

Глава 6. Процессы, деятельности и артефакты жизненного цикла

6.1. Системная инженерия

«Система» - самоназвание движения хиппи в СССР в конце 70-х годов.

«Система» — множество элементов, находящихся в отношениях и связях друг с другом, которое образует определённую целостность, единство». **Википедия.**

«Не поминай имени Господа Бога твоего всуе». Третья заповедь **Ветхого Завета.**

Слово «система» является настолько общим местом, что предметом системной инженерии становятся не все «системы» по определению, а только обладающие рядом свойств:

- сложность, большое количество составляющих элементов (порядка $10^6 \dots 10^9$ и выше), наличие нескольких уровней иерархии;
- междисциплинарность изучения и описания: система не может быть описана в рамках одной дисциплины и ее законов;
- целостность – воспринимается и существует как единое целое, а не как простой набор элементов, демонстрирующих взаимосвязанное поведение;
- множественность представлений – имеет различные целостные *аспекты* (представления) с точки зрения разных групп лиц, имеющих к ней отношение.

В качестве примера рассмотрим наиболее сложные известные системы - живые организмы и программные системы («железо» + ПО). Здесь важен не только количественный фактор, но и наличие примерно одинакового числа уровней иерархии. Для программной системы это:

1. Среда функционирования: твердотельные молекулярные структуры, полупроводники;
2. Базовые элементы: транзистор, вентиль;
3. Схемотехника: комбинационные схемы, триггеры, счетчики, регистры;
4. Микроархитектура: функциональные узлы, автоматы, операции, микропрограммирование;
5. Макроархитектура: управление памятью, кэш, контроллеры, конвейеры;
6. Аппаратная архитектура: система команд, ассемблер;
7. Среда исполнения: языки программирования, трансляторы, средства сборки и исполнения, операционные системы;
8. Программная единица: ООП, классы, паттерны, (конструирование, технологии программирования);
9. Программный комплекс: архитектура ПО, архитектурные стили, слои, подсистемы, взаимодействие, протоколы, функциональное и бизнес-проектирование (проектирование, программная инженерия).

Не следует забывать, что аппаратная составляющая имеет сопоставимую сложность [6-24]. Оценка иерархии программной системы:

- большинство уровней создаются на основе предыдущего по принципу количественной композиции элементов («состоит из...»), отдельные из них (уровень 6) – как функционал предыдущего.
- каждый уровень базируется на собственной области знаний и технологиях
- каждый уровень может быть оценен в $10^2 \dots 10^3$ элементов предыдущего, т.е. $10^6 \dots 10^9$ на «железо» (в количестве вентилях) и столько же - на программное обеспечение (команд, SLOC);
- на верхних уровнях ПО отсутствуют формальные методы создания системы «под задачу», на их месте - технологические приемы (паттерны) и тестирование.

Высшие живые организмы, в свою очередь, состоят из подсистем. В качестве примера рассмотрим уровни иерархии и изучающие их научные дисциплины для нервной системы человека [6-25]. Попытаемся сопоставить уровни с предыдущей системой:

1. Среда функционирования: гены, белки, ферменты (*молекулярная биология, генетика*);
2. Базовые элементы: нейромедиаторы, рецепторы, мембраны (*нейрофизиология*);
3. Схемотехника: нейрон, синапсы, аксон, дендриты, спайки, пластичность, ионотропные, метаболитропные и генетические процессы (*нейрофизиология*);
4. Микроархитектура: виды нейронов, взаимодействие нейронов, нейронные ансамбли, пластичность (*нейрофизиология*);
5. Макроархитектура: кора головного мозга (колонки, слои), структура (гистология) других отделов нервной системы (*нейрофизиология*);
6. Архитектура нервной системы: кора и ее зоны, таламус, гиппокамп, полушария, механизмы обучения и памяти (*нейрофизиология, психофизиология*);
7. Высшие функции нервной системы: поведение, высшая нервная деятельность (*психофизиология, психология*);

Оценка сложности системы:

- верхние функциональные и нижние структурные уровни изучаются различными научными дисциплинами – *психофизиология, нейрофизиология, молекулярная биология и генетика*;
- нейрофизиология дает значительный объем фактического, экспериментально проверяемого и систематизированного материала об уровнях 2-3, как о нейронных структурах, так и о происходящих в них процессах. Например, одних только нейромедиаторов известно несколько тысяч видов;
- уровни 4-5 описаны на уровне структур, происходящие в них процессы не поняты и не имеют формального (воспроизводимого) описания;
- психофизиология не имеет средств детального наблюдения за соответствующими нейродинамическими процессами, а довольствуется интегральными оценками;

- структурные и функциональные компоненты уровней связаны между собой, имеет место генезис структуры в процессе функционирования, что отсутствует в компьютерных системах (синаптическая пластичность).

Из приведенного сопоставления становится понятным важности *системного подхода* при описании, анализе и проектировании «настоящих» систем. Современные попытки смоделировать высшую нервную деятельность человека можно проиллюстрировать достаточно простой аналогией: требуется разработать компьютерную систему («железо» и ПО) на основе имеющихся знаний. Известно:

- подробно о физике полупроводников, схемотехнике;
- фрагментарно о микроархитектуре;
- общие сведения о всем остальном – внешний вид, структура (из чего состоит), интегральное поведение, неизвестны – модели, методы, система команд, технологии программирования.

Общность программной и системной инженерии

Системная инженерия имеет много общего с программной инженерией. Похожесть заключается в методах и артефактах, различие – в предметных областях. Можно сказать, что наиболее развитые «тяжеловесные» методологии программной инженерии и сам свод знаний (SWEBOOK) стремятся реализовать принципы системной инженерии применительно к программной. И наоборот. Системная инженерия впитала в себя многое из программной. Рассмотрим кратко основные положения системной инженерии согласно [6-16].

Система. Иерархическая композиция элементов (холонов) по принципу *часть-целое*. Не является системой *классификация* (периодическая система элементов Менделеева) или *набор практик, рекомендаций* (система Станиславского). Для систем справедлив принцип *эмерджентности* – целое больше суммы ее частей.

Система, в свою очередь, является составной частью других систем. Системный подход классифицирует саму систему и ее окружение следующим образом:

- *целевая система* – собственно создаваемая система;
- *операционное (эксплуатационное) окружение* – системы, поддерживающие функционирование целевой системы или связанные с ней в процессе эксплуатации;
- *обеспечивающие системы*, создающие и поддерживающие систему в продолжении ее жизненного цикла, например, системы разработки и сопровождения ПО.

Системная инженерия является более общей дисциплиной, поэтому она занимается вопросами онтологии – фундаментальными сущностями системной (программной) инженерии. В основе лежат функциональные абстракции – «**альфы**» системной инженерии и ассоциаций между ними (рис.61-01). Альфы могут состоять из «**подальф**», а также иметь воплощение в **рабочих продуктах/артефактах**. Альфа имеет **состояния**, отражающие «прогресс» соответствующей абстракции системы (ALPHA - *Abstract-Level Progress Health Attribute*). Семь основных альф системной инженерии:

- **стейкхолдеры (заинтересованные стороны)** - лица, деятельность которых может иметь отношение к системе. Более широкая категория, нежели пользователи программной системы, включает в том числе и разработчиков;

- **возможности** – обстоятельства, которые обуславливают его разработку: требования пользователей, технологическое, финансовое и ресурсное обеспечение, наличие команды и ее квалификация, временные рамки и т.п.;
- **определение системы и воплощение системы** – проектное описание системы и ее реальная реализация «в материале». Для программных систем воплощение – процесс развертывания и исполнения кода системы. Определение системы в свою очередь состоит из трех подальф: **требования, архитектура и внеархитектурные составляющие** проекта;
- **работы** – деятельности, технологические процессы, связанные с ними метрика, планирование и операционный менеджмент;
- **команда и технологии.**

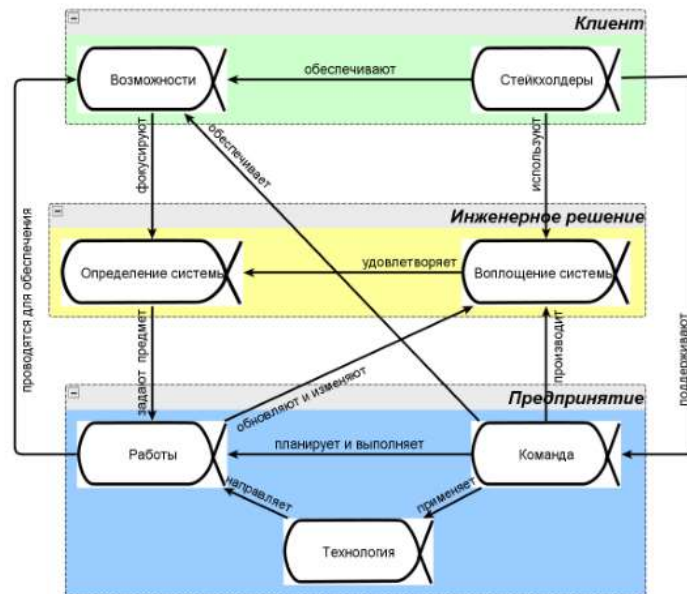


Рис.61-01. «Альфы» системной инженерии

Альфы, подальфы и рабочие продукты – не единственные составные элементы, которые используются для описания методологий разработки. Метамоделю (модель для описания моделей) закреплена в стандарте **ISO/IEC 24744:2014. Разработка программного обеспечения. Метамоделю для методик разработки**. Аналогичная модель доведена до языка спецификаций в документе OMG (Object Management Group) **SMSC/15-12-02 Ядро и язык методов программной инженерии** [6-26]. В нем определяется объектно-ориентированная модель (в виде диаграмм классов) для набора сущностей системного подхода: *альф, их состояний, рабочих продуктов (артефактов) деятельности и их пространств (видов), компетенций, практик, уровней детализации, паттернов*, а также их графическая и текстовая нотации. Фактически имеет место графическая (текстовая) среда описания методологии (рис.61-02).

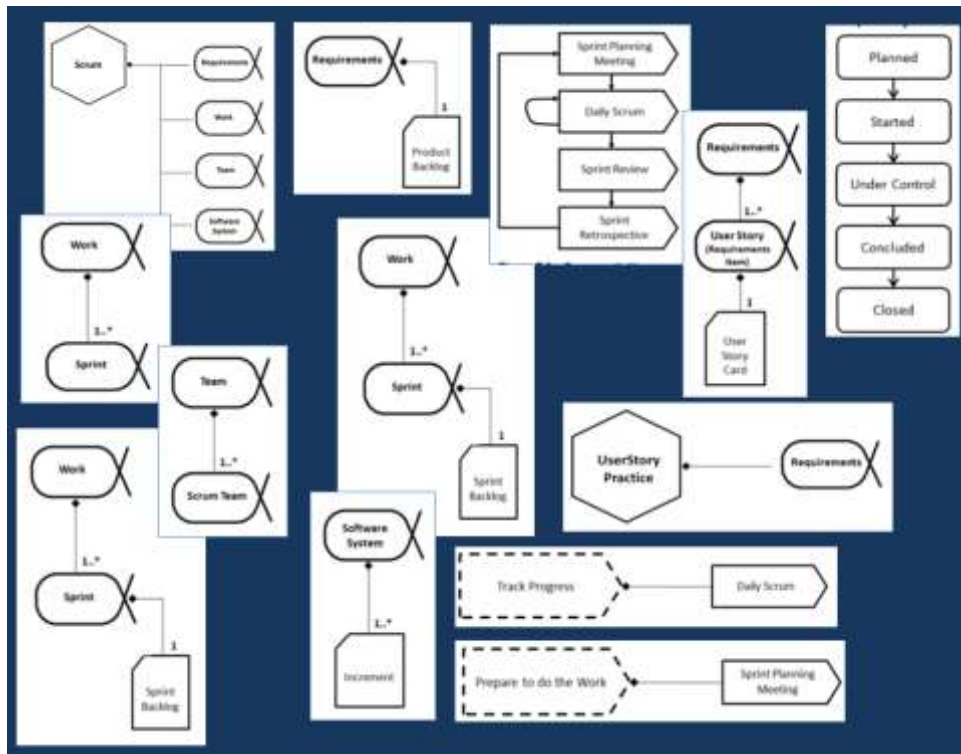


Рис.61-02. Описание методологии SCRUM в соответствии с SMSC/15-12-02.

Другой принцип, который роднит программную и системную инженерию – единство представления системы через множество *срезов/аспектов/инстанций*. В программной инженерии имеет место модель представления архитектуры в пяти аспектах - «4+1». Для технических систем используется представление в трех аспектах:

- компонент – функциональный аспект, функциональная схема, компоненты и их связи;
- модуль – продуктный аспект, физический элемент (модуль), воплощение компонента;
- размещение – аспект места (пространственный), размещение экземпляра модуля, схема размещения.

Описание системы в каждом из срезов/аспектов представляет собой иерархическую структуру вида «целое-части» и называется **разбиением** (breakdown). В

- функциональное разбиение (functional breakdown structure) - функциональная декомпозиция;
- разбиение работ;
- разбиение установки (размещения);
- разбиение документов и т.д..

Для технических систем с аспектами компонент/модуль/размещение разработаны стандарты (ISO/IEC 81346) для спецификации (именования) составляющих элементов трех указанных иерархий.

Из программной инженерии унаследовано понятие *жизненного цикла системы*, закрепленное в стандарте ISO/IEC 15288–2005. *Системная инженерия Процессы жизненного цикла систем*.

И, наконец, ключевым понятием системной инженерии является *архитектура*. Стандарт ISO/IEC 42010:2011. *Системная и программная инженерия. Описание архитектуры*. [6-

27]. Согласно ему, архитектура определяется как «*основные понятия и свойства системы в ее окружении, воплощенного в его элементах, отношениях и в принципах его дизайна и эволюции*». Архитектура является абстракцией (альфой), в то время как **описание архитектуры** – рабочий продукт (артефакт). Также в контекст описания архитектуры входит сама система, **окружение** и **стейкхолдеры (заинтересованные стороны)**, каждый из которых имеет **собственные интересы в системе** (рис.61-03).

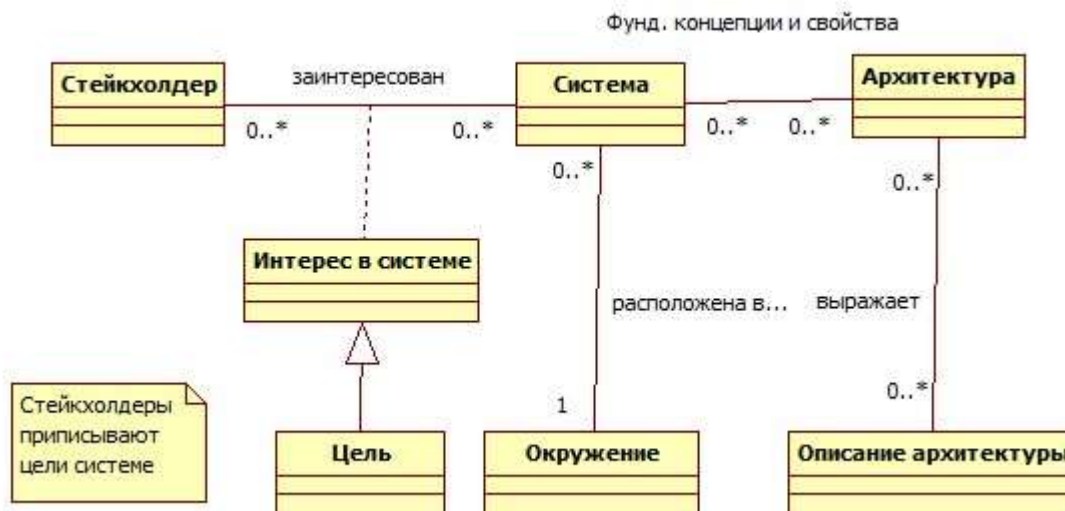


Рис. 61-03. Контекст описания архитектуры в ISO/IEC 42010:2011

В контексте описания архитектуры стандарт определяет (рис. 61-04):

- со стороны стейкхолдеров архитектура характеризуется **системой интересов**, по отношению к ее описанию у них имеют место **факторы беспокойства** (System concern). Это могут быть опасения относительно удовлетворения системой потребностей стейкхолдера, начиная от функциональных и заканчивая атрибутами качества, такими как надежность, производительность, доступность. Напомним, что разработчики также являются заинтересованными сторонами с собственными системными проблемами - факторами беспокойства, например, доступность информации, параллелизм, стоимость, график, модифицируемость, модульность, наличие тупиков, проблемы интеграции, доступность данных и пр.;
- описание архитектуры включает в себя ряд **обоснований, архитектурных видов и точек зрения**, архитектурный вид описывается рядом **моделей**, для которых определены виды (например, диаграмма классов или диаграмма состояний в UML);
- **архитектурный вид** (архитектурное представление) – рабочий продукт, выражающий отдельный *аспект/ипостась* системы (и ее архитектуры), соответствует принципу *множественности представлений* системы;
- **архитектурная точка зрения** – рабочий продукт для консолидации в архитектурном представлении конкретных проблем системы (факторов беспокойства), т.е. рассматривает архитектурное представление в связи с отношением к системе стейкхолдеров, но не прямо, а через обозначаемые ими проблемы;

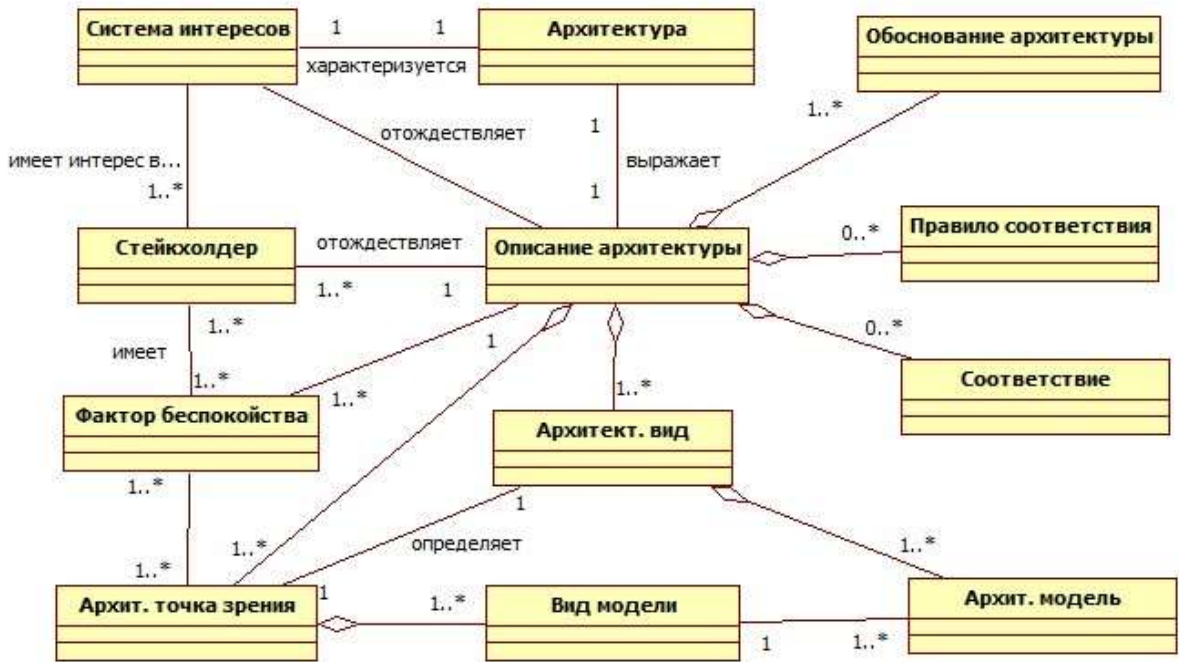


Рис. 61-04. Описание архитектуры в ISO/IEC 42010:2011

Следует заметить, что система имеет разнообразие архитектурных представлений в соответствии с отношением к ней заинтересованных сторон, в число которых входят все, кто имеет отношение к ее разработке. Применительно к программной инженерии – это «бизнес-архитектура», «системная архитектура» и «архитектура ПО».

С описание архитектуры связаны **системные решения**, но не прямо, а посредством реализации в них отдельных **элементов описания** (рис.61-05). Решения могут быть зависимы друг от друга, могут повышать одни факторы беспокойства и относиться к другим. Элементы описания должны быть согласованы через **соответствия**, которые определяются **правилами соответствия**.

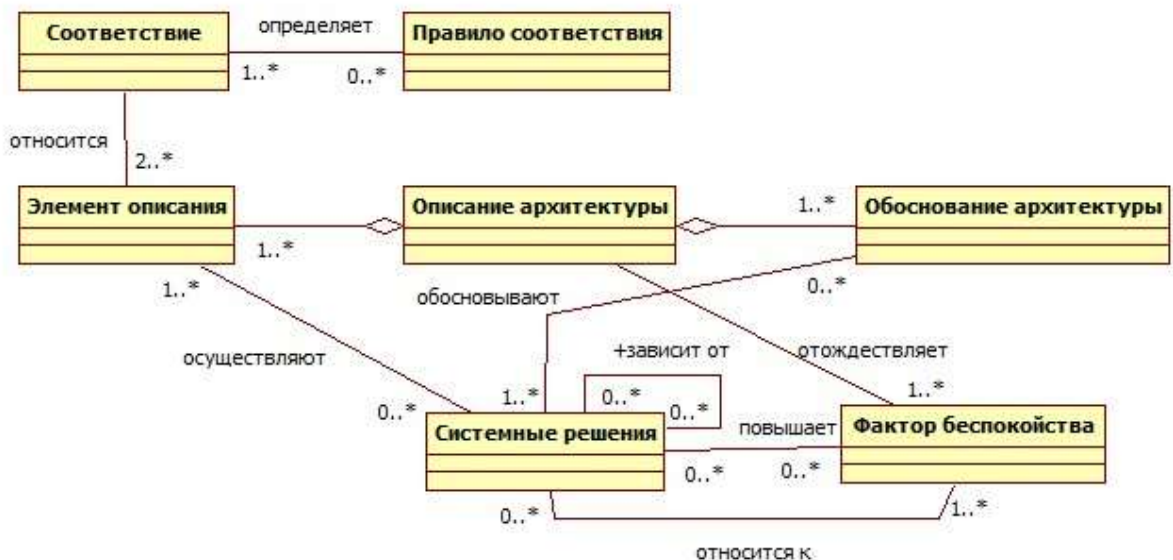


Рис. 61-05. Системные решения и обоснование архитектуры в ISO/IEC 42010:2011

И, наконец, в стандарте определены роль и место таких понятий как **каркас архитектуры (архитектурный фреймворк)** (рис.61-06) и язык описания архитектуры (рис.61-07).

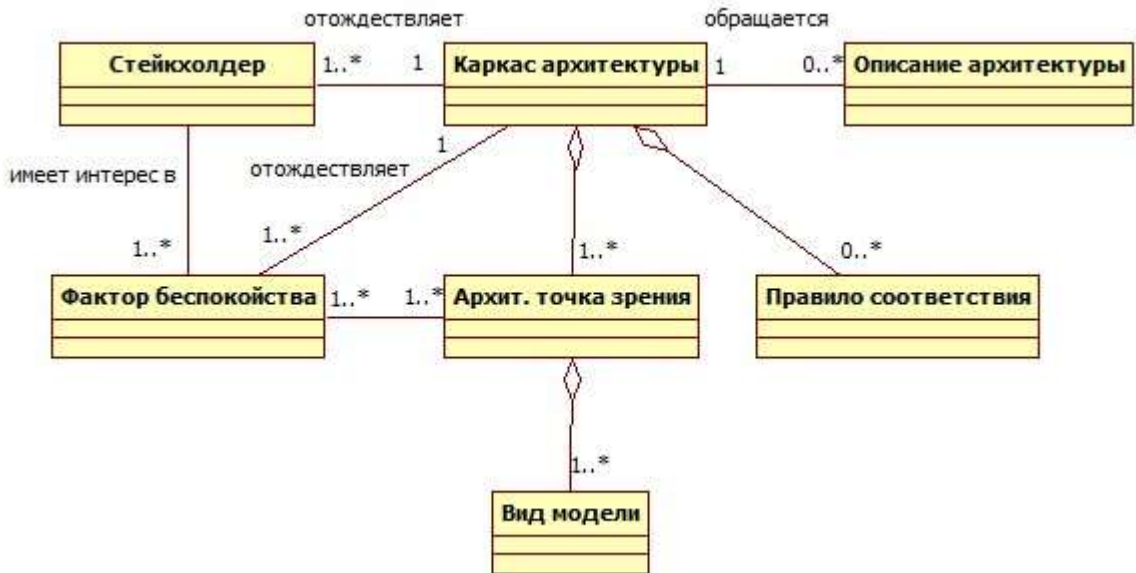


Рис. 61-06. Архитектурные каркасы в ISO/IEC 42010:2011



Рис. 61-06. Язык описания архитектуры в ISO/IEC 42010:2011