно для описания деятельности Оператора связи. Но в реальности они (Операторы) располагают вполне конкретными системами. Одного понимания назначения блоков в карте **eTOM** им явnone достаточно – ведь совершенно не понятно, как увязать системы, задействованные в информационной инфраструктуре Оператора, и бизнес-процессы воедино. Некоторые системы участвуют в нескольких процессах одновременно. Таким образом, Операторам связи потребовался способ соотнести все те системы, которые уже задействованы в их инфраструктуре, с картой бизнес-процессов **eTOM**. Именно из-за этой потребности появилась карта **TAM**.

TAM представляет реальные приложения, предлагаемые на рынке **OSS** решений, и уже существующие системы внутри инфраструктуры Оператора, и помогает более рационально подойти к вопросу организации общего информационного пространства у Оператора.

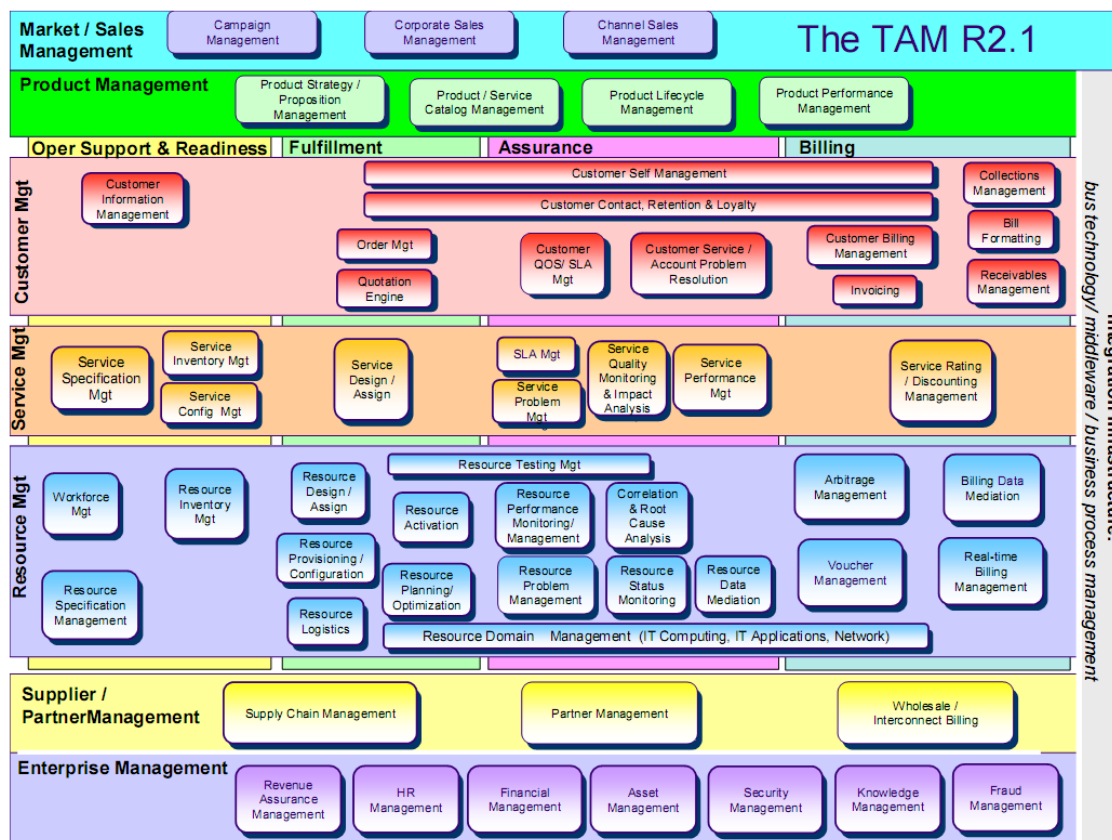


Рис. 5.12. Карта приложений TAM

Плюсы от использования TAM заключаются в том, что, во-первых, появляется некий стандарт для названий приложений. Во-вторых, для данных приложений формируются стандартизированные требования – четко оговаривается не только название приложения, но и некий набор функций, который оно должно реализовывать. И в-третьих, такая модульная структура построения инфраструктуры помогает более четко определить цели при внедрении систем класса **OSS**, заранее сформулировать где, между какими приложениями должны быть налажены интерфейсы, и определить, какие процессы эксплуатации явно «обделены» автоматизацией.

Давайте рассмотрим несколько примеров описания систем из **TAM**.

CustomerSelfManagementApplications

Приложения, предназначенные для самообслуживания клиентов, обеспечивают интерфейс для клиентов Оператора и позволяют им самостоятельно реализовывать часть функций, не привлекая персонал call-центра или офиса Оператора. Как правило, клиентский портал реализуется на веб-технологиях. Такого рода приложения весьма популярны как среди клиентов, так и среди Операторов, ведь они позволяют обеспечить доступ к информации 24 часа в день, 7 дней в

неделю и позволяют клиентам избежать длительного ожидания свободного оператора call-центра. Существует множество различных форм таких приложений, но в основной массе они представляют собой управляемое, обеспечивающее безопасность приложение, которое взаимодействует с системой **OSS** на стороне Оператора. Приложения, нацеленные на самообслуживание, должны обеспечивать высокий уровень безопасности как пользовательских данных, так и относительно интеграции с нижележащими системами. В плане функциональных возможностей такие приложения должны предоставлять полный набор функций самообслуживания, поддерживающих все стадии жизненного цикла заказа, включая:

- безопасность информации и аутентификацию пользователей;
- управление учетными записями клиентов (как для индивидуальных клиентов, так и для семей и предприятий);
- электронные продажи и управление заказами для широкого спектра коммуникационных продуктов и услуг;
- онлайн-поддержку решения проблем и выставление нарядов на ремонтно-восстановительные работы;
- электронное представление счетов/накладных и их анализ;
- активацию услуг.

Service Quality Monitoring & Impact Analysis Applications

Приложения мониторинга качества предоставляемых услуг и анализа неисправностей нацелены на то, чтобы предоставить Операторам возможность определить уровень качества предоставляемых клиентам услуг. В идеальном случае этот подход ориентируется на клиента и оценивается именно тот уровень качества услуг, который ощущает конечный потребитель, но также должны измеряться и те характеристики, по которым можно отследить ухудшения качества услуг и все вытекающие из этого проблемы. Анализ неисправностей, в свою очередь, необходим для прогнозирования возможного ухудшения качества услуг и его влияния на конкретных клиентов.

Такие приложения, как правило, поддерживают 4 подпроцесса, определенных в карте **eTOM**:



Рис. 5.13. Приложения Управления качеством предоставляемых услуг

Особо важны такие приложения с точки зрения предоставления информации другим приложениям, отвечающим за **SLA**.

- Мониторинг качества услуг
 - получение информации от разнообразных управляющих систем;
 - получение аварийных сигналов;
 - получение информации о тестировании.
- Анализ качества услуг
 - преобразование полученной информации в вид, пригодный для оценки качества услуг.
- Восстановление услуг
 - выдача рекомендации по улучшению качества на основе анализа качества предоставляемых услуг.
- Определение слабых мест в сети, где качество услуг ухудшается вследствие каких-либо ограничений.
- Уведомление об уровне качества предоставляемых услуг.

Resource TestingManagementApplications

Приложения, отвечающие за тестирование сетевых ресурсов, следят за тем, чтобы ресурсы различных видов работали корректно. Эти приложения относятся к областям процессов «Обеспечение качества» (Assurance) и «Выполнение заказов» (Fulfillment) из карты **eTOM**. Так, например, в области «Выполнение заказов» такие приложения следят за корректным предоставлением услуги, а в области «Обеспечение качества» – отвечают за изолирование неисправностей и за их устранение.

Приведем некоторые возможности этого вида приложений:

- Управление процессами тестирования.
- Автоматическое и ручное инициирование тестирования.
- Управление жизненным циклом тестирования.
- Управление централизованным тестированием.
- Управление доступностью централизованного тестирования.
- Тестирование отдельных элементов и общее тестирование.
- Управление результатами тестирования.
- Пояснение результатов тестирования.

Модель приложений **TAM** является инструментом для высокоуровневого моделирования архитектуры системы **OSS**, структуризации существующих в инфраструктуре Оператора систем и выбора поставщиков решений **OSS** из множества, предлагаемого на рынке. **TAM** предлагает универсальную модель именования функций в системах поддержки эксплуатации, вводя в использование новый словарь терминов – названий подсистем **OSS**.

5 Интерфейсы NGOSS для интеграции компонентов OSS

5.1 Причины и подходы к интеграции компонентов OSS

Задача интеграции встает перед Оператором или Системным Интегратором, поскольку из-за обширности функциональной области систем **OSS** ни один из производителей не способен одинаково хорошо реализовать все приложения, обозначенные на карте **TAM**. Это приводит к тому, что у Оператора параллельно используются компоненты **OSS** разных производителей. Эти компоненты могут не знать друг о друге, или, наоборот, обмениваться информацией, если они были предварительно интегрированы друг с другом. Системы **OSS** различаются интерфейсами, функциональностью, моделями данных, что усложняет процесс интеграции, а если требуется «увязать» не две, а несколько разных систем, то интеграция может обернуться большими затратами. Выходом из подобной непростой ситуации может быть создание систем, которые будут оперировать общими данными на базе одинаковых информационных моделей, пользоваться функциями друг друга, взаимодействовать через стандартные интерфейсы.

Подобные стандартизирующие подходы описывает TMForum, предлагая открытые интерфейсы: **OSS/J**, **MTNM**, **MTOSI**. Каждый из этих интерфейсов имеет свою область применения. Так, **MTNM** используется при взаимодействии систем управления сетью с оборудованием, **MTOSI** и **OSS/J**— при взаимодействии между собой различных систем управления и их элементов. Рассмотрим эти интерфейсы подробнее.

5.2 Интерфейсы взаимодействия MTNM, MTOSI, OSS/J

5.2.1 OSS/J

Интерфейс приложения – это набор функций, процедур и методов, с помощью которых можно управлять деятельностью приложения извне.

Спецификация интерфейса приложения – это способ определить, на каком языке и о чем смогут разговаривать целевое и внешние приложения.

Одно из направлений TeleManagementForum – это разработка спецификаций открытых интерфейсов, которые могут применять разработчики систем поддержки эксплуатации для доступа к своим функциям. Унификация интерфейса к различным приложениям **OSS** разных производителей позволяет в дальнейшем упростить процесс интеграции разрозненных компонентов в единое решение для Оператора связи.

Разработка **OSS/J** интерфейсов заключается в том, чтобы сформулировать универсальные требования для доступа к функциям приложений, зная предметную область этих функций. Например, если известно, что приложение занимается обработкой заказов на услуги связи, то не надо изобретать интерфейс или способ доступа к управлению этими заказами через интерфейс – нужно взять готовый интерфейс **OSS/J**.

Разработка интерфейсов **OSS/J** ведется в рамках технологии Java и использует в качестве информационной основы направления программы **NGOSS: SID** и **TAM**.

С помощью модели приложений **TAM** определяются виды деятельности приложений **OSS**: хранение данных, обработка заказов, управление неисправностями и прочие. Таким образом, выделяется конечный список универсальных интерфейсов, которых должно быть достаточно для доступа к любым функциям компонентов **OSS** в пределах модели **TAM**.

Модель **SID** взята за основу информационной модели интерфейса **OSS/J**, что позволяет определить язык и смысловую нагрузку понятий, о которых будет вестись разговор между приложениями через интерфейс **OSS/J**.

На рисунке ниже приведены интерфейсы **OSS/J**. Зеленым отмечены те интерфейсы, которые уже доступны для использования. Оранжевым цветом отмечены интерфейсы, проработанные не до конца. А также представлены интерфейсы, разработка которых запланирована на будущее.

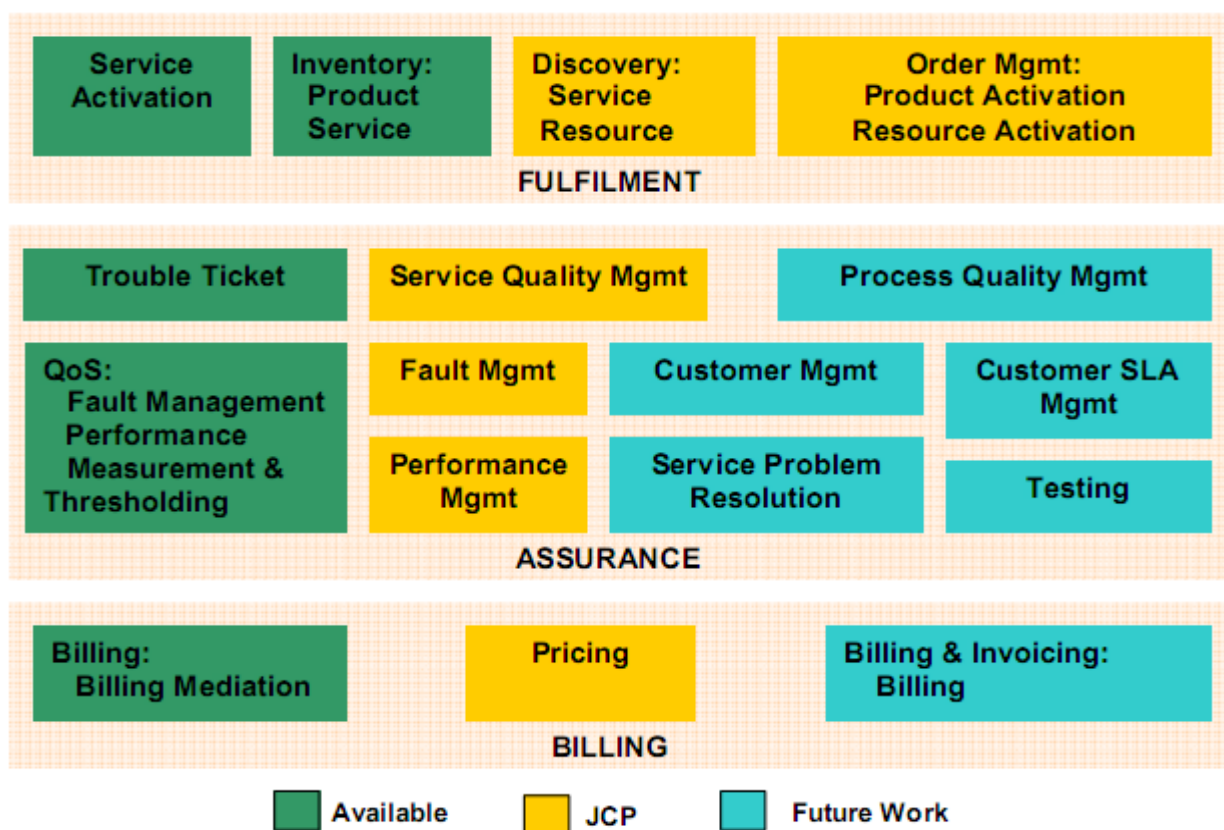


Рис. 6.1. Интерфейсы OSS/J

5.2.2 MTNM

Согласно логической уровневой архитектуре **TMN**, предложенной МСЭ (М.3010) и предназначенной для структурирования функций управления (архитектура **TMN** подробно рассмотрена в главе 4), эти функции объединяются в группы, именуемые *логическими уровнями*. Всего выделяется четыре таких уровня:

- Уровень управления бизнесом (BML, BusinessManagementLayer).
- Уровень управления услугами (SML, Service Management Layer).
- Уровень управления сетью (NML, Network Management Layer).
- Уровень управления сетевыми элементами (EML, ElementManagementLayer).
- Непосредственно саму сеть также можно представить как отдельный, нижний уровень этой иерархии — уровень сетевых элементов (NEL, NetworkElementLayer).

Набор функций, соответствующих каждому логическому уровню, может быть реализован в отдельной системе. Систему с набором функций уровня управления сетевыми элементами будем называть «система управления элементами» (ElementManagementSystem, **EMS**). Такая система позволяет конфигурировать элементы сети, производить мониторинг их состояния и производительности, обрабатывать аварийную сигнализацию. Как правило, **EMS**-системы разрабатываются непосредственно производителями телекоммуникационного оборудования и обслуживают лишь оборудование определенной технологии или определенного производителя, то есть позволяют управлять лишь *частью* сети Оператора.

Соответственно, систему, в которой реализованы функции уровня управления сетью, мы будем называть «система управления сетью» (NetworkManagementSystem, **NMS**). **NMS** позволяет обеспечить управление и мониторинг *сети в целом*, вне зависимости от технологий и протоколов передачи данных. Поэтому **NMS** обычно оперирует логическими ресурсами (например, каналами). Для управления сетью система **NMS** взаимодействует с системами нижележащего уровня EML. Таким образом, инициированные на уровнях управления бизнесом и услугами процессы через систему **NMS**, а затем через **EMS** доходят непосредственно до сетевого оборудования. Например, в процессе подключения/отключения услуг или мониторинга производительности сети оборудование конфигурируется или опрашивается.

При создании системы управления согласно принципам **TMN** встает проблема взаимодействия системы управления сетью (**NMS**) с нижележащими системами **EMS**. Большое количество **EMS**-систем разных производителей, с различающимися интерфейсами, функционалом, моделями данных становится для Оператора «узким местом» в автоматизированном предоставлении услуг и обслуживании сети. Поэтому, одним из залогов эффективной работы систем **NMS** является возможность их совместной работы с разнообразными **EMS**-системами в сети Оператора. А для этого **EMS** и **NMS** должны взаимодействовать через стандартизованный интерфейс посредством заранее оговоренных протоколов и использовать одинаковые модели данных.

Для решения проблемы унифицированного управления гетерогенными сетями TMForum разработал **MTNM**-интерфейс — Multi-TechnologyNetworkManagement. Интерфейс **MTNM** представляет собой открытый интерфейс для обмена информацией между системами управления сетью и системами управления элементами. Если уточнить, то **MTNM** — это реализация на основе **CORBA**¹ интерфейса между уровнями NML и EML логической архитектуры **TMN**.

Информация, передаваемая через **MTNM**-интерфейс, участвует в процессах областей *Выполнения заказов* и *Обеспечения качества* на уровнях *Услуг* и *Ресурсов* карты **eTOM**. Это процессы:

¹CORBA — это технологический стандарт написания распределённых приложений. Задача CORBA — обеспечить интеграцию изолированных систем, дать возможность программам, написанным на разных языках и работающим на разных узлах сети, взаимодействовать друг с другом так же просто, как если бы они находились в адресном пространстве одного процесса.

- конфигурации и активации услуг (*Service Configuration & Activation*);
- управления проблемами и услугами (*Service Problem Management*);
- предоставления ресурсов (*Resource Provisioning*);
- управления проблемами и ресурсами (*Resource Trouble Management*);
- управления производительностью ресурсов (*Resource Performance Management*);
- сбора и обработки данных от ресурсов (*Resource Data Collection & Processing*).

Описание информационной модели **MTNM**-интерфейса производится на языке **UML** и не привязано к какому-либо определенному языку реализации и протоколу обмена сообщениями. Чтобы лучше представить информационную модель интерфейса, приведем упрощенную UML-диаграмму классов **MTNM**-интерфейса:

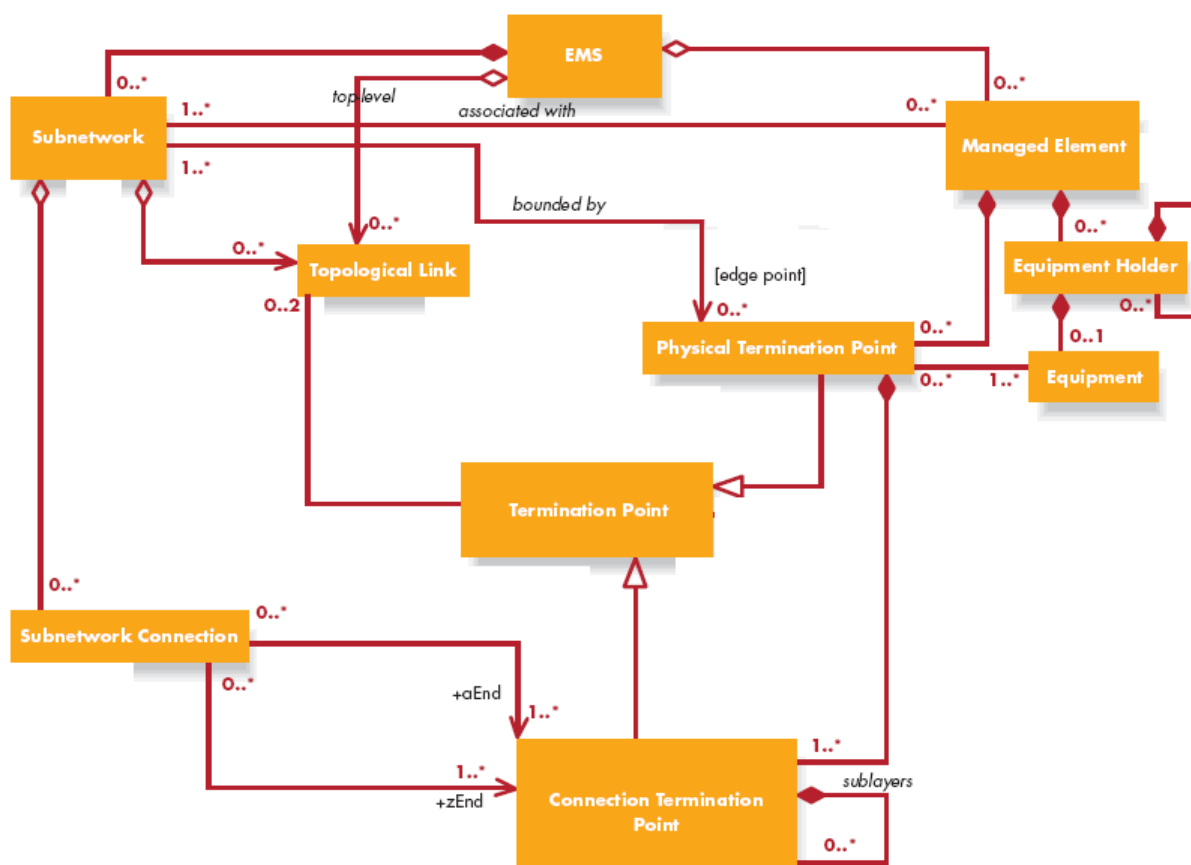


Рис. 6.2. Диаграмма классов MTNM

На UML-диаграмме указаны отношения между классами в информационной модели. Назначение этих классов и наиболее интересные их атрибуты сведены в таблицу:

Класс	Описание	Пример атрибутов	Общие атрибуты
EMS	Представляет абстракцию EMS, управляющей подсетью	<ul style="list-style-type: none"> • Версия ПО • Тип EMS 	<ul style="list-style-type: none"> • Имя • Пользовательская метка • Короткое имя EMS • Владелец • Дополнительная информация
Subnetwork	Подсеть. Содержит управляемые элементы (ManagedElement) и физические соединения между ними (TopologicalLink). Через подсеть возможно установление сквозного соединения — Subnetwork Connection	<ul style="list-style-type: none"> • Тип подсети • Поддерживаемые скорости 	
Managed Element (ME)	Управляемый элемент. Является отображением сетевого элемента (NE) для EMS	<ul style="list-style-type: none"> • Местонахождение • Версия ПО • Название продукта • Производитель • Поддерживаемые скорости 	
Topological Link (TL)	Элемент топологии сети — физическая связь между портами сетевых элементов	<ul style="list-style-type: none"> • Направление • Конечные точки • Скорость • Аварийная сигнализация 	
Termination Point (TP)	Сетевое окончание. Логическая абстракция, представляет собой конечную точку TopologicalLink или SubnetworkConnection	<ul style="list-style-type: none"> • Направление • Наличие точки мониторинга 	
Connection Termination Point (CTP)	Окончание логического соединения (SNC)	<ul style="list-style-type: none"> • Состояние соединения • Режим отображения 	
Subnetwork connection (SNC)	Транспортный маршрут данных через подсеть. Формируется как ассоциирование портов на границе подсети.	<ul style="list-style-type: none"> • Направление • Конечные точки • Скорость 	
Physical Termination Point (PTP)	Физическое окончание. Отображает окончание TopologicalLink. Фактически, является абстракцией порта сетевого элемента	<ul style="list-style-type: none"> • Направление • Наличие точки мониторинга 	
EquipmentHolder	Описание статов, стоек, слотов для оборудования	<ul style="list-style-type: none"> • Тип • Аварийная сигнализация • Установленное оборудование • Допустимое оборудование • Состояние • Местонахождение • Производитель 	
Equipment	Описание аппаратных средств элементов сети	<ul style="list-style-type: none"> • Состояние оборудования • Аварийная сигнализация • Номер оборудования • Производитель 	

Рис. 6.3. Классы MTNM

При моделировании сети за основу была взята рекомендация ITU-T G.805 — «Общая функциональная архитектура транспортных сетей», которая была существенно упрощена. Рассмотрим, как на базе информационной модели MTNM-интерфейса представляются сеть и сетевые элементы, описывается оборудование. На рисунке 6.4 показано, как будет представлена сеть в соответствии с информационной моделью MTNM. Система управления сетью (NMS)

обменивается информацией с EMS-системами через интерфейс MTNM. А взаимодействие непосредственно с сетевыми элементами каждой подсети осуществляет соответствующая EMS. Активные элементы сети представляются в информационной модели интерфейса как «управляемые элементы» — ManagedElement (ME). Порты сетевых элементов отображаются как «физические окончания» — PhysicalTerminationPoint (PTP). Порты могут быть физически связаны между собой. Эта связь будет представлена объектом класса Topological Link (TL). При установлении сквозного соединения через всю подсеть (SubnetworkConnection) это соединение будет ограничено точками-окончаниями (ConnectionTerminationPoint).

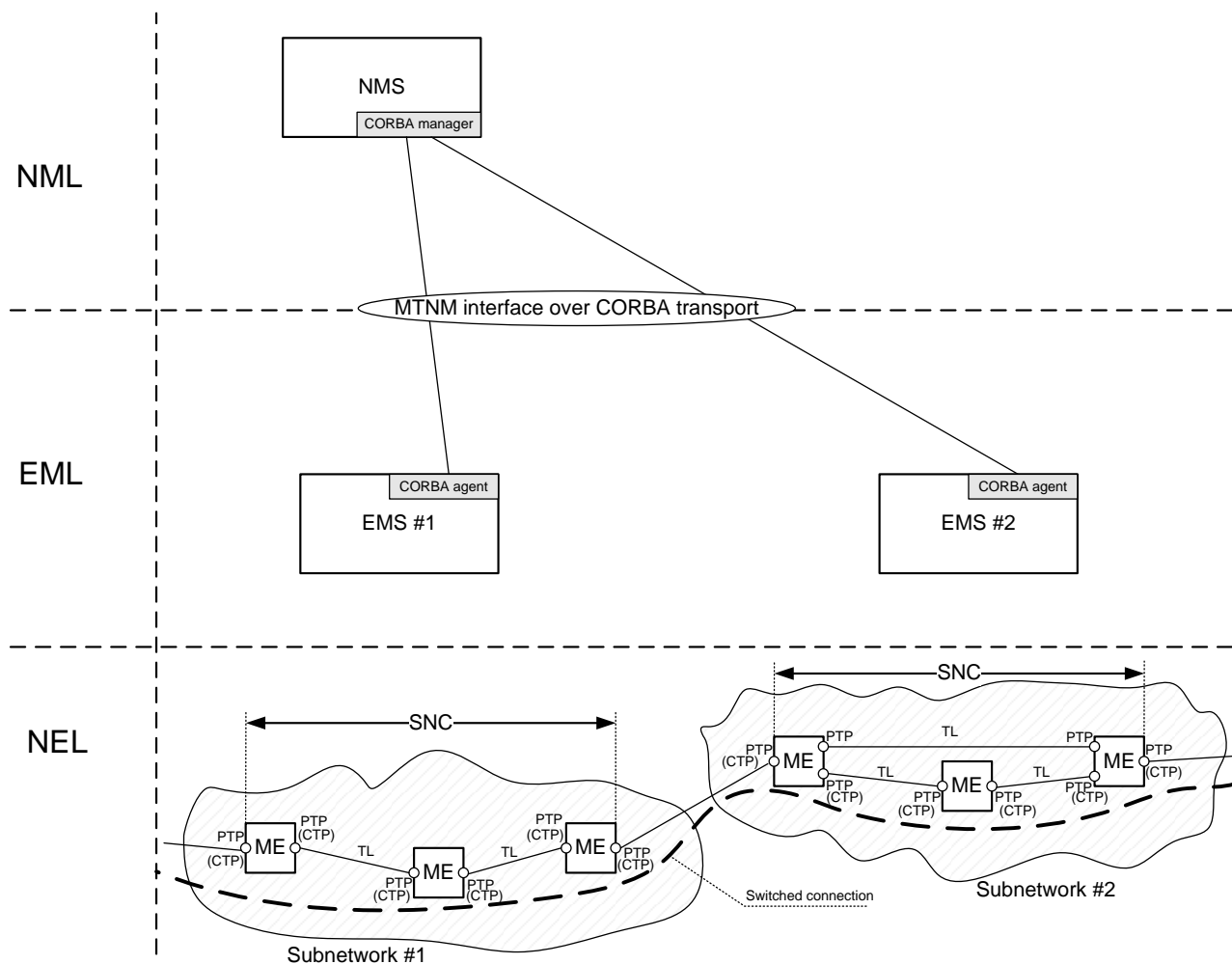


Рис. 6.4. Сеть в соответствии с MTNM

Интерфейс проработан для таких технологий сетей, как: WDM (технология уплотнения по длине волны в одном оптическом волокне), SDH/SONET (синхронная цифровая система передачи), ATM (пакетная асинхронная технология передачи данных), Ethernet (пакетная технология преимущественно локальных и городских компьютерных сетей), DSL (технология широкополосного доступа по медной паре).

Описание сети и сетевых элементов согласно приведенной информационной модели позволяет системе управления сетью (NMS) единообразно управлять гетерогенными сетями: конфигурировать сетевые элементы (NetworkElement, NE) и EMS-системы, исследовать сеть,

управлять соединениями, управлять учетом оборудования, осуществлять управление и мониторинг производительности, обрабатывать аварийную сигнализацию.

Через интерфейс **MTNM** можно видеть сеть не только с точки зрения управляемых элементов сети (**ME**) и связей между ними (**TL**), но и представлять элементы сети как физическое оборудование, и описывать его местоположение вплоть до платы. Для учета оборудования используется следующая модель: сетевой элемент представляет собой набор плат (**Equipment**) которые находятся в одном или нескольких, возможно вложенных, держателях оборудования (**EquipmentHolder**). Как видно из UML-диаграммы, имеется возможность строить иерархические модели учета оборудования — то есть объект класса **Holder** может содержать в себе такой же объект. Этот принцип наглядно представлен ниже:

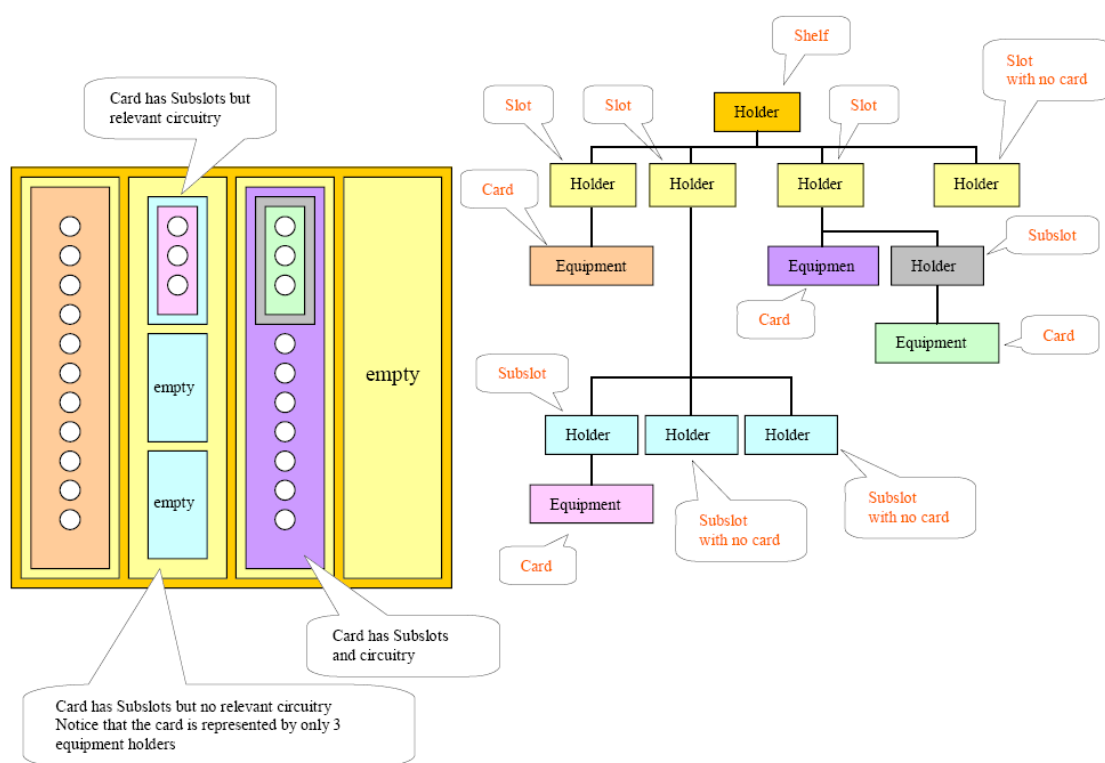


Рис. 6.5. Иерархическая модель учета оборудования

На рисунке 6.5 приведен пример использования информационной модели MTNM для учета оборудования. Оборудование (**Equipment**) размещается в контейнерах для оборудования — **EquipmentHolder**, или просто **Holder**. **Holder** может быть пустым, или содержать в себе оборудование. В примере показан стеллаж (объект класса **Holder** с атрибутом «тип держателя» равным **Shelf**). Стеллаж имеет четыре слота (объекты класса **Holder** с атрибутом «тип держателя» равным **Slot**). Первый (левый) слот полностью занят оборудованием. Второй имеет три подслота (объекты класса **Holder** с атрибутом «тип держателя» равным **Subslot**). В одном из трех подслотов размещено оборудование, а оставшиеся два — пустые. В третий слот установлено оборудование, которое само имеет разъем для установки карт-расширений. В четвертый (правый) слот оборудование не установлено.

5.2.3 MTOSI

При создании интерфейса **MTOSI** (Multi-Technology Operations System Interface) целью **TMF** было описание унифицированного открытого интерфейса для использования между операционными системами (OS, Operations System). Причем здесь под OS понимается любая система уровня управления элементами (EML), управления сетью (NML) и/или уровня управления услугами (SML) логической архитектуры **TMN**. В терминах карты **eTOM**, интерфейс обеспечивает передачу информации, используемой в процессах областей Выполнения заказов и Обеспечения качества на уровнях Услуг и Ресурсов карты **eTOM**. Рисунок 6.6 поможет лучше представить себе место интерфейса **MTOSI** в системе управления Оператора.

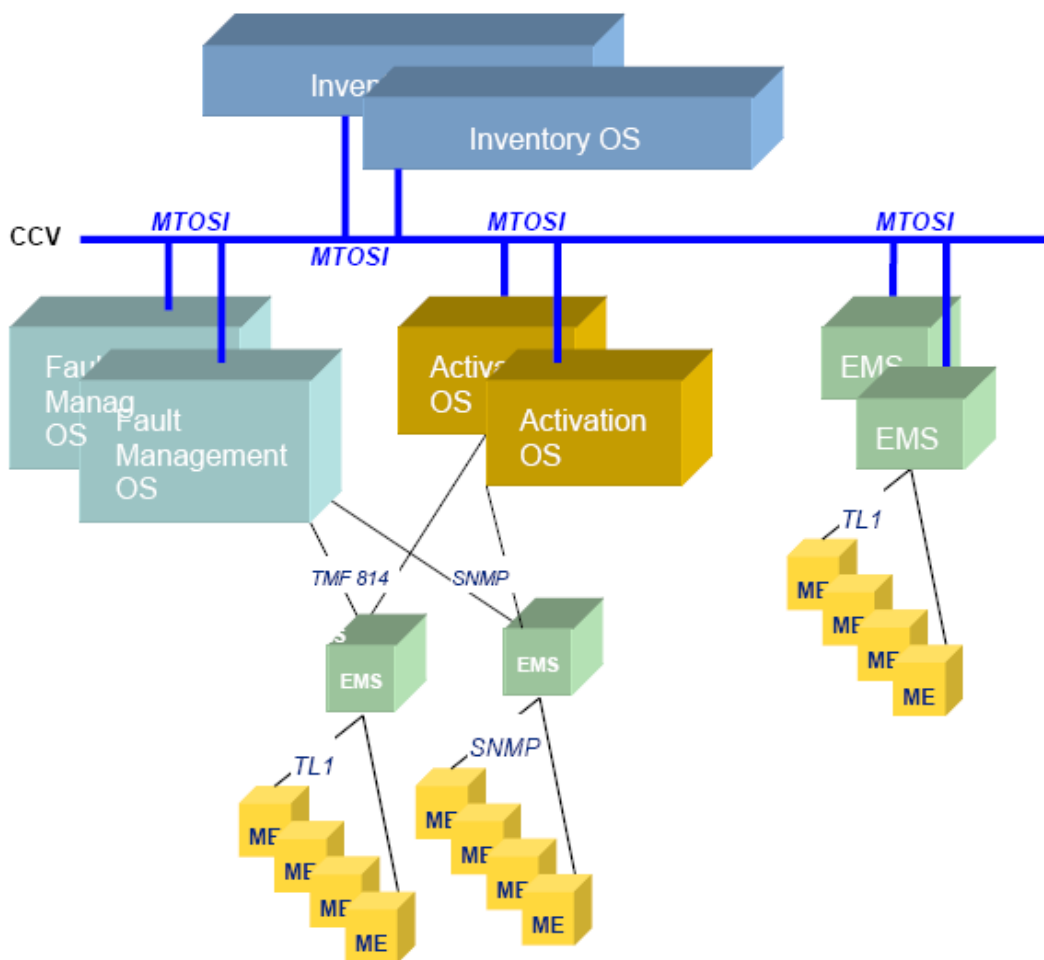


Рис. 6.6. Место интерфейса MTOSI в системе управления Оператора

Важно отметить, что MTOSI основан на той же информационной модели (TMF 608), что и интерфейс MTNM. Таким образом, абстракция сети, модель учета оборудования, обработка аварийной сигнализации в MTOSI точно такие же, как и в MTNM, что обеспечивает совместимость между интерфейсами на логическом уровне. Однако информационная модель MTOSI расширена с учетом специфики использования интерфейса — для обеспечения взаимодействия между несколькими операционными системами (OS) через общую среду (шину). Дополнены механизмы уведомлений и групповых операций с оборудованием, об изменениях в сети, об ошибках.

Коренным различием между **MTNM** и **MTOSI** является то, что транспортом для сообщений **MTOSI** могут являться JMS (JavaMessageService), HTTP, SOAP в режиме общей логической шины, в то время как **MTNM**, как правило, реализуется на CORBA, то есть в клиент-серверной архитектуре. Поэтому **MTNM** поддерживает только режим общения точка-точка (**NMS—EMS**), в то время как **MTOSI** может работать в режиме общей шины (на рисунке —CommonCommunicationVehicle, **CCV**).

Фактически, интерфейс **MTOSI** — это расширение возможностей **MTNM** с точки зрения места его применения интерфейса. Если **MTNM** ориентирован на использование исключительно в режиме «одна **NMS** — много **EMS**», то **MTOSI** не имеет таких ограничений и используется при обмене информацией между системами уровней EML, NML и SML.

6 Решения OSS АРГУС

Занятия практического курса базируются на программном обеспечении компании НТЦ АРГУС. В этом разделе мы дадим краткую характеристику используемых продуктов и укажем на связь их функций с приложениями и информационной моделью, используемой в **NGOSS**.

6.1 О продуктах АРГУС

Продукты **АРГУС** предназначены для оптимизации и комплексной автоматизации эксплуатационных процессов Оператора связи. Смена парадигмы ведения бизнеса Операторов связи приводит к постепенному переходу от продажи ресурсов к продаже услуг, реализуемых на ресурсах. Продукты **АРГУС** фокусируются на поддержке такого перехода для Оператора связи, не желая уступить конкурентные преимущества своим соперникам на рынке.

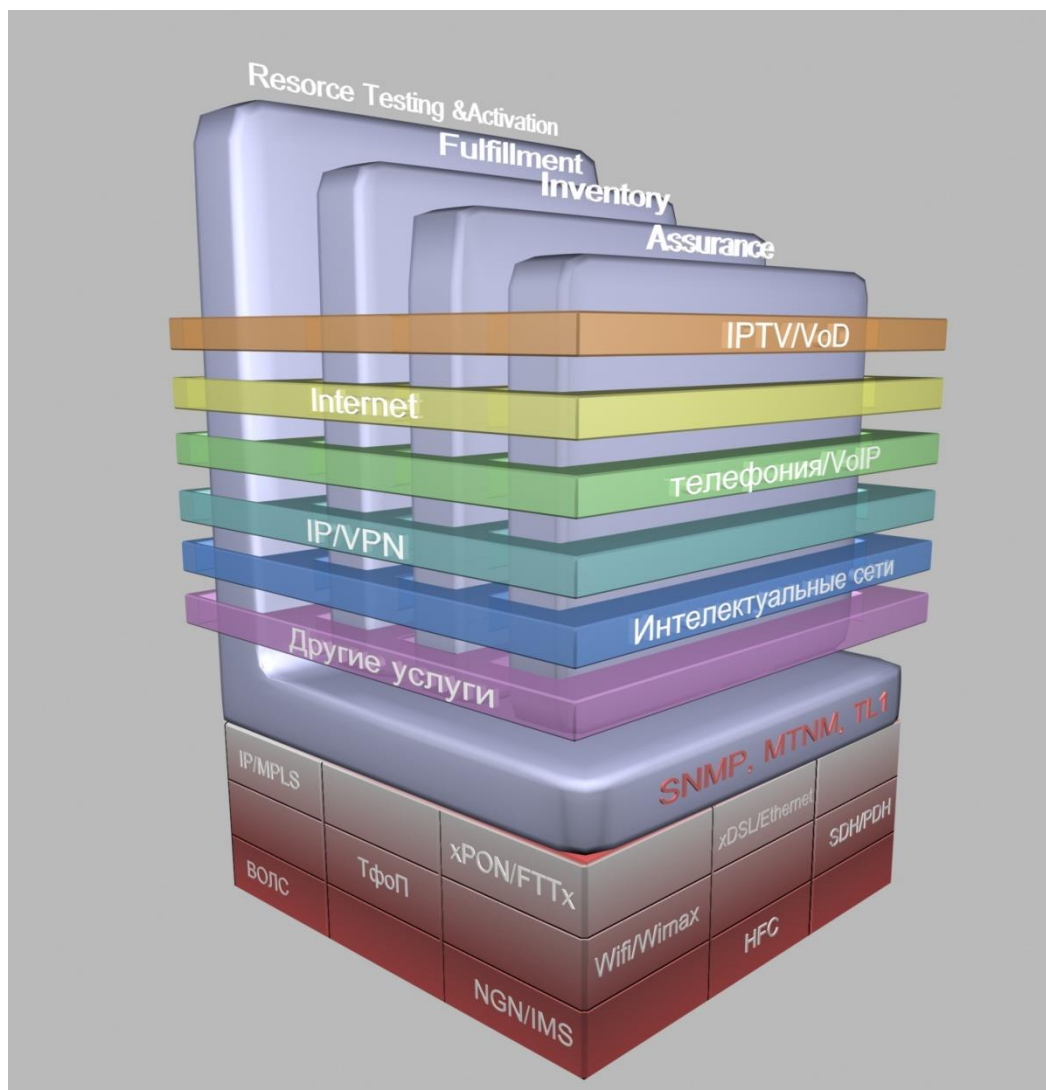


Рис. 7.1. Сквозная поддержка услуг продуктами Аргус

Системы **АРГУС** обеспечивают эффективную поддержку процессов предоставления и обеспечения качества услуг в любой современной операторской сети вне зависимости от сложности технологической и аппаратной базы. Каждая из систем отвечает за свою область эксплуатационных процессов:

- Resource Testing and Activation (Взаимодействие с оборудованием) – система **СИРИУС** – отвечает за воздействие на активное оборудование. Система поддерживает все современные стандарты в области управления гетерогенными сетями, такие как MTNM (multi-technology network management), SNMP (simple network management protocol) и пр.
- Fulfillment и CRM (управление отношениями с клиентами и выполнение заказов на услуги) – система **Абонентский Отдел (АО)** – отвечает за своевременное и правильное выполнение заказов и обеспечение клиентов продуктами (продукт – это комплекс услуг, представленный как маркетинговое предложение) согласно произведенным заказам.
- Inventory – система **АРГУС – Технический Учет (АРГУС-ТУ)** – отвечает за функции инвентаризации и учета клиентов, услуг и ресурсов.

- Assurance – система **Комплексная Техническая Поддержка (КТП)** – отвечает за постоянный контроль качества услуг и техническое обслуживание.
- ProductLifecycleManagement (Управление жизненным циклом продуктов) – система **ProductLab** – отвечает за процессы, связанные с жизненным циклом телекоммуникационных продуктов: создание новых продуктов, выпуск их на рынок, управление эффективностью продуктов и их вывод с рынка в случае снижения эффективности.
- BusinessIntelligence (Бизнес-анализ) – система **Аналитика** – позволяет руководителям компании принимать взвешенные и своевременные бизнес-решения на основании анализа данных, полученных от других систем, с использованием аналитических инструментов и методов (расчет KPIs, rootcauseanalysis, datamining, выявление трендов и построение прогнозов).

Продукты **АРГУС** изначально ориентировались на реализацию требований, специфических для российского Оператора связи. Каждая из систем создавалась, исходя из индивидуальных потребностей российских Операторов связи. При этом разработка велась эволюционно, и работа с каждым внедрением приносила что-то новое в общую архитектуру. Эта особенность развития объясняет универсальность применения продуктов **АРГУС** на российских сетях, которая, с одной стороны, способна удовлетворить любые уникальные пожелания заказчика, а с другой – предоставить готовые решения «наболевших» проблем Оператора связи. Системы могут выступать как комплекс, но каждая из систем может быть внедрена как отдельная составляющая стороннего комплекса приложений **OSS**. Вопросы интеграции приложений особенно важны для современного рынка систем поддержки эксплуатации, поэтому НТЦ АРГУС уделяет первостепенное внимание поддержке стандартов и требований TeleManagement Forum к **OSS**-решениям, таких как **eTOM, SID, TNA, TAM** и др.

6.2 АРГУС - Технический Учет

Фундаментом, на который опираются остальные продукты **АРГУС**, является система **АРГУС-ТУ**, которая выполняет функции инвентаризации.

Система инвентаризации **АРГУС-ТУ** решает задачи учета ресурсов и услуг Оператора. Иными словами, при помощи системы **АРГУС-ТУ** Оператор может видеть полную картину своей сетевой инфраструктуры и услуг, которые на ней предоставляются. Информация о ресурсах сети используется в процессах эксплуатации: управление устранением неисправностей, управление услугами и развитием, управление активным оборудованием с функциями сбора аварийной сигнализации, измерений медных абонентских линий, включения/отключения услуг и ДВО. Инвентаризация информации о клиентах предоставляет все необходимые данные для корректной работы процессов взаимодействия с клиентом, продаж и обеспечения лояльности. Инвентаризация услуг необходима для централизованного учета всех взаимосвязей между ресурсами и клиентами.

Являясь фундаментом, на котором базируются остальные системы **АРГУС**, система **АРГУС-ТУ** поддерживает современные стандарты и требования, предъявляемые к системам класса Inventory. Из их числа особенно стоит выделить один из инструментов методологии **NGOSS**: модель общей информации и данных Shared Information and Data model (**SID**). Положенная в

основу системы **АРГУС-ТУ**, информационная модель **SID** дает системе универсальный и расширяемый язык описания информации Оператора, который существенно упрощает процессы интеграции приложений и предлагает к использованию собранные «лучшие практики» ведущих производителей отрасли **OSS**.

В системе **АРГУС-ТУ** предполагается хранение огромного объема информации. Чтобы не реализовать в ней концепцию «большого мусорного ведра», в котором хранится много ненужного, НТЦ АРГУС разработал средства проверки актуальности хранимых данных. Автоматизированные агенты сбора информации, во-первых, смогут автоматизировать процесс ввода информации в базу знаний Inventory, а во-вторых, обеспечить актуальность уже введенной и хранимой информации.

На сегодняшний день система поддерживает работу во всех технологических доменах современного Оператора связи: сети ТфОП, кабельная инфраструктура, оборудование PDH и SDH, сети передачи данных (IP/MPLS), оборудование беспроводного доступа, сети мобильной связи, волоконно-оптические сети, сети мультисервисного и оптического доступа, телеграфные сети и т.д. Это обеспечивает информацией процессы управления разными услугами от телефонной связи до IPTV и интерактивных информационных услуг.

В системе реализуется принцип однократного ввода информации: однажды введенная информация об объекте автоматически распространяется во все связанные с ним документы.

Одним из удобств является переход на полностью цифровой документооборот, что исключает возможность случайной потери информации.

Все формы документов, с которыми работает пользователь, являются полными аналогами бумажных документов.

В системе **АРГУС-ТУ** помимо учётных функций существует богатый набор дополнительных инструментов, таких как: бронирование ресурсов, подбор ресурсов для услуг, автоматическая схемография и т.п.

Стоит отметить, что, благодаря использованию передовых технологий и поддержке основных принципов взаимодействия систем, **АРГУС-ТУ** может работать в сочетании с продуктами сторонних производителей.

Как говорилось выше, **АРГУС-ТУ** является фундаментом для работы других приложений **OSS**. В составе комплекса продуктов **АРГУС** на нем разворачиваются две системы: **АРГУС-АО** и **КТП**, соответствующие областям Fulfillment и Assurance карты **eTOM**.

6.3 Абонентский Отдел

Основным приложением, входящим в состав системы **АО** и определяющим специфику ее работы, является «Обработка заказов на подключение/изменение услуг» (**Order Management**).

Приложение «Обработка заказов на подключение/изменение услуг» занимается управлением и контролем над активными клиентскими заказами, т.е. заказами, находящимися в процессе своей реализации. В **Order Management** Оператор связи может указать логику обслуживания заказа каждого типа, указать, какие действия и при каких условиях должны быть совершены. В

соответствии с этой логикой **Order Management** будет заниматься управлением и контролем над работой систем, возможности которых можно использовать в процессах реализации услуг: RT&A, Workforce management и других. В ходе выполнения заказов назначаются и будут контролироваться следующие действия: установка контролирующих выполнение работ таймеров, сбор отчетов о выполнении команд, сохранение истории своей работы и прочее. За счет выделения этого приложения выполняется ключевой принцип технологически нейтральной архитектуры **TNA (NGOSS)** – отделение бизнес-процессов от реализации конкретных компонентов, т.к. при переносе логики бизнес-процессов в компоненты (в WM или RT&A) ограничивается возможность их повторного использования. Следует обратить внимание, что при создании/редактировании алгоритма работы **Order Management** могут использоваться услуги (интерфейсы) не только приложений **АРГУС**, но и приложений сторонних производителей, например биллинговой системы, местной системы технического учета, приложений **BSS** и других. При этом существенную поддержку при использовании сторонних приложений играет соответствие интерфейсов этих приложений требованиям мировых стандартов (MTNM, MTOSI, OSS/J).

Формализация логики обработки каждого из заказов выполняется с помощью графов состояний. Для заказа каждого типа в системе формируется граф состояний заказа. Сначала в графе указываются все возможные состояния, в которых может находиться заказ, включая исходное, в которое попадает заказ после его оформления в системе. Затем с каждым из состояний заказа сопоставляются задачи, которые должны быть реализованы посредством взаимодействия с доступными приложениями (WM, RT&A и другими). В завершение в графе состояний заказов описываются условия перехода из одного состояния в другое. Упрощенный вид графа состояний для подключения клиента к продукту (услуге связи) представлен на рисунке 7.2.

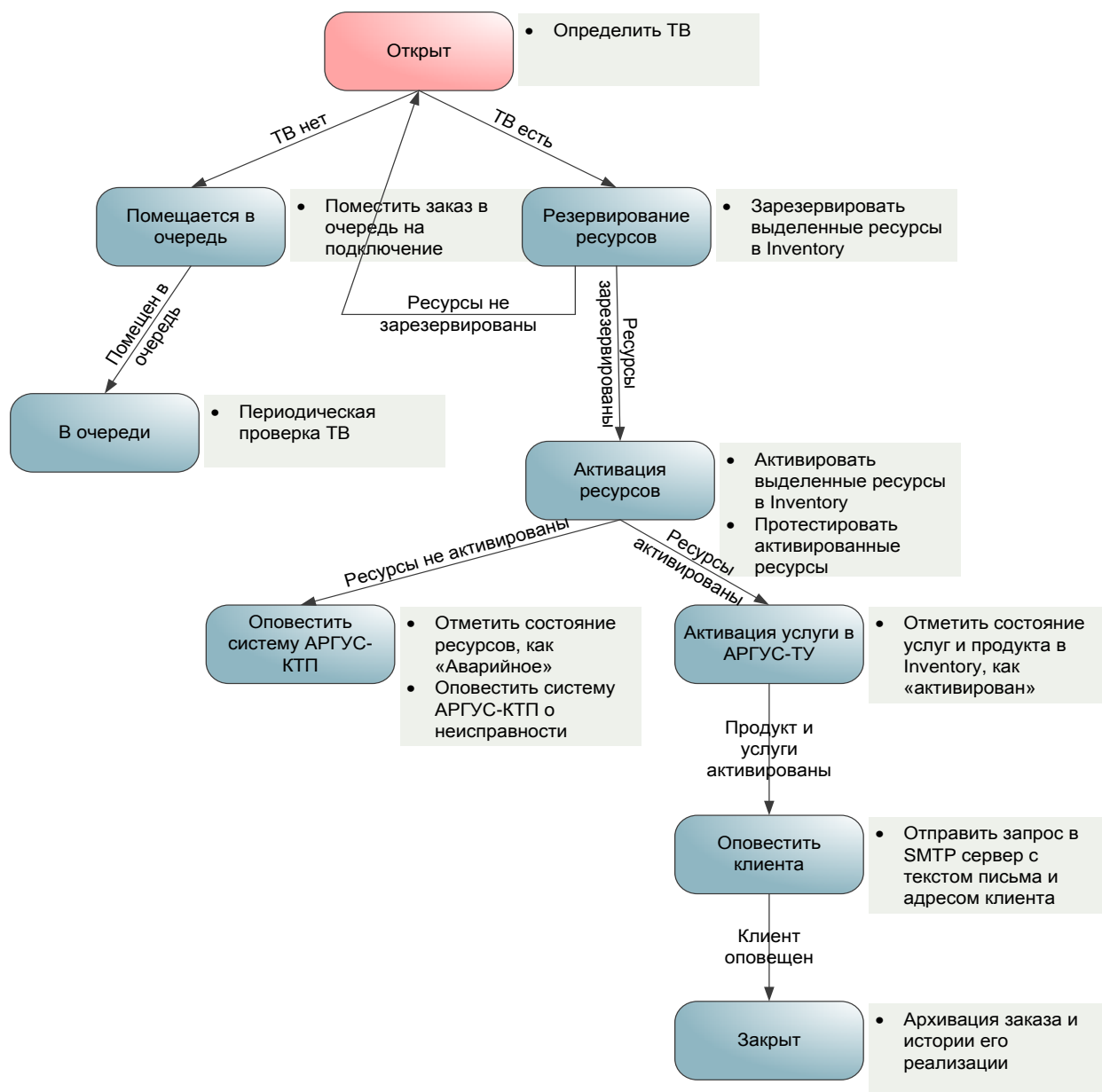


Рис. 7.2. Граф состояний заказа типа «Подключение продукта»

Начальное состояние, в котором оказывается заказ после его выпуска, отмечено красным цветом. Состоянию «Открыт» назначено выполнение задачи «Определить ТВ (техническую возможность)», что подразумевает отправку запроса в систему **АРГУС-ТУ** (или стороннюю систему класса Inventory) на определение возможности подключения соответствующего продукта клиенту с учетом исходных данных (например, «адрес проживания клиента»). При получении запроса на определение ТВ, система **АРГУС-ТУ** выполняет заранее записанные алгоритмы определения ТВ (например, анализирует характеристики медной витой пары, ведущей к указанному в исходных данных адресу, на предмет возможности ее использования для переноса сигналов ADSL-модемов). О результатах анализа **АРГУС-ТУ** извещает приложение **Order Management** (т.е. сообщает о наличии ТВ и при положительном результате – идентификаторы ресурсов, на базе которых можно предоставить услуги и продукт). При положительном результате анализа (техническая возможность есть) **Order Management** переводит заказ по ветке, которой

выполняется условие «Техническая возможность есть» в следующее состояние «Резервирование ресурсов», за которым закреплена задача «Зарезервировать ресурсы, назначенные в ходе определения ТВ». Заказ будет обрабатываться в приложении **Order Management** до тех пор, пока не попадет в состояние «Закрыт» или «Отменен» (не показано на рисунке в целях его упрощения), после чего он и история его реализации помещается в архив.

6.4 Комплексная Техническая Поддержка

Следующее приложение, которое мы рассматриваем в данном обзоре, это «**Customer Service/Account Problem Resolution**». Оно несет ответственность за обработку логики работы системы **КТП**. Область ответственности этого приложения ограничивается обработкой возникающих проблем с ресурсами, услугами и продуктами. «Проблема», которая открывается в системе, символизирует зарегистрированную неисправность в работе телекоммуникационной сети, которую необходимо устранить. В работе этого приложения может прослеживаться аналогия с работой системы **АО. Order Management** определяет логику действий для автоматизации обработки заказов, а **Customer Service/Account Problem Resolution** определяет логику действий для обработки проблем.

Однако между приложениями есть одно существенное различие. При обработке заказана подключение/изменение продуктов мы можем учесть большинство различных ситуаций и состояний, в которых может находиться заказ. Количество таких состояний и возможных ситуаций невелико даже для самых сложных с точки зрения логики обработки заказов: 20-40 состояний, по 1-4 задачи в каждом. В случае обработки проблем мы можем выделить две больших группы проблем: стандартные и уникальные. К стандартным проблемам мы отнесем те проблемы, алгоритм устранения которых заранее известен и может быть записан с помощью графа состояний проблемы, аналогично тому, что мы рассмотрели для **Order Management** (например, при сбое электропитания в стойке DSLAM необходимо восстановить конфигурационный файл). Такие проблемы могут устраняться по заранее сохраненной логике. К уникальным проблемам мы отнесем те, логику обработки которых заранее написать невозможно. Обычно такие проблемы связаны с анализом нетиповой и редко возникающей неисправности, длительным и, как правило, творческим процессом ее поиска и разрешения при непосредственном участии инженеров и техников.

Для устранения стандартных проблем приложение **Customer Service/Account Problem Resolution** предоставляет средство описания логики их устранения – граф состояния проблем. Это средство аналогично графу состояния заказов, рассмотренного для **Order Management**. Оно предоставляет гибкие возможности анализа состояния и генерации управляющих команд для устранения проблем, возникающих на телекоммуникационной сети. Для обработки проблем второго типа **КТП** предоставляет персоналу возможность взаимодействовать с сетевым оборудованием через средства автоматизации, специализированной системы взаимодействия с оборудованием **СИРИУС**, что расширяет набор доступных инструментов для анализа и устранения неисправностей непосредственно с рабочего места инженера.

Система **КТП** может также быть интегрирована с одной из систем мониторинга состояния сетевых элементов, сигнализации, услуг или с системой превентивного анализа качества обслуживания абонента, которая способна заранее оповестить систему **КТП** о «прогнозируемых проблемах»,

т.е. таких, что с некоторой вероятностью произойдут через определенный промежуток времени. Подобная совместная работа приложений позволяет организовать сквозную автоматизацию процессов устранения проблемы от момента ее обнаружения (например, в системе мониторинга **СПАЙДЕР**) до момента ее устранения в системе **КТП**.

При разработке продуктов **АРГУС** уделяется большое внимание соответствию рекомендациям TeleManagementForum. Системы, рассмотренные в этом методическом пособии, достаточно полно охватывают подходы и инструменты концепции **NGOSS**.

7 Литература

1. Самуйлов К.Е., Чукарин А.В., Яркина Н.В. Бизнес-процессы и информационные технологии в управлении телекоммуникационными компаниями. - М.: Альпина Паблшерз. - 2009. - 442 с.
2. Business Process Framework (eTOM) For The Digital Services Industry Addendum D: Member Approved Process Model Process Decompositions and Descriptions Framework Release 13.0 GB921D // TeleManagement Forum August, 2013
3. Гольдштейн А.Б., Атцик А.А., Сизюхин К.С. SID: абстракция на службе практики // Connect – 2012 – №10 – с.54-58