

## ОТКРЫТЫЕ СИСТЕМЫ: КОНЦЕПЦИЯ ИЛИ РЕАЛЬНОСТЬ

---

Эволюция открытых систем

Определение открытой системы

Интерпретация "открытости"

Стандартные платформы

Сетевая обработка

Наличие и доступность

Другие определения открытых систем

Потенциал открытых систем

Стандарты Открытых Систем

---

Сейчас, по мнению ведущих экспертов, грядет новая эпоха в области информационных технологий - эпоха открытых систем.

Импульс к применению возможностей открытых систем возник в информационных технологиях не вчера. Он следовал, в частности, из растущего использования разнообразных аппаратных платформ, исполняющих приложения, предназначенного для решения все более широкого диапазона задач.

Внедрение гетерогенных систем и желание разделять между такими системами информацию привели к необходимости их совместной работы<sup>2</sup>. Кроме того, совместимые двоично компьютеры сулили переносимость<sup>3</sup> программ с платформы на платформу.

Импульс к применению возможностей открытых систем исходит не только от пользователей. Он исходит от разработчиков, озабоченных необходимостью сокращать расходы и время переноса своих приложений на различные платформы. Он исходит от производителей, старающихся разрабатывать новые платформы, на которых может выполняться широкий диапазон приложений. Он исходит от национальных и международных стандартизирующих организаций, обнаруживших, что им необходимо своевременно разрабатывать стандарты, отвечающие требованиям промышленности.

### **Эволюция открытых систем**

Ранняя история концепции открытых систем - это история пользователей, стремящихся к переносимости приложений. Создание сети компьютеров IBM 360, обладающих единым набором команд и способных исполнять одну и ту же операционную систему, было большим шагом вперед в достижении переносимости, поскольку код, разработанный для одной системы серии, мог выполняться и на других. Кроме того, IBM предоставляла лицензии на свою ОС пользователям, которые предпочли купить компьютеры той же архитектуры у других производителей. Однако системы IBM 360 не были массовыми и на большей части рынка этот рывок к переносимости никак не сказался.

Ранние стандарты языков, например, стандарты Фортрана и КОБОЛа, предлагали отдельные решения проблемы переносимости и для приложений и для программистов. Стандарты позволяли создавать переносимые приложения, хотя зачастую и ограничивали функциональные возможности. Переносимость достигалась также тем, что к стандартам присоединились многочисленные производители многочисленных платформ. Когда языки созревали, их разработкой и сопровождением начинали заниматься национальные и международные организации по стандартизации, делая их независимыми от конкретного производителя. Достижение этого уровня переносимости было первым примером истинных возможностей открытых систем.

Однако переносимость без достаточной функциональности, вообще говоря, не слишком полезна. Достаточная функциональность без переносимости полезна, но ограничивает развитие технологии и возможность выбора. Оптимальный вариант для каждого пользователя - достаточная переносимость и достаточная функциональность.

Для некоторых пользователей совместимость на уровне двоичного кода в рамках серии IBM360 - достаточное решение обеих проблем. Для других достаточен стандартизированный язык. Для третьих требуется еще решение вопросов, оставшихся вне рассмотрения стандартов языков.

Следующая фаза в развитии концепции открытости связана с областью интерактивной компьютерной обработки и расширяющимся диапазоном требующих переноса продуктов. В этот период Digital разработал свои VAX-системы, работающие под управлением операционной системы VMS. Каждый компьютер этой линии, вне зависимости от размера, мог исполнять один и тот же набор приложений. Параллельно с этим происходило взрывное развитие сетевых технологий, таких как DECnet или Internet (TCP/IP), объединивших системы и локальные сети академических и военных организаций Соединенных Штатов. Когда сетевая обработка стала реальностью, пользователи начали обращать внимание на совместимость и интеграцию как на необходимые атрибуты открытых систем.

Тем временем в мире персональных компьютеров операционная система MS-DOS корпорации Microsoft утверждала достоинства стандартной среды для пользователей персоналок. Низкая цена и широкая распространенность создали огромный рынок для данной ОС и ее приложений. Двоичная совместимость разрешила массу проблем, хотя также не свободна от некоторых ограничений. Многие приложения, выполняющиеся в MS-DOS, могут выполняться на любой совместимой системе. Но эта система ограничена архитектурой Intel 80x86 с 16-разрядной адресацией, графикой низкого разрешения и однопроцессорностью. Для среды MS-DOS характерен также риск быстрого распространения вирусов, поскольку система слабо (или никак не) защищена на программном и аппаратном уровнях.

Хотя ОС UNIX была разработана для создания MS-DOS, позднее с появлением достаточно мощных микропроцессоров она проявила себя как наиболее перспективное открытое операционное окружение. Исторически ОС UNIX оказалась самым жизненным вариантом для создания общей базы переносимости. Она удовлетворяет ряду требований, предъявляемых к открытым системам. При соответствующем подходе к разработке программного обеспечения приложения для основанных на UNIX`е систем могут быть весьма переносимы как в другие UNIX-системы, так, во многих случаях, и в другие системы, удовлетворяющие стандартам на интерфейсы, подобным тем, которые разработаны X/Open и POSIX.

Одна из причин рассматривать систему UNIX в качестве хорошего кандидата на использование в открытых системах состоит в том, что эта ОС почти целиком написана на языке высокого уровня, модульна и относительно гибка. ОС UNIX составлена из основных компонентов, включающих ядро, инструментальные утилиты и оболочку. Ядро, образующее стержень UNIX`а, состоит из относительно маленького набора предоставляющих системные ресурсы программ, непосредственно взаимодействующих с аппаратурой.

Утилиты выполняют основные действия по обработке данных, обращаясь в определенной последовательности к процедурам ядра. Отдельные утилиты, решающие простые задачи, могут объединяться с другими утилитами для выполнения более сложных действий. Оболочка предоставляет пользовательский интерфейс и действует в точности так же, как любая другая программа. Поскольку она не интегрирована в ядро, ее можно разработать заново (что и было сделано) при изменении требований.

Хотя ОС UNIX машинно-независима, некоторые сервисы и часть кода зависят от аппаратуры. Приложения, использующие особенности конкретной версии UNIX`а, подобно приложениям MS-DOS реализационно-зависимы.

Еще один привлекательный аспект ОС UNIX состоял в готовности AT&T предоставлять лицензии на нее. Но один из результатов такой гибкости - множество различных и несовместимых реализаций. К тому же, не все поставщики выбрали

лицензионные продукты, останавливаясь вместо этого на разработке подобных UNIX`у систем с различной степенью совместимости. Деятельность ряда групп, таких как UniForum, POSIX и X/Open, направлена на поиск общего функционального ядра, которое позволит достичь переносимости между различными системами.

Кроме весьма авторитетных операционных систем, упомянутых выше, в стадии исследований и разработки находится целый ряд систем, специально проектируемых исходя из требований, предъявляемых разнородной распределенной сетевой средой. Некоторые из них - подходящие кандидаты для использования в среде открытых систем.

Сегодня внимание в открытых системах обращено не только на к операционным системам, но и к стандартным интерфейсам объединения существующих систем, приложений и пользователей. Данный подход состоит в концентрации на выработке международно признанных промышленных стандартов и добавления множества компонентов в единое модульное операционное окружение. Предполагается, что реализация стандартов в каждой системе создаст унифицированную структуру, которая уменьшит трудности в соединении разнородных схем.

Международно признанные стандарты должны быть реализованы для каждого системного компонента сети, включая каждую операционную систему и приложение. До тех пор, пока компоненты остаются верны таким стандартам, они соответствуют целям открытых систем. Однако для некоторых особо специфичных компонентов стандартов может не существовать - в таких случаях реализация стандартов слишком сложна либо слишком накладна.

Чтобы исполнить обещанное, открытые системы должны не только обеспечить операционное окружение, в котором приложения можно легко перемещать между различными аппаратными и программными платформами, а пользователи могут легко перемещаться между системами; они должны функционировать сообща с уже используемыми, стоящими миллиарды долларов продуктами.

#### **Определение открытой системы**

До сих пор мы использовали термин "открытые системы", не давая ему точного определения. Одна из основных трудностей для тех, кто связан с миром открытых систем, состоит в том, что данный термин обозначает разное для разных людей и организаций. Фактически же, с практической точки зрения, открытая система должна быть тем, что ждут от нее в каждой организации.

Однако для целей дальнейшего обсуждения нам будет удобно прийти к соглашению об определении открытых систем. Комитет IEEE POSIX 1003.0 остановился на определении, которое дает широкую и исчерпывающую трактовку понятия открытых систем.

**В соответствии с этим определением открытая система есть "система, реализующая открытые спецификации на интерфейсы, сервисы и поддерживаемые форматы данных, достаточные для того, чтобы обеспечить должным образом разработанным приложениям возможность переноса с минимальными изменениями на широкий диапазон систем, совместной работы с другими приложениями на локальной и удаленных системах и взаимодействия с пользователями в стиле, облегчающем тем переход от системы к системе".**

Ключевой момент в этом определении - использование термина "открытая спецификация", что в свою очередь определяется как "общедоступная спецификация, которая поддерживается открытым, гласным согласительным процессом, направленным на постоянную адаптацию новой технологии, и соответствует стандартам."

По данному определению открытая спецификация не зависит от технологии, т.е. не зависит от специфического аппаратного и программного обеспечения или от

продуктов конкретного производителя. Она одинаково доступна любой заинтересованной стороне. Более того, открытая спецификация находится под контролем общественности и поэтому все, кого она затрагивает, могут участвовать в ее разработке.

Многие консорциумы и частные компании разрабатывают спецификации, не попадающие под это определение. Однако ключевой характеристикой в определении открытой спецификации является не источник, а поддержка. Например, то, что Фортран был разработан IBM, не помешало появлению открытой спецификации, поскольку стандарт Фортрана сейчас поддерживается открытым и гласным согласительным процессом.

Данное IEEE определение открытой системы, выделяющее основанные на открытых спецификациях переносимость и возможность совместной работы, будет основой всего дальнейшего обсуждения.

Определение POSIX 1003.0 подходит для этой книги по ряду причин. Возможно, важная - источник определения. Определение, данное IEEE POSIX 1003.0, разработано в результате широкого обсуждения, в котором участвовали представители самых разных секторов информационных технологий - как производители, так и пользователи.

Используемый IEEE согласительный процесс гарантирует, что принятое определение содержит в себе весь диапазон перспектив. Определение не ограничено и не подвержено чрезмерному влиянию ни одной из заинтересованных сторон. Более того, это формальный процесс стандартизации, обеспечивающий критерий стабильности выработанных решений. Он также гарантирует всем равный доступ к информации и равную возможность участия в стандартизации.

Как результат широкого участия в разработке определения открытой системы, оно могло соединить различные точки зрения на идею открытости. Одновременно это не запрещает организациям при развертывании открытых систем учитывать и свое собственное мнение.

Определение, данное POSIX, позволяет нам с течением времени развивать понимание открытых систем. Не останавливаясь на конкретных продуктах или областях технологии, оно предоставляет пользователям остов, который может расширяться с изменением потребностей.

### **Интерпретация "открытости"**

Существует много различных интерпретаций слова "открытый". В этой книге открытая система определяется как прикладное окружение, основанное на интерфейсных стандартах, обеспечивающих переносимость приложений, "переносимость" пользователей и возможность совместной работы. Такой подход может отличаться от других, но лишен ограниченности ряда общепринятых определений, каждое из которых обращается к какому-либо частному аспекту цельной картины открытых систем.

Цитируемое определение IEEE было разработано пользователями и производителями с учетом широкого диапазона вопросов, важных для многих разных организаций. В этом разделе рассматривается ряд распространенных интерпретаций "открытости" и обсуждается, насколько они соответствуют определению IEEE.

### **Стандартные платформы**

Одна из распространенных точек зрения состоит в признании "открытой" системы, основанной на стандартной аппаратуре и/или стандартной операционной системе. Пример такой системы - IBM PC или любая совместимая с ней система. Большинство таких платформ используют одну и ту же операционную систему и выполняют практически одни и те же приложения.

Многие полагают, что именно операционная система определяет, является ли система открытой, и что предоставляемые ОС сервисы играют существенную роль в

том, как компьютерная система функционирует и как она вписывается в объемлющее окружение. Такая трактовка открытости основана на теории, что архитектуры, работающие под управлением одной и той же ОС, будут автоматически исполнять одни и те же приложения, решая проблему переносимости; что они будут взаимодействовать друг с другом, решая проблему возможности совместной работы; что пользовательский интерфейс (монитор, клавиатура, мышь) будет оставаться единым, решая проблему "переносимости" пользователей.

В теории это так. Но совместные изолированные приложения нуждаются во многих средствах, не предоставляемых непосредственно операционной системой. Графика, сетевая обработка, почтовая служба, средства распределенной обработки, управление и передача данных - все это механизмы, которые необходимы для приложений, но не предоставляются, вообще говоря, самой операционной системой. Кроме того, операционные системы будут меняться с изменением типа и числа терминалов. Все эти изменения могут понадобиться, чтобы приспособиться к новым или более эффективным приложениям. Любое из них может воздействовать на переносимость и совместную обработку.

Достижение целей открытых систем требуется стандартизация в значительно более широком диапазоне областей и на уровне более высоком, чем уровень реализации.

### **Сетевая обработка**

Некоторые представители компьютерного сообщества полагают, что под открытыми системами подразумевается соединение вместе различных типов систем от разных производителей. В этом случае ударение делается на названии, данном набору стандартов ISO на коммуникации продуктов от разных производителей, - Взаимодействие Открытых Систем (Open Systems Interconnection - OSI). Как и операционные системы, сети существенны для открытых систем, но не достаточны сами по себе и без связи с другими аспектами.

Сетевые стандарты направлены на возможность совместной работы и затрагивают проблемы переносимости, но не обеспечивают полного решения этих проблем.

### **Наличие и доступность**

К области открытых систем относится и ряд вопросов, связанных с доступностью технологии или спецификаций. Существует несколько подходов к ним. Технология, предоставляемая единственным источником, но для многочисленных платформ (например, OSF Motif), - другой пример технологии, считающейся открытой. Любой производитель может приобрести лицензию на Motif и продавать этот продукт покупателям. Еще один пример - операционная система UNIX, предоставляющая первоначально AT&T, а впоследствии UNIX System Laboratories (USL), дочерней компанией AT&T. USL владеет UNIX`ом и контролирует выдачу лицензий. Реализации, базирующиеся на AT&T UNIX, доступны из многих разных источников, включая производителей аппаратуры, предлагающих свои UNIX- компьютеры, и производителей программного обеспечения, предоставляющих реализации для разных платформ. В обоих случаях поставщик может адаптировать базовую технологию к особенностям своей реализации.

Несколько отличный пример - система управления базами данных ORACLE. Ее можно получить только у корпорации Oracle или у производителей аппаратного обеспечения, имеющих торговое соглашение с Oracle; тем не менее, система доступна на весьма широком спектре платформ от различных производителей. В этом случае, приобретающие лицензию не имеют права менять технологию ORACLE.

Открытые системы часто ставят в один ряд с "собственными" системами. В лучшем случае это создает путаницу. В историческом и правовом смысле термин "собственный" означает, что технология принадлежит какой-либо одной компании. Это

верно практически для всех продаваемых сегодня продуктов. Например, AT&T контролирует операционную систему UNIX System V, Microsoft контролирует операционную систему MS-DOS и так далее. То, что эти компании предпочитают предоставлять за разумную цену лицензии на свои технологии всем заинтересованным сторонам, достойно аплодисментов, но они вовсе не обязаны так поступать. Более того, архитектурные зависимости, как и дальнейшая эволюция интерфейсов, контролируется исключительно владельцем.

Группы, обладающие различными степенями влияния, могут публиковать требования, спецификации и даже стандарты, относящиеся к специфическим продуктам и реализациям. Даже если эти спецификации реализуются, продукты остаются собственными. Слово "собственный" в этой книге не используется, поскольку оно не точно отражает сущность реализаций открытых систем.

### **Другие определения открытых систем**

Определение, данное IEEE и базирующее открытые системы на стандартных интерфейсах, не противоречит ни одной из приведенных выше трактовок; скорее, оно все эти трактовки примиряет, поскольку направлено на более широкие цели. Основанная на стандартах среда открытых систем может включать в себя все типы продуктов, упомянутых выше. Ценность конкретного продукта и необходимость соблюдения стандартов должны быть взвешены. Интеграцию уже существующей системы, быть может, не вполне удовлетворяющей стандартам, часто можно осуществить, не вступая в противоречие с формированием основанной на стандартах открытой среды.

При планировании открытых систем важно концентрировать внимание на стандартах интерфейсов, а не на специфических продуктах или реализациях; мнение, что ключом к открытым системам является какой-то специфический продукт, весьма рискованно. Если в процессе планирования не рассматривать стандарты интерфейсов, приложение может быть ограничено конкретной платформой и не будет переносимо в другие системы; к тому же, новая технология, способная предоставлять организации преимущества перед конкурентами, может не позволить воспользоваться такими приложениями.

Подход к открытым системам, основанный на стандартных интерфейсах, дает пользователю дополнительный запас прочности, поскольку продукты, реализующие спецификации и стандарты, согласованы между собой и могут быть протестированы для того, чтобы в этом удостовериться. Согласованность для пользователя - реальный источник независимости от производителя.

В конечном счете, открытая система, как бы она не определялась, должна отвечать устремлениям клиентов. Главный вопрос, который пользователи всегда должны задавать сами себе: "Обладает ли данный продукт достаточной функциональностью для потребностей моих открытых систем?" Лучший способ, гарантирующий правильный ответ на этот вопрос, заключается в следовании тщательно разработанному плану. Формулировка такого плана требует определения целей и приоритетов и тщательного анализа, насколько каждый элемент окружения им соответствует.

### **Потенциал открытых систем**

В идеальном мире открытых систем все приложения будут выполняться во всех системах, работникам потребуется минимальная подготовка для работы в одном окружении, они смогут без дополнительной подготовки переходить в другое окружение, все системы можно будет объединять, чтобы свободно разделять и прозрачным образом обмениваться информацией.

Как показывает кривая 1 на рисунке, новые, технологически продвинутые платформы обладают ограниченным диапазоном приложений, в то время как сложные приложения годами разрабатываются и уточняются для платформ, специально для этих

приложений подготовленных. Кривая зависимости между степенью технологической продвинутой платформы и сложностью приложений - это гипербола, форму которой задают вложения в создание реализационно зависимого кода, обучение персонала, установку баз данных и поддержку.

Совместимость новых и уже существующих реализаций сдвигает кривую влево. Но не ликвидирует зависимость в реализации, поэтому форма кривой не изменяется.

Международно признанные стандарты открытых систем, однако, действительно изменяют форму кривой, позволяя разрабатывать приложения, которые в состоянии охватить более широкий спектр технологий (кривая 2). Эти приложения будут выполняться и на платформах, которые появятся в будущем.

В то же время, существуют два ограничения, не позволяющие стандартам, мгновенно адаптируясь к усложнениям технологии и приложений, превратить данную кривую в теоретическую горизонтальную прямую (кривая 3). Во-первых, стандарты не полны. Этот недостаток не устранить никогда из-за нарастающей сложности и объема работы. Во-вторых, есть пределы предвидению технологического роста, которое может быть заложено в стандарты. Производители используют новшества, чтобы разработать еще более передовые продукты. Пользовательский спрос на новинки вряд ли уменьшается.

Тем не менее, стандарты дают преимущества пользователям и поставщикам, указывая более эффективный путь из нашего нынешнего положения туда, куда мы хотим двигаться.

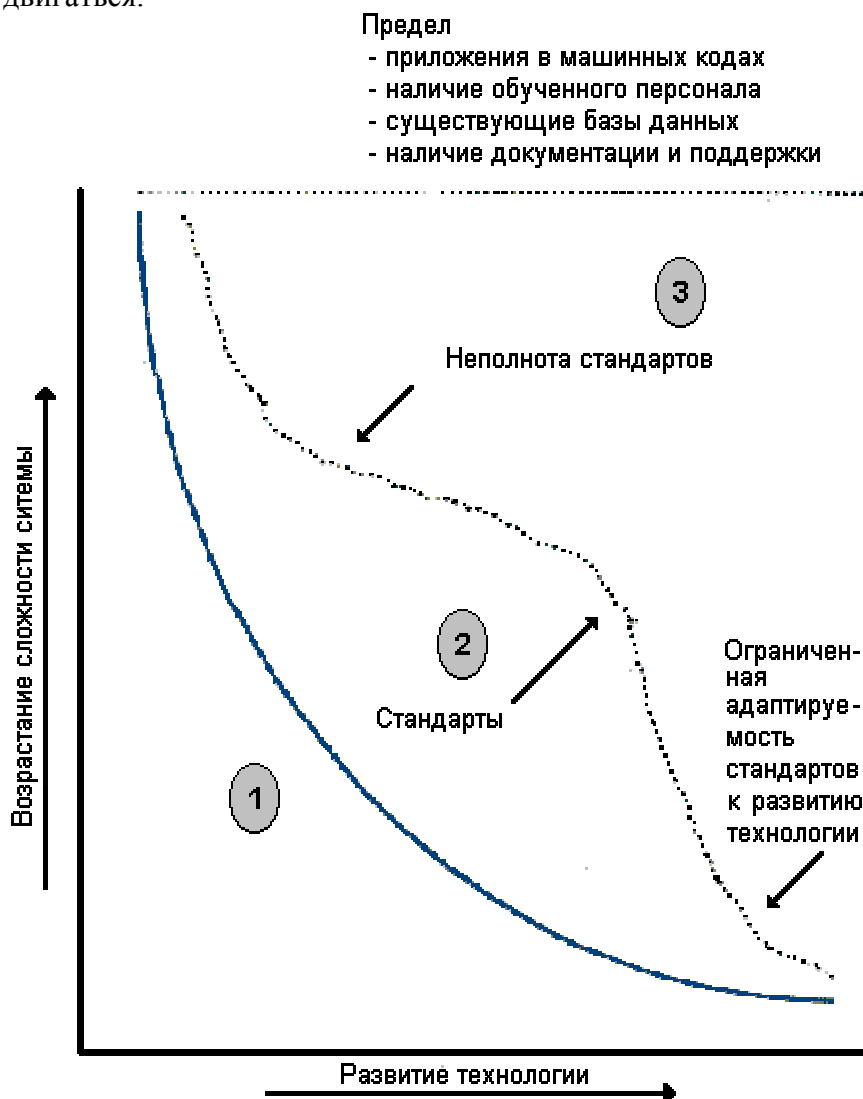


Рис. 1.1. Ограничения открытых систем.

## Стандарты Открытых Систем

События последнего времени и тенденции в индустрии информационных технологий ясно показали, что пользователи возлагают вполне определенные надежды на открытые системы. Как говорилось ранее, основные ожидания пользователей связаны со следующими областями: переносимость приложений, мобильность людей, доступность приложений, возможность совместной работы систем. Пользователям необходимо быть уверенными в том, что они получают то, что хотят.

В дополнение к традиционному воздействию, оказываемому пользователями как покупателями в рамках их взаимоотношений с производителями, пользователи открытых систем оказывают воздействие и на процессы установления и реализации стандартов.

Сегодня стандарты определяют направления развития открытых систем и всей индустрии информационных технологий. Более 250 подкомитетов официально стандартизирующих организаций работают над стандартами информационных технологий. Более 1000 стандартов либо уже принято этими организациями, либо находится в процессе разработки. Но со стандартами и их реализацией связана большая путаница; видны и маневры поставщиков, призывающих стандартизовать их продукты, либо оказывающих воздействие на определение направлений стандартизации.

Стандарт *де факто* - термин, обозначающий продукт какого-либо поставщика, который захватил большую долю рынка и который другие поставщики стремятся эмулировать, копировать или использовать для того, чтобы захватить свою часть рынка.

Одна из главных причин значимости современной программы стандартизации - осознание опасности злоупотребления стандартами *де факто*. В 60-е и 70-е годы создание стандартов *де факто* ставило пользователей в зависимое от производителей положение при использовании основных средств обработки данных и телекоммуникаций. Важный аспект сегодняшней работы по стандартизации - преодоление этой зависимости через продвижение стандартных интерфейсов.

Стандарт *де юре* создается формально признанной стандартизирующей организацией. Он разрабатывается при соблюдении правил консенсуса в процессе открытой дискуссии, в которой имеет шанс принять участие каждый. Ни одна группа не может действовать независимо, создавая стандарты для промышленности. Если какая-либо группа поставщиков создаст стандарт, не учитывающий требования пользователей, она потерпит неудачу. То же самое происходит, если пользователи создают стандарт, с которым не могут или не будут соглашаться поставщики, этот стандарт также не будет успешным. Когда участие различных групп в разрабатывающих стандарты организациях сбалансировано, согласительный подход к стандартам *де юре* позволяет всем участникам воплотить свои реалистичные ожидания и продвинуться в своих достижениях. Стандарты *де юре* не могут быть изменены, не пройдя через процесс согласования под контролем организации, разрабатывающей стандарты. Стандарты OSI, Ethernet, POSIX, SQL и большинство стандартов языков - примеры такого рода стандартов.

В своей книге "Стандартизация в информационных технологиях" Карл Ф.Каргил (Carl F.Cargill) представляет детальную картину процесса стандартизации и модель того, как этот процесс может развиваться в соответствии с современными требованиями индустрии информационных технологий.



# МЕТОДОЛОГИЧЕСКИЙ БАЗИС ОТКРЫТЫХ СИСТЕМ

---

## Базовые понятия концепции открытых систем

### Структура методологического базиса

#### Многоуровневая модель

#### Архитектурные спецификации

#### Уровень базовых спецификаций

---

*Структура современного методологического базиса открытых систем имеет определяющее значение для развития всей информационной индустрии. Предложенная в данной статье семиуровневая декомпозиция спецификаций информационных технологий может быть полезной для анализа данной области, а также при разработке профилей открытых систем.*

Методологический базис информационных технологий (ИТ), представляет собой основу для создания наиболее экономически рентабельных технологий и систем, удовлетворяющих свойствам открытости или, так называемых открытых систем. В первую очередь к свойствам открытости относятся переносимость и переиспользуемость программного обеспечения, интероперабельность и масштабируемость систем. Наиболее значительными результатами в становлении методологического базиса открытых систем сегодня являются:

- создание целостной системы авторитетных специализированных международных организаций по разработке и стандартизации ИТ открытых систем;
- разработка эталонных моделей и соответствующих им базовых спецификаций для важнейших разделов области ИТ, что позволило сформировать концептуальный и функциональный базис пространства ИТ;
- разработка и широкое использование концепции профиля, предоставляющей аппарат для спецификации и документирования комплексных открытых ИТ, параметризующих и комбинирующих функциональности базовых спецификаций и/или профилей;
- разработка таксономии профилей, представляющей собой классификационную систему ИТ и обеспечивающую систематическую идентификацию профилей в пространстве ИТ;
- разработка концепции и методологии конформности (соответствия) реализаций ИТ-систем тем спецификациям, которые ими реализуются.

По существу сформировались не только концептуальные и методологические основы открытых систем, но и достаточно развитый аппарат конструирования и сертифицирования новых открытых технологий в пространстве базовых стандартизованных решений. Таким образом сегодня можно говорить о формировании новой научной дисциплины, безусловно базовой для информатики, но не получившей еще общепризнанного названия. Предлагаемыми названиями являются: "IT Fundamentals", "The Foundations of IT", "Analysis of IT", "IT-Science", "Itology", "Анализ ИТ" или "Итология".

### **Базовые понятия концепции открытых систем**

Прежде, чем перейти к анализу указанных выше методологических основ, необходимо ввести набор базовых определений концепции открытых систем. При этом следует иметь ввиду, что ИТ имеют две формы представления:

- в виде ИТ-спецификаций или описаний функциональных возможностей и поведения ИТ, задокументированных в строго регламентированной форме;
- в виде ИТ-систем, представляющих собой реализации конкретных ИТ-спецификаций.

Стандарт (по определению ISO). Технический стандарт или другой документ, доступный и опубликованный, коллективно разработанный или согласованный и общепринятый в интересах тех, кто им пользуется, основанный на интеграции результатов науки, технологии, опыта, способствующий повышению общественного блага и принятый организациями, признанными на национальном, региональном и международном уровнях.

Базовый стандарт или базовые спецификации (формальный стандарт или стандарт de-ure). Международный стандарт, принятый ISO или Рекомендация организации ITU-T (до 1993 г. - ССИТТ) - международного союза по телекоммуникации.

Эталонная модель (Reference Model). Структурированное множество понятий и их взаимосвязей для некоторой предметной области, осуществляющее концептуальную структуризацию данной области и имеющее достаточно обобщенное описание. По существу эталонная модель является формой метазнаний, определяющих принципиальную декомпозицию или архитектурную спецификацию конкретной предметной области.

ИТ-система (IT system). Совокупность ресурсов информационных технологий, предоставляющая сервисы на одном или более интерфейсов.

OSE (Open Systems Environment - Окружение открытых систем). Полный набор интерфейсов, услуг, форматов, а также пользовательских аспектов, обеспечивающих интероперабельность и/или переносимость приложений, и данных в рамках соответствующих спецификаций базовых стандартов и профилей ИТ. Важным и почти обязательным свойством открытости является свойство масштабируемости ИТ. В эталонной модели прикладного пользовательского интерфейса (API) [4] под открытой системой фактически понимается система, реализующая OSE - окружение, удовлетворяющее стандартам.

Переносимость (portability). Свойство ИТ-системы, позволяющее с возможно меньшими накладными расходами осуществлять перенос ее программного обеспечения, информации и пользователей с одной прикладной платформы на другую.

Интероперабельность (interoperability). Возможность совместного использования информации и ресурсов компонентами распределенной системы.

Масштабируемость (scalability). Свойство системы, позволяющее ей эффективно работать в широком диапазоне параметров, определяющих ее собственные технические и ресурсные характеристики и/или ее поддерживающей среды: число процессоров, число узлов сети, максимальное число обслуживаемых пользователей, количество обрабатываемых транзакций.

PAS (Publicly Available Specifications - Общедоступные спецификации). По существу это понятие охватывает опубликованные стандарты de-facto, например, промышленные стандарты. Близким по смыслу понятию PAS является понятие открытых спецификаций, определенное в эталонной модели API [4]. Примерами PAS могут служить спецификации IEEE POSIX и X/Open XPG4, разработанные с целью обеспечения переносимости приложений, а также спецификации IETF для TCP/IP.

Профиль (Profile). Набор, состоящий из одного или более базовых стандартов и/или ISP, содержащий указание области применимости, список выбранных классов сервиса, тестовых наборов, опций и параметров базовых стандартов и ISP, необходимых для выполнения конкретной прикладной функции.

ISP (International Standardized Profile - Международный стандартизованный профиль). Согласованный на международном уровне официальный документ, описывающий один или несколько профилей. В эталонной модели API используется близкое понятие стандартизованного профиля.

OSE-профиль. Профиль, который специфицирует все поведение ИТ-системы или часть ее поведения на одном или большем числе интерфейсов OSE.

OSI-профиль. Конкретный профиль, составленный из базовых спецификаций, соответствующих модели OSI [5], возможно дополненных базовыми стандартами и/или профилями для представления данных обмена и их форматов - так называемыми F-профилями.

API-профиль. Профиль, определяющий конкретную комбинацию базовых спецификаций прикладного пользовательского интерфейса в соответствии с моделью POSIX [6], возможно дополненных базовыми стандартами и/или профилями для представления данных обмена и их форматов.

Таксономия (Taxonomy) - классификационная схема, применяемая для однозначной идентификации профилей или наборов профилей.

### Структура методологического базиса МНОГОУРОВНЕВАЯ МОДЕЛЬ

Методологический базис ИТ, основную часть которого составляют спецификации ИТ различных уровней абстракции, формируется на основе иерархического подхода, что способствует анализу его структуры с помощью некоторой многоуровневой модели. На основе определенного опыта работы в данной области на рис.1 предложена семиуровневая модель, представляющая собой достаточно полную классификационную схему спецификаций ИТ. В данной модели выделены следующие уровни спецификаций ИТ:



Структура знания итологии.

- 1) Концептуальный уровень или уровень метазнаний - состоит из архитектурных спецификаций, называемых эталонными моделями (Reference Models). Архитектурные спецификации предназначены для структуризации спецификаций функций, определяющих семантику конкретных областей ИТ.
- 2) Функциональный уровень или уровень базовых спецификаций (базовых стандартов), включающий также PAS и предназначенный для определения индивидуальных функций или наборов функций, описанных в эталонных моделях.
- 3) Предметные или локальные профили ИТ (например, OSI-профили, API-профили), т.е. профили, разрабатываемые на основе использования базовых спецификаций, относящихся к предметной области, описанной одной эталонной моделью (возможно вместе с профилями форматов данных, т.е. F-профилями).
- 4) OSE-профили - спецификации поведения открытых систем на их границах (интерфейсах), комплексизирующие базовые спецификации и/или профили, базирующиеся на различных эталонных моделях.

5) Полные OSE-профили открытых платформ и систем - спецификации, предназначенные для описания поведения ИТ-систем на всех их интерфейсах.

6) OSE-профили прикладных технологий. Полная спецификация окружений прикладных технологий обработки данных (например, банковских систем, распределенных офисных приложений и т.п.), построенных на принципах открытости, т.е. удовлетворяющих условиям переносимости, интероперабельности, а также масштабируемости.

7) Стратегические профили (например, GOSEP - Government's Open System Environment Profile), т.е. профили, рассматриваемые в данном случае не как спецификации одной технологии, а как наборы стандартов, определяющих техническую политику в области телекоммуникации и открытых технологий крупной организации или даже государства.

Как следует из описания модели, нижние два уровня выполняют роль концептуального и функционального базиса области ИТ. Рассмотрим подробнее текущее состояние этих уровней.

### **АРХИТЕКТУРНЫЕ СПЕЦИФИКАЦИИ**

Ядром рассматриваемой методологии служат метазнания, структурирующие область ИТ. Роль таких метазнаний выполняют эталонные модели, определяющие архитектуру наиболее важных и достаточно независимых разделов ИТ. Таким образом каждая эталонная модель представляет собой концептуальный и методологический базис конкретного раздела ИТ, определяя структуру множества базовых спецификаций, соответствующих данному разделу. Наиболее известными эталонными моделями являются:

- Базовая эталонная модель взаимосвязи открытых систем (Basic Reference Model for Open Systems Interconnection - RM-OSI).
- Руководство по окружению открытых систем POSIX (Portable Operating System Interface for Computer Environments - RM API).
- Эталонная модель для открытой распределенной обработки (Reference Model for Open Distributed Processing - RM-ODP).
- Эталонная модель управления данными (Reference Model for Data Management - RM DF).
- Эталонная модель машинной графики (Reference Model of Computer Graphics - RM CG).
- Эталонная модель текстовых и офисных систем (ISO/IEC TR02501-1), а также общая (general) модель распределенных офисных приложений.

В процессе разработки находятся следующие эталонные модели:

- Конформность (соответствие, подобие) и методы тестирования конформности, называемые также методами аттестационного тестирования;
- основы общей безопасности (generic security frameworks);
- качество OSI-сервиса (Quality of Service for OSI).

### **УРОВЕНЬ БАЗОВЫХ СПЕЦИФИКАЦИЙ**

Базовые спецификации являются основными строительными блоками, из которых конструируются конкретные открытые технологии. Хотя PAS, в соответствии с данным выше определением, охватывают стандарты de-facto, которые не являются международными стандартами, сейчас интенсивно осуществляется процесс принятия наиболее распространенных и сопровождаемых PAS в качестве международных стандартов (по специально разработанной ISO быстрой процедуре баллотирования PAS), что открывает возможность использования PAS в качестве элементов стандартизованных профилей ИТ. Системный подход к проектированию профилей опирается на классификацию базовых спецификаций и PAS, в основе которой используется по существу ортогональный набор эталонных моделей. В частности, возможна следующая классификация базовых спецификаций и PAS.

- а)** Базовые функции ОС: определяются стандартами по окружению открытых систем POSIX (Portable Operating System Interface for Computer Environments).
- б)** Функции управления базами данных: язык баз данных SQL (Structured Query Language); информационно-справочная система IRD (Information Resource Dictionary System); протокол распределенных операций RDA (Remote Data base Access); PAS Microsoft на открытый прикладной интерфейс доступа к базам данных ODBC API.
- в)** Функции пользовательского интерфейса, которые включают следующие ИТ: MOTIF из OSF для графического пользовательского интерфейса (GUI) и стандарт OPEN LOOK; X Window вместе с GUI и телекоммуникациями; стандарты для виртуального терминала (Virtual Terminal - VT), включая процедуры работы VT в символьном режиме через TCP/IP; стандарты машинной графики GKS (Graphical Kernel System), GKS-3D (Graphical Kernel System - 3 Dimentional), PHIGS (Programmers Hierarchical Interactive Graphics System), а также CGI (Computer Graphics Interface).
- г)** Функции взаимосвязи открытых систем, включающие: спецификации сервиса и протоколов, разработанные в соответствии с моделью OSI (рекомендации серии X.200); стандарты для локальных сетей (IEEE 802); спецификации сети Internet.
- д)** Функции распределенной обработки, включая следующие базовые спецификации OSI: вызов удаленной процедуры RPC (Remote Procedure Call), фиксация, параллельность и восстановление CCR (Commitment, Concurrency and Recovery); протокол надежной передачи (RT); обработка распределенной транзакции DTP (Distributed Transaction Processing); управление файлами, доступ к файлам и передача файлов FTAM (File Transfer, Access and Management); управление открытыми системами (OSI Management); API для доступа к сервису Object Request Broker (ORB) в архитектуре CORBA и API, определяющий базовые возможности такого сервиса (Common Object Services - COS 1), а также язык спецификации интерфейсов объектов IDL (Interface Definition Language) и его проекции на объектно-ориентированные языки.
- е)** Распределенные приложения: спецификации специальных сервисных элементов прикладного уровня модели OSI, стандартов Internet, OMG, X/Open. Как, например: система обработки сообщений MHS (Message Handling System - X.400), служба справочника (The Directory - X.500); спецификации распределенных приложений с архитектурой клиент-сервер и распределенных объектных приложений.
- ж)** Структуры данных и документов, форматы данных: средства языка ASN.1 (Abstract Syntax Notation One), предназначенного для спецификации прикладных структур данных - абстрактного синтаксиса прикладных объектов; форматы метафайла для представления и передачи графической информации CGM (Computer Graphics Metafile); спецификация сообщений и электронных данных для электронного обмена в управлении, коммерции и транспорте EDIFACT (Electronic Data Interchange for Administration, Commence and Trade); спецификации документов: спецификации структур учрежденческих документов ODA (Open Document Architecture); спецификации структур документов для производства, например: SGML (Standard Generalized Markup Language); языки описания документов гипермедиа и мультимедиа, например: HyTime, SMDL (Standard Music Description Language), SMSL (Standard Multimedia/Hypermedia Scripting Language), SPDS (Standard Page Description Language) [50], DSSSL (Document Style Semantics and Specification Language), HTML (HyperText Markup Language); спецификация форматов графических данных, например: форматов JPEG, JBIG и MPEG.
- з)** спецификации инструментальных окружений (в частности, языков реализации и их библиотек) и CASE-окружений. Анализ базовых спецификаций ИТ показывает, что современная методологическая база открытых систем представляет собой сложную систему концептуальных, структурных, функциональных, поведенческих и лингвистических моделей, взаимосвязанных между собой, а также вспомогательных процедур и средств. При этом следует отметить динамичность развития всей этой

системы, поддерживаемого целенаправленной деятельностью развитой инфраструктуры специализированных международных институтов.

Приведенный обзор базовых спецификаций ИТ является достаточно общим и возможны другие подходы к классификации и анализу спецификаций ИТ. Однако, следует подчеркнуть, что вся область спецификаций ИТ, несмотря на свою обширность и техническую сложность, легко систематизируется, что важно при использовании спецификаций в процессе разработки новых открытых систем и технологий, например, посредством аппарата профилирования.

### **Литература**

Методологический базис открытых систем

16.04.1996

*Открытые системы, #04/1996 Владимир Сухомлин*

Открытые системы: концепция или реальность

17.04.1993

*Открытые системы, #04/1993 А.Я.Олейников*