

Системная инженерия

Стандарты системной инженерии

Основные вопросы

- Терминология и онтология
- ISO 15288 – практики жизненного цикла системной инженерии
- ISO 42010 – архитектурное описание

Термины

Речевые сообщества и сообщества значений

Онтологии

ТЕРМИНОЛОГИЯ И ОНТОЛОГИЯ

Как описать?

- Естественный язык
- Язык профессиональных сообществ
- Терминология
- Как и почему были выделены термины →
онтология

Сообщества – 1/2

Semantics of Business Vocabulary and Rules – OMG SBVR <http://www.omg.org/spec/SBVR/> выделяет следующие сообщества [людей]:

- Речевые сообщества (speech communities):
 - Один естественный язык
 - Терминология конкретной предметной области. Обычные источники [в инженерии]:
 - Учебники
 - Непосредственное общение
 - Словарь определений стандарта (например: ГОСТ 34.320-96, ISO/IEC/IEEE 15288 etc.)
 - NB! Речевых сообществ для одного ЕЯ много → достичь однозначного соглашения по терминологии даже в области общих интересов очень трудно

Сообщества – 2/2

- Сообщества значений (semantic communities)
 - совокупность людей, которые одинаково понимают суть окружающих предметов и явлений

Значение, смысл, понятие

- Смысл [текста, сообщения, иной информации] определяется той ситуацией, в которой используется эта информация → прагматика: то, что надо делать, получив информацию
- Внеситуационная связь символов с объектами → семантика
- Термин – только слово
- То, что термином обозначается в реальном мире – понятие (concept)
- Если группа людей воспринимает мир через одинаковые понятия – это сообщество значений
- Если группа людей использует одинаковые термины – это речевое сообщество
- Сообщество значений всегда разбито на речевые сообщества

Согласование терминов – 1/2

- Терминологический фашизм – только один термин объявляется правильным, а все остальные — неправильными (ср. «Grammar nazi»()). Варианты проявления:
 - Требование безусловной единственности используемого термина. При этом синонимы исключаются
 - Требовать соответствия [произвольно отобранным] стандартам
 - Например: Только определённым ГОСТам, а не учебникам или другим ГОСТам
 - Требовать использования отечественного корня в слове
 - Пример: «шарокат» вместо «бильярд»
 - Настаивать на соблюдении традиций

Согласование терминов – 2/2

- Изоляция (~~терминологический пофигизм~~) – в начале каждого документа определяют список используемых обозначений и терминов
 - Нет «заведомо правильных вариантов» или ссылок на авторитетные источники
 - Часто в разных документах [одного проекта] используются разные термины
- Строгость значений с разрешением синонимии терминов, обозначающих одно понятие
 - В начале проекта долго договариваются о понятиях и обозначающих их терминах
 - Допускает использование любых терминов при условии, что адресат понимает, о чём речь

Онтология – определения

- Онтология:
 - Наука, отвечающая на вопрос «что есть в мире». Варианты: «учение о бытии», «учение о сущем»
 - Конкретный вариант ответа на этот вопрос
- Онтологическое описание – запись, выражение в какой-то терминологии онтологии. Грубо: состоит из объектов (понятий) и связей между ними
 - Часто путают онтологическое описание мира и саму онтологию

Онтология – зачем?

- Исторически, первыми создателями онтологий были философы:
 - Использование ЕЯ
 - Отсутствие четкой терминологии
 - Еще хуже: метафоричность, неоднозначность
- 20 век: разработчикам ИИ понадобился способ описания мира:
 - Однозначный
 - Интерпретируемый ЭВМ
 - Новое определение онтологического описания: разделяемое формальное описание/представление набора понятий (An ontology is a formal specification of a shared conceptualization)
- Конкретная онтология – это один из «взглядов» на мир. Разные онтологии рассматривают мир с разных сторон. Онтология как наука изучает способы, которыми даются эти ответы

Онтология vs. семантика

- Семантика описывает связь знаков / символов / терминов с означаемыми ими понятиями
 - Символ '\$' формально означает валюту «американский доллар» (USD)
 - «Что означает в тексте символ \$?» — это вопрос про семантику
- Онтологическая проблема отвечает на вопрос: «что такое американские доллары?» (или «что такое переменные окружения?»)
 - Есть ли такое понятие в мире как отдельная сущность, явление, находится ли это явление только в наших головах?
- Часто путают вопросы «что означает знак X [здесь]» и «что такое X [вообще]»

Бонус: онтики и онтологии

- Онтик – набор фактов о каком-то предмете/предметной области
- Онтология – набор фактов о мире
 - по крайней мере, о широкой предметной области

Описание мира

| Задача / вопрос | Решение / ответ |
|---------------------------------------|--|
| Как описать мир? | Этап 1. Выделить понятия Этап 2. Определить связи между понятиями |
| Как выделить понятия? | Этап 1. Набрать описание конкретных объектов Этап 2. Разделить их на группы Этап 3. Обобщить восприятие некоторых групп как понятие |
| Как определить связи между понятиями? | Замечание 1. Связь тоже может быть понятием (или представителем группы однотипных связей) Замечание 2. Наличие отношений между группами / понятиями является «группообразующим» |

Индивиды, классы

- Наиболее общий подход к группированию объектов – теория множеств
- Выделяют:
 - Сущности
 - Категории
 - Отношения между ними
- Все сущности/объекты в первую очередь делятся на:
 - Конкретные (индивиды)
 - Абстрактные (классы и отношения)
- Индивид – индивидуальный, уникальный объект, существующий в физическом мире
 - Экстенционализм (extensionalism): В физическом мире существуют объекты, имеющие место в пространстве-времени. Это место индивида в 4D называется «экстент» (extent)
 - За бортом остаются (?): социальные, энергетические, организационные объекты

Индивиды vs. классы

- Пример: Класс автомобилей $\{ x \mid \text{isCar}(x) \}$. Каждый автомобиль имеет экстент. Имеет ли Класс автомобилей экстент?
- Варианты ответа:
 - Нет, не имеет. Класс – абстрактный объект
 - Да, имеет, так как класс это множество, объединяющее свои члены

Классификаторы и разбиения

- С инженерной точки зрения, разбиение индивидов на категории, множества \neq разбиению индивидов на части
- Разбиение на категории, выделение множеств – классификация
- Разбиение на части – инженерное разбиение (breakdown)

Классификаторы

- Классификатор – описание в виде иерархических отношений класс – часть класса (дерево специализаций, отношения подмножества)
- Используется много классификаторов:
 - Разные взгляды на мир моделируются вариантами классификаторов
 - Разнообразие мира внутри одного взгляда моделируется таксономией

Классификаторы: примеры – 1/2

- Биологическая таксономия:
 - Жизнь
 - Домен
 - Царство
 - Тип
 - Класс
 - Порядок
 - Семейство
 - Род
 - Вид

Классификаторы: примеры – 2/2

- Борхес: Животные делятся на:
 - а) принадлежащих Императору,
 - б) набальзамированных,
 - в) прирученных,
 - г) молочных поросят,
 - д) сирен,
 - е) сказочных,
 - ж) бродячих собак,
 - з) включённых в эту классификацию,
 - и) бегающих как сумасшедшие,
 - к) бесчисленных,
 - л) нарисованных тончайшей кистью из верблюжьей шерсти,
 - м) прочих,
 - н) разбивших цветочную вазу,
 - о) похожих издали на мух.

Отношения

- Отношение – абстрактный объект, представленный парой объектов или группой из нескольких объектов
- Одно отношение это отношение между индивидами:
 - Является Матерью (Королева Елизавета, Принц Чарльз) – предикатный вид
 - Является Матерью: Королева Елизавета → Принц Чарльз – функциональный вид
- Класс отношений:
 - МАТЕРИНСТВО = $\{ \langle x, y \rangle \mid \text{Является Матерью}(x, y) \}$

Классификация vs. Отношения

- Наиболее распространенные отношения:
 - IS_A: Классификация: основа классификаторов (таксономий)
 - IS_PART_OF: Часть-целое: инженерное разбиение
 - INHERITS_TO: Класс-элемент: ООП
 - CONNECTED_TO: Присоединен(элемент1, элемент2): построение сложных объектов

Примеры онтологий – 1/3

- Стандарт ISO 15926 – стандарт единого представления данных для взаимодействия независимых ИС
 - Предметная область: установки непрерывного производства. Стандарт описывает сведения о ЖЦ установки непрерывного производства целиком
 - Рождён в нефтеперерабатывающей отрасли
 - Опирается на идеи семантических сетей и Resource Description Framework
- ISO 15926 представлен в виде комплекта отдельно публикуемых частей:
 - ISO 15926-1 Обзор ISO 15926
 - ISO 15926-2 Модель данных
 - Описывает обобщенную концептуальную модель данных, поддерживающую представление всех аспектов ЖЦ установки непрерывного производства. Онтология поддерживает все группы описаний, которыми могут обладать по отношению к установке инженеры-технологи, инженеры по оборудованию, операторы, инженеры по техническому обслуживанию и другие специалисты
 - ISO 15926-3 Справочные данные, касающиеся геометрии и топологии
 - Описывает геометрию и топологию, в терминах языка OWL переопределяет геометрические конструкторы
 - etc.
- РФ:
 - ГОСТ Р ИСО 15926-1-2008 введен в действие 01.01.2010
 - ГОСТ Р ИСО 15926-2-2010 введен в действие 31.08.2011

Примеры онтологий – 2/3

- Онтология IDEAS (International Defence Enterprise Architecture Specification for exchange Group) – используется для онтологического представления для инженерных архитектурных описаний стран NATO
 - http://dodcio.defense.gov/TodayinCIO/DoDArchitectureFramework/dodaf20_ontology1.aspx
 - Содержит описания:
 - Систем – коммуникационных, компьютерных сетей, РИС
 - Информационных связей между системами
 - Спецификации информации (Information specifications) – объектные модели (с указанием ограничений безопасности) обмена через информационные связи
 - Платформы, оборудование и здания (platforms & facilities)
 - Системные и прикладные функции (system & operational functions)
 - Люди и организации
 - Метаданные: владелец архитектуры, разработчик, наименование, версия, etc.

Примеры онтологий – 3/3

- JC3IEDM (The Joint C3 Information Exchange Data Model) – онтология, предназначенная для объединения C3 систем управления войсками во время объединенных операций
- C3 – Consultation, Command and Control – тип систем управления войсками, обеспечивающий управление, сбор данных о выполнении команд и согласование действий
 - C4I
 - Computers (Technology focus) or
 - Computing (Human activity focus)
 - C4I2 – Command, Control, Communications, Computers, Intelligence, and Interoperability

ISO 15288 – ПРАКТИКИ ЖИЗНЕННОГО ЦИКЛА СИСТЕМНОЙ ИНЖЕНЕРИИ

Общие понятия

- ISO/IEC/IEEE 15288:2015 – Systems and software engineering – System life cycle processes – описывает процессы и этапы ЖЦ [любой системы]
 - Разработка стандарта началась, когда осознали необходимость обобщения Systems Engineering process framework
 - Ранее применялся стандарт MIL STD 499A от 1969 года

Особенности – 1/2

- Представляет общую всестороннюю интегрированную структуру (framework) для описания управления (managing) полным ЖЦ систем. При этом:
 - Пригодно для организаций любого размера: малых, средних и больших
 - Может быть использовано для внутреннего использования и как база для взаимодействия с субконтракторами
 - Применима для любой области
 - Применима для любой модели ЖЦ
- Определяет набор процессов, концепций и связанную с ними терминологию
 - Может быть применена на любом уровне иерархии системы в течение всего ЖЦ
 - Не является последовательной или «однонаправленной»
 - Допускает параллельное выполнение процессов и этапов

Особенности – 2/2

- Применима к системам, созданным человеком (man-made 0_0), «собранным» из [одного или более] элементов типа:
 - Hardware, software, data, humans, processes (например, процесс предоставления пользователю услуги), procedures (например, инструкция оператора), facilities, materials and naturally occurring entities
- Сосредотачивается на «что», а не на «как»
- Включает процесс подстройки (tailoring process)
- Включает отсылку на систему систем (System of Systems – SoS)

Этапы ЖЦ

- Разработка концепции (concept)
- Проектирование (development)
- Производство (production)
- Использование (utilisation)
- Поддержка (support)
- Вывод из эксплуатации (retirement)

Процессы – 1/4

- Категории процессов
 - Технические (Technical)
 - Управления проектом (Technical management)
 - Agreement
 - Организационные (Organisational project-enabling)
- Организационные процессы:
 - Управление моделью (?) ЖЦ (life cycle model)
 - Управление инфраструктурой
 - Управление портфолио (Portfolio management)
 - Управление персоналом (Human resource management)
 - Управление качеством
 - Управление знаниями (?) (Knowledge management)

Процессы – 2/4

- Технические:
 - Определение необходимости создания / модификации (Business or mission analysis process)
 - Определение потребностей и требований СХ (stakeholder needs & requirements definition)
 - Определение системных требований
 - Описание архитектуры (Architecture definition)
 - Низкоуровневое проектирование (Design definition)
 - Анализ системы

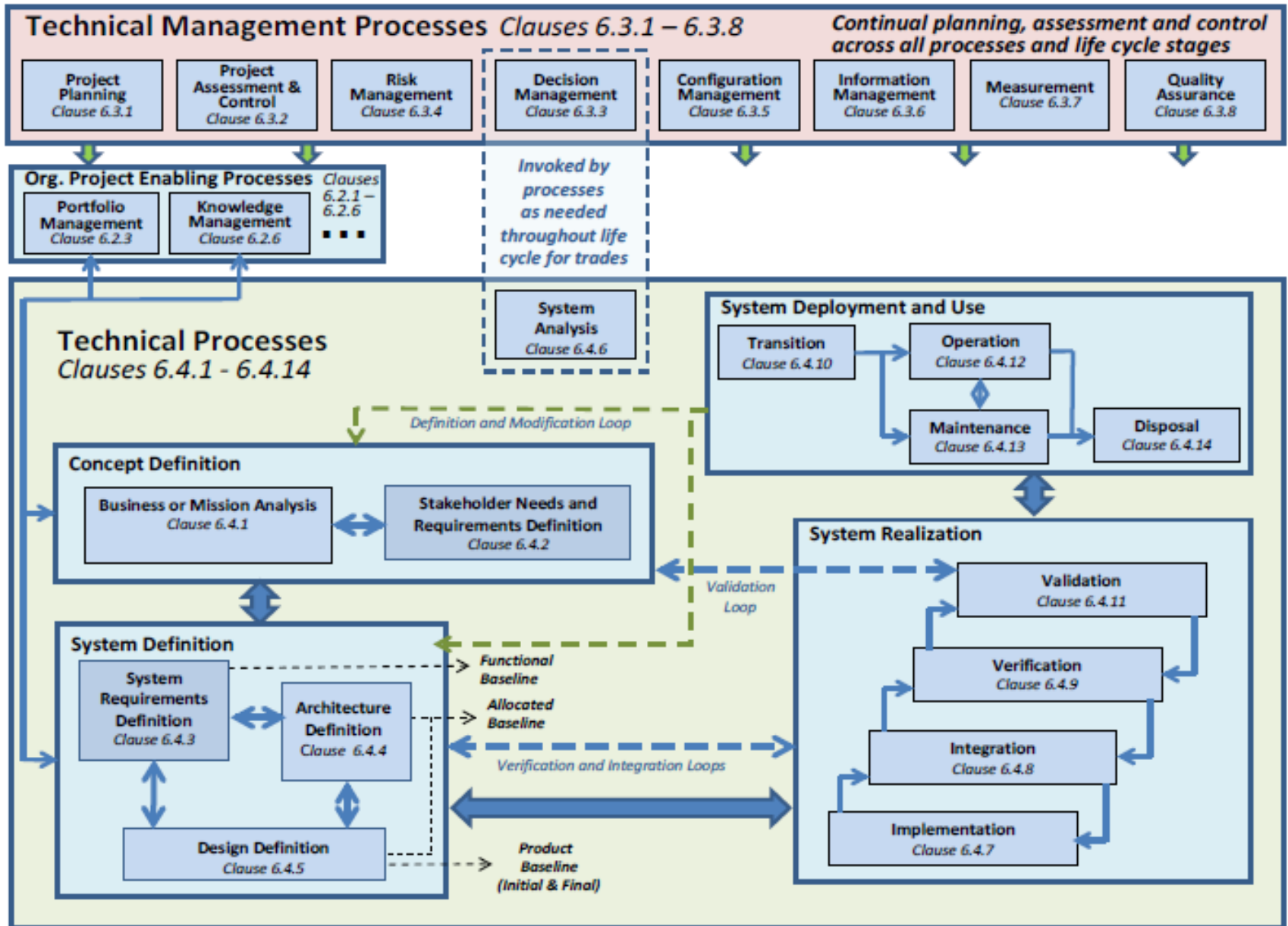
Процессы – 3/4

- Технические (продолжение):
 - Воплощение (implementation)
 - Интеграция
 - Верификация
 - Передача заказчику (?) (transition)
 - Валидация
 - Использование (operation)
 - Поддержка / сопровождение (maintenance)
 - Вывод из эксплуатации (disposal)

Процессы – 4/4

- Технического управления:
 - Планирование проекта
 - Оценка проекта и управление (project assessment and control)
 - Управление принятием решений (decision management)
 - Управление рисками
 - Управление конфигурациями
 - Управление информацией (Information management)
 - Measurement process
 - Управление качеством (Quality Assurance)
- Согласования (agreement) :
 - Acquisition process
 - Supply process

Взаимосвязь процессов



Особенность стандарта

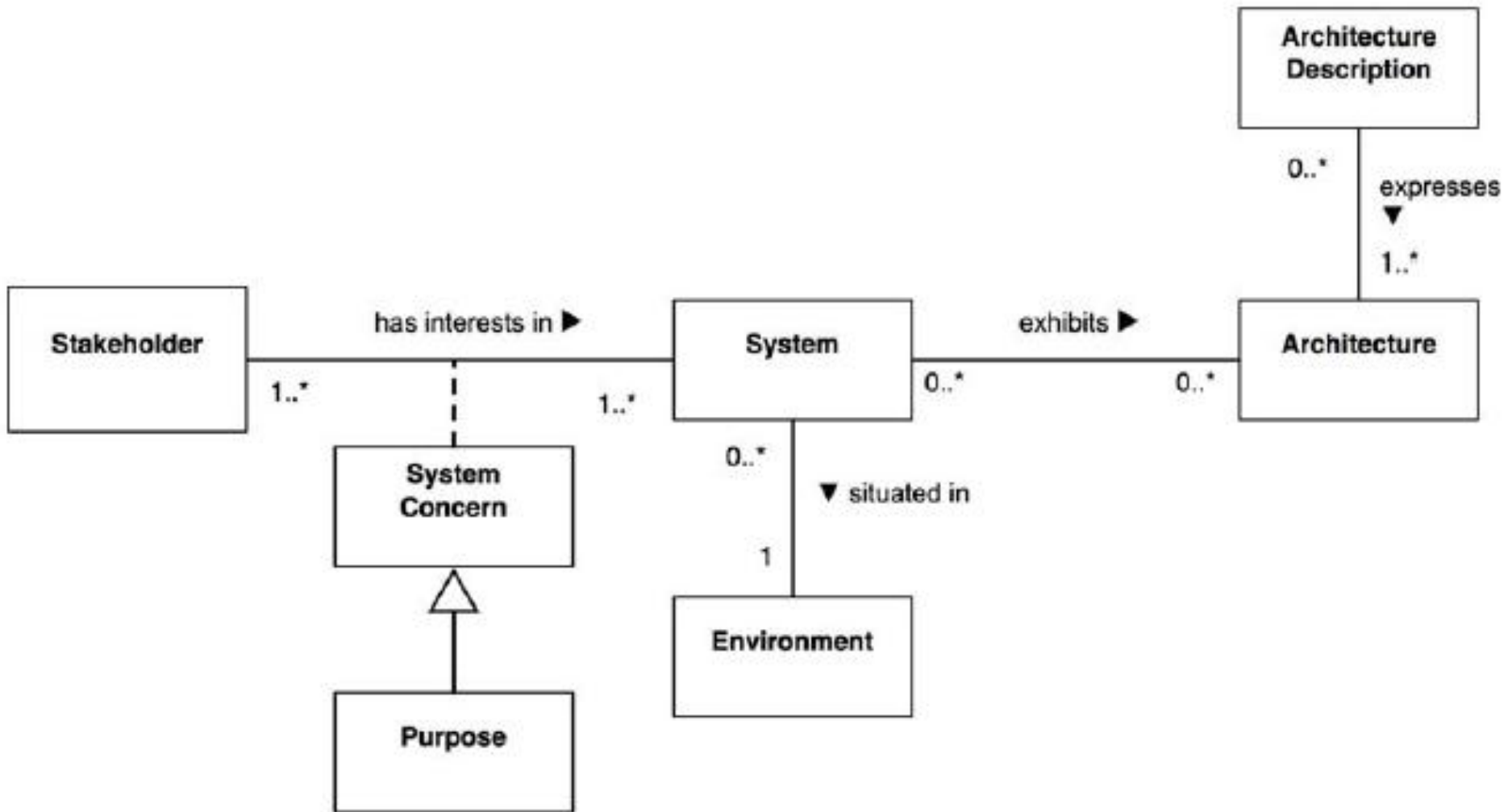
- Для каждого процесса определены:
 - Назначение
 - Результаты
 - Операции / деятельность при реализации (activities)
- Стандарт обобщает 25 процессов, 123 результата 403 операций

ISO 42010 – АРХИТЕКТУРНОЕ ОПИСАНИЕ

Общие понятия

- ISO/IEC/IEEE 42010:2011 – Systems and software engineering — Architecture description – определяет требования к описанию системной, программной и enterprise архитектур
 - Стандартизует практики описания архитектур, задавая термины, представляя концептуальные основы отображения архитектуры (expressing), ее обсуждения и reviewing
 - Задаёт требования, применимые к описаниям архитектур, архитектурных frameworks и языков описания архитектур
- NB! Различаются архитектура и описание архитектур

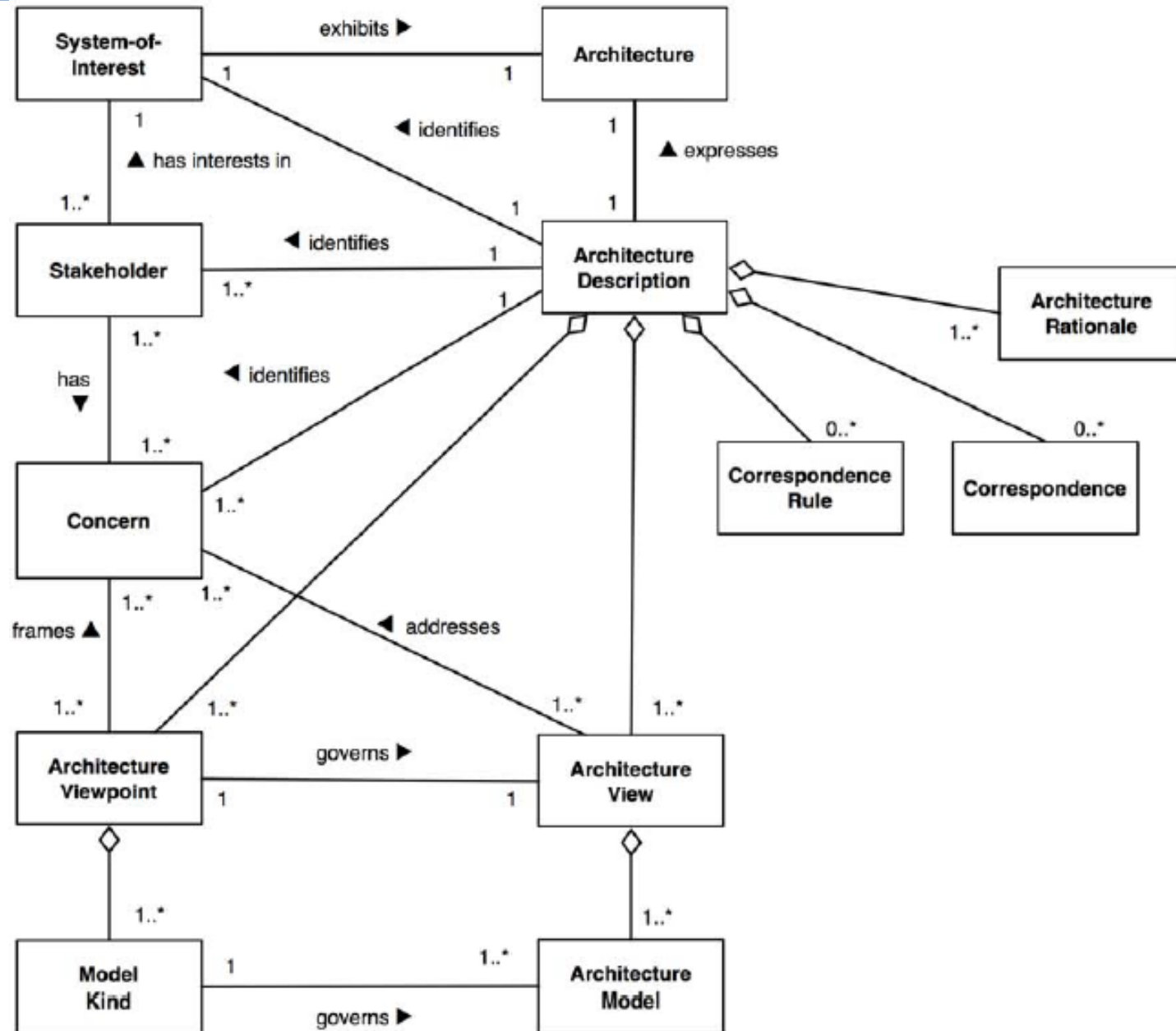
Связь архитектуры, системы и СХ



Описание архитектуры

- Описывает (expresses) архитектуру
- Идентифицирует целевую систему (system of interest)
- Идентифицирует 1 или более СХ
- Обозначает (identifies) 1 или более concerns (about the system of interest)
- Включает 1 или более сторон рассмотрения архитектуры (viewpoints) и 1 или более architecture views
- Включает 0 или более отношений
- Включает 0 или более правил отношений
- Включает 1 или более обоснований архитектуры

Состав архитектуры



Architecture viewpoint – 1/7

- Часть архитектурного описания
- Покрывает (frames) 1 или более concerns СХ, касающихся целевой системы
- Задаёт (governs) точно один architecture view – оно должно соответствовать
- Состоит из 1 или более типов моделей

Architecture viewpoint: состав – 2/7

- Включает несколько разделов. Некоторые могут быть опущены:
 - Имя Viewpoint
 - Включает также синонимы
 - Общее описание (overview)
 - Введение или краткое описание, включая ключевые особенности такой точки рассмотрения
 - Concerns and «anti-concerns»
 - Список важных особенностей (беспокойств 0_0 ?)
 - Помогает создателю архитектуры принять решение о необходимости включения viewpoint в архитектуру для данной системы
 - «Anti-concerns» полезны для исключения из рассмотрения ненужных моделей, неподходящих для данной системы

Architecture viewpoint: состав – 3/7

- Типичные CX (список CX, которым интересны views из данного viewpoint)
 - Примечание: Разработчик архитектуры должен описать какие concerns относятся к каждому CX
- Типы моделей (описаний) (Model kinds)
 - Общее описание типа моделей (Introduction)
 - Идентификация типа
 - Описание языков, нотации, соглашений, методик моделирования (conventions, language or modelling techniques)
 - могут быть описаны с использованием метамодели для вида модели, который определяет структуру и соглашения для ее моделей
 - Описываются ресурсы, которые будут доступны в Viewpoint и словари построения view. Также описываются аналитические методы и/или другие операции, которые будут использоваться на моделях этого вида
 - Примечание: Описание типа моделей не стандартизовано. Можно описать, например, с помощью:
 1. Метамодели
 2. Шаблона модели для заполнения пользователями
 3. Языком описаний или ссылкой на существующий язык описаний моделей
 4. Комбинацией способов 1...3

Architecture viewpoint: тип модели – 4/7

- Описание типа моделей с помощью:
 - Метамоделли:
 - Метамодель описывает элементы AD соответствующие словарю типа модели. Метамодель задает для моделей описываемого типа:
 - Сущности: Какие основные типы элементов могут присутствовать в моделях?
 - Атрибуты: Какими свойствами описываются эти сущности?
 - Отношения: Какие типы отношений между сущностями могут описываться?
 - Ограничения: Какие типы ограничений, наложенных на сущности, атрибуты и отношения, могут описываться?
 - Сущности, атрибуты, отношения и ограничения являются элементами архитектурного описания
 - Примечание: Если viewpoint описывает несколько типов моделей, то бывает полезно описать отдельный viewpoint специально для описания всех типов моделей с помощью метамodelей. Кроме того, часто полезно использовать одну метамодель для выражения нескольких связанных точек зрения (например, при определении архитектурного фреймворка (architecture framework))
 - Шаблона (шаблонов):
 - Задает шаблон (template) или форму, определяющие формат и/или содержимое моделей описываемого типа

Architecture viewpoint: тип модели – 5/7

- Описание типа моделей с помощью (продолжение):
 - Языка описаний
 - Идентифицирует существующую нотацию или язык описания моделей (model language)
 - Иначе описывает новый язык специально для моделей описываемого типа: синтаксис, семантику, поддерживающие [программные] средства
 - Операций
 - Описывает типы операций доступных с моделями данного типа

Architecture viewpoint: состав – 6/7

- Зависимости (Correspondence rules)
 - Описывает зависимости, описанные в данном viewpoint или в использованных типах моделей. Обычно такие зависимости будут «меж – модельными» или «меж – view» так как ограничения типа моделей описаны как часть описания типа моделей
- Операции с views
 - Методы, применяемые к представлениям или их моделям. Операции можно разделить на категории:
 - **Методы создания** могут описываться в виде:
 - Процедурных руководств по основным шагам реализации (process guidance): как начать, что делать дальше
 - Руководство по содержанию (work product guidance): шаблоны для представлений этого типа
 - Эвристика, стили, шаблоны или другими способами
 - **Методы интерпретации** задают способы обеспечить облечение понимания представлений читателем (CX)
 - **Методы анализа** используются для проверки, анализа, преобразования, прогнозирования, применения и оценки архитектурных решений из этого представления
 - **Методы проектирования или реализации** используются для создания (конструирования) системы, используя архитектурные решения из этого представления

Architecture viewpoint: состав – 7/7

- Примеры
 - Данный раздел содержит примеры, то есть любую дополнительную информацию, полезную читателям (СХ)
- Источники
 - Описывает источники информации данного viewpoint, если они есть, включая авторство, историю, ссылки на литературу, предшествующий уровень техники (решений)

Architecture view

- Часть описания архитектуры
- Задается одним из architecture viewpoint
- Отражает одно или более concerns заданных СХ
- Состоит из 1 или более архитектурных моделей

ISO 24744 – ОПИСАНИЕ МЕТОДОВ РАЗРАБОТКИ

Общие понятия

- ISO/IEC 24744:2014 Software engineering – Metamodel for development methodologies –
- Development methodologies may be described in the context of an underpinning metamodel, but the precise mechanisms that permit them to be defined in terms of their metamodels are usually difficult to explain and do not cover all needs. For example, it is difficult to devise a practice that allows the definition of properties of the elements that compose the methodology and, at the same time, of the entities (such as work products) created when the methodology is applied. ISO/IEC 24744:2014 introduces the Software Engineering Metamodel for Development Methodologies (SEMDM), a comprehensive metamodel that makes use of a new approach to defining methodologies based on the concept of powertype. The aim of SEMDM is to define methodologies in information-based domains, i.e. areas characterized by their intensive reliance on information management and processing, such as software, business or systems engineering. The SEMDM combines key advantages of other metamodeling approaches with none of their known drawbacks, allowing the seamless integration of process, modelling and people aspects of methodologies. Examples are given where other metamodels are mapped to SEMDM and a brief synopsis of problems is provided.
- Various methodologies are defined, used, or implied by a growing number of standards, and it is desirable that the concepts used by each methodology be harmonized. A vehicle for harmonization is the SEMDM. Conformance to this metamodel will ensure a consistent approach to defining each methodology with consistent concepts and terminology.