

Натуральные основания и методы описания систем

А.В. Зорин

Аннотация— В работе представлен новый системный язык, который характеризуется обоснованной конструкцией исходя из фундаментальных понятий.

Проанализирован ряд языков системных описаний и моделирования, в частности, UML, SysML, BPMN, TOGAF и др., используемых настоящее время при проектировании систем различного назначения. Отмечен ряд проблем этих языков, в том числе ориентированность на специальные области применения, большое количество элементов и схем с отсутствием их обоснования, слабость отражения аспектов времени и пр.

Также проведён анализ сложностей при документировании систем, в том числе противоречий в терминологии крупных проектов, в первую очередь из-за расхождений в глоссариях используемых стандартов.

Цель работы определена как создание системного языка, основанного на фундаментальных понятиях (соответственно с минимумом внутренних противоречий), имеющего наряду графическим представлением (как методом наглядного описания) также текстовое (для подготовки и машинной обработки документов), а также формализованные правила связывания текстовых конструкций между собой (как инструмент обоснования и построения терминов).

Предложен метод описания систем, базирующийся на базовых естественно-научных понятиях: сущность, взаимодействие, пространство, время, атрибут, отношение. Представлен язык описания систем NSL (Natural System Language, природный язык систем) в текстовом (NSL-T), формульном (NSL-F) и графическом изложениях (NSL-G). Продемонстрировано последовательное инкрементное построение глоссария по разделам Основные объективные понятия, Основные субъективные понятия и далее до разделов Система и Программа.

Демонстрация результатов предлагаемого подхода выполнена на сравнении наименований и описаний наиболее употребительных диаграмм языка UML.

Приведены возможные применения разработанного метода и языка NSL для моделирования систем и построения терминологии, а также в области искусственного интеллекта.

Ключевые слова— системный анализ, моделирование, документирование, язык, искусственный интеллект.

I. ВВЕДЕНИЕ

Данная работа появилась на основании анализа и обобщения опыта проектирования различных сложных технических и организационных систем и создания соответствующей документации.

Статья получена 27 октября 2024.

А.В. Зорин – независимый исследователь, кандидат технических наук, эксперт ООН / МАГАТЭ (email: zorin@ase.ru)

1.1. ЯЗЫКИ ПРОЕКТИРОВАНИЯ СИСТЕМ

В настоящее время при проектировании систем различного назначения широко используется ряд языков системных описаний, в частности:

- UML (Unified Modeling Language, унифицированный язык моделирования) – графический язык, предназначенный в первую очередь для разработки программного обеспечения и системного проектирования [1]-[2];

- SysML (Systems Modeling Language, язык моделирования систем) – предметно-ориентированный язык графического моделирования систем, прежде всего аппаратно-программных комплексов [3]-[4];

- BPMN (Business Process Model and Notation, модель и нотация бизнес-процессов) – графический язык моделирования бизнес-процессов [5]-[6].

Преимуществом предлагаемых ими описаний задач и решений различных предметных областей является наглядность. В сочетании с формально определёнными правилами языков (типизация объектов, их соединителей и пр.) такие описания получают средства поддержки и проверки корректности.

Ни в коей мере не умаляя достоинств этих языков, необходимо в то же время отметить, что практика их применения выявила у них ряд недостатков, например:

- UML по логике описаний (классы, наследование и пр.) более близок программистам; он содержит обширный набор диаграмм, трудно объяснимый по своему составу (с точки зрения полноты и достаточности); логика соединителей также достаточно непрозрачна;

- SysML преодолевает узость программистского взгляда и более удобен для описания технических систем, он более лаконичен и эффективен для взаимопонимания соответствующих специалистов, что одновременно сужает его применение для других дисциплин (хотя такие примеры имеются);

- BPMN содержит большой набор элементов, позволяя отражать практически все аспекты бизнес-процессов и документооборота, но это же приводит сложности в его освоении из-за обилия элементов; кроме того, он плохо применим в сложных задачах из-за ряда встроенных приёмов оформления (например, необходимости вынесения каждого участника бизнес-процесса в отдельную дорожку, что делает большие схемы необозримыми при их слабом наполнении).

Понимание проблем этих языков привело к появлению других подходов и дополнительных инструментов. Например, в TOGAF (The Open Group Architecture Framework, фреймворк архитектуры The Open Group) включены важные для проектирования архитектур предприятий аспекты, начиная с явного включения бизнес-целей в описательные схемы верхнего уровня и дальнейшего их поэтапного развития в проектирование, планирование и внедрение IT-архитектуры предприятия [7].

Существенные шаги по сокращению необоснованного разнообразия элементов привели к появлению такой спецификации как ArchiMate (Architecture-Animate, архитектура оживленная) – язык моделирования архитектуры предприятия для поддержки описания, анализа и визуализации архитектуры и бизнес-процессов [8]. Достижение цели ArchiMate – «быть настолько лаконичным, насколько это возможно», можно проиллюстрировать тем, что если в UML и BPMN содержат около соответственно 150 (UML) и 250 (BPMN) сущностей, то ArchiMate использует 40 основных элементов, 18 дополнительных и 11 типов связей. При этом базовых типов элементов всего 6, и они модифицированы на 7 иерархических уровнях (от стратегии предприятия до программно-технической реализации), что резко улучшает понимание языка.

Концептуально упрощение типизации было, видимо, задумано в спецификации MOF (Meta-Object Facility, мета-объектное средство) с его ещё большей абстракцией базовых элементов и строгой архитектурой метамоделирования [9].

Несколько в стороне находится язык OWL (Web Ontology Language) с его целью поддержать сетевые онтологии для автоматической обработки и интеграции машинами информации, доступной в сетях [10]. В нём, в отличие от предыдущих спецификаций, была явно выделена задача семантического связывания и агрегации информации (в предыдущих концепциях эта задача по умолчанию считалась решённой).

Из существенных недостатков всех перечисленных спецификаций является слабость отражения аспекта времени, хотя, например, SysML как язык проектирования систем нуждается в достаточно ясном его описании и операциях с ним (скажем, для систем реального времени, хотя практически все системы в той или иной степени – системы реального времени). Проблема, видимо, состоит в том, что время – сложное философское понятие, поэтому дальше диаграмм последовательности в большей части продвижения нет (при изобилии любых других диаграмм – классов, компонентов, структур и т.п.), как и нет методов его представления кроме проприетарных инструментов пошагового шагового моделирования (например, [11]).

1.2. ЯЗЫКИ ДОКУМЕНТИРОВАНИЯ СИСТЕМ

Понятно, что рассмотренные языки проектирования / моделирования создают серьёзный задел для описания систем. Однако в целом документация проектов гораздо шире, чем язык графических диаграмм – например, при создании систем управления для АЭС она в первую

очередь текстовая: начинается со спецификации требований, затем реализуется в технических описаниях на всём жизненном цикле создания и заканчивается документацией по эксплуатации и т.п. [12].

При этом квалифицированные специалисты сталкиваются с проблемой противоречий в терминологии (гlossариях) документов в одном проекте, если смотреть глубже – в гlossариях стандартов, используемых в проекте при его разработке. В больших проектах стандартов достаточно много (например, в проекте АЭС их несколько тысяч), при этом в таких проектах почти всегда действует несколько групп стандартов даже в рамках одной дисциплины – это могут быть стандарты ГОСТ, МЭК, МАГАТЭ и пр. Попытки их гармонизировать безрезультатны, т.к. они не имеют общей словарно-понятийной базы (для чего и создаются метаязыки).

Эта неопределённость в терминологии начинается с самых основ проектирования систем – с того, с чего (после определения требований) начинается уже собственно разработка: понятия архитектуры.

Сравним предлагаемые определения:

1) IEC 61513-2011 [12], ГОСТ Р МЭК 61513-2020 [13]:

3.27 I&C architecture organizational structure of the I&C systems of the plant which are important to safety.	3.27 архитектура СКУ (I&C architecture): Организационная структура СКУ станции, которые являются важными для безопасности.
---	---

Определение очевидно почти дословно коррелирует со словарём МАГАТЭ (далее).

2) IAEA Nuclear Safety and Security Glossary [14]:

Первоисточник: architecture Organizational structure of the instrumentation and control systems of a plant that are important to safety.	В переводе на русский: архитектура Организационная структура важных для безопасности систем контроля и измерительных приборов станции.
--	--

Очевидные замечания:

- традиционное ошибочное смешение понятий «структура» и «архитектуры»;
- неудачный перевод «организационная структура» (по-русски «оргструктурой») называют структур подразделений и должностную / кадровую структуру предприятия).

3) ГОСТ Р 57100-2016 [15]:

«3.2 архитектура (системы) (architecture): Основные понятия или свойства системы в окружающей среде, воплощенной в ее элементах, отношениях и конкретных принципах ее проекта и развития».

4) IEEE Std 1471-2000 IEEE [16]:

Первоисточник: 3.5 architecture The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.	В переводе на русский: 3.5 архитектура: Фундаментальная организация системы, воплощенная в ее компонентах, их взаимоотношениях друг с другом и с окружающей средой, а также принципы, определяющие ее дизайн и эволюцию.
--	--

Это определение представляется наиболее удачным – оно соответствует смыслу того, что подразумевается под архитектурой в строительстве. Отметим, что определение архитектуры в [15] восходит к [16], однако является неточным переводом – словосочетание «в окружающей среде, воплощенной в ее элементах» мало осмыслено.

С концептуальной точки зрения, подход последних определений близок к понятной концепции описания и проектирования систем IDEF (Integration Definition for Function Modeling, определение интеграции для моделирования функций) [17]. В ней проектирование систем начинается с представления её в диаграмме контекста (context diagram), где указываются все внешние связи (условия, объекты, взаимодействия), далее описание детализируется внутрь, вводя элементы / подсистемы описываемой системы, с передачей им внешних связей.

Эти замечания имеют характер не лингвистических придинок: в результате неясности в терминологии, при проектировании систем упускаются важные аспекты технических документов, вплоть до целых разделов, документация неверно структурируется, влияя негативно на порядок работ и целостность проекта, и т.п.

В период 1990-2010 годов в стандартах, связанных с проектированием систем, имели место ещё большие расхождения, вплоть до документов внутри одной организации (МЭК, МАГАТЭ и пр.). В период 2010-2020 самые существенные расхождения были преодолены путём издания глоссариев каждой организации в одном первоисточнике, но существенные расхождения между группами стандартов остались.

Ещё большим объём расхождений становится при включении в проекты документов ИСО, которые прямо пересекаются и не совпадают со специальными техническими стандартами, например, в областях управления требованиями, конфигурацией и пр., формально доминируя над последними ввиду общих требований ИСО по управлению качеством (например, [18]).

II. ЦЕЛЬ РАБОТЫ

Целью настоящей работы является построение языка, который характеризовался бы следующим:

1) Он должен иметь ясную, обоснованную конструкцию исходя из фундаментальных понятий (как аксиоматика геометрии) и соответственно минимум внутренних противоречий;

2) Язык должен допускать не только графические представления, но и текстовые:

- При разработке больших описаний требуется машинная обработка, а графические представления могут быть обработаны только при их переводе в машинно-читаемый формат (это предлагается в ряде случаев их записью в XML или эквивалентных нотациях);

- Язык текстовых описаний априори понятен заведомо большему широкому кругу специалистов одного проекта; кроме того, техническая документация

должна оперировать текстовыми конструкциями в связке с иными описаниями;

3) Правила связывания текстовых конструкций между собой должны быть также формализованы, иначе это связывание будет субъективным и сомнительным.

Такой язык может использоваться для формирования обоснованного аппарата описания и моделирования систем.

Важная сторона его возможных приложений – формирование текстовых документов. Для выбора обоснованной базы и правил глоссариев, на основе такого языка может быть проведена гармонизация терминологии, в т.ч. в стандартах, за счёт выхода на уровень метаязыка.

Возможны также актуальные приложения в области искусственного интеллекта (ИИ) – в части формализации знаний, в том числе сетевых. В целом основная часть методов ИИ восходит ещё к понятию перцептрона [19], т.е. сопоставлению и узнаванию образов машинами, их обучению на основании фильтрации экспертами предварительно найденных связей и закономерностей, в сочетании с эвристическими инструментами различного рода. Терминологический хаос обычных первоисточников заметно превосходит уровень разночтений в терминологии нормативных и технических документов, что может приводить ИИ к неверным заключениям.

Выходом из этого представляется тегирование информационных объектов словарём метаязыка. При наличии формальных правил связки этих тегов, объекты можно будет гораздо более достоверно категорировать и связывать между собой, в том числе расширяя применение ИИ на области, важные для безопасности (промышленность, медицина и т.п.).

III. ПРЕДЛАГАЕМЫЙ ПОДХОД

Предлагается минималистическая схема системных описаний. Базис включает то, что является нашими встроенными представлениями и навыками, позволяющими нам составлять рассуждения (в частности, составить этот материал).

Описания, как правило, максимально простые, что достигается поэтапным дедуктивным введением понятий и записью последующих с добавлением, как правило, одной характеристики (инкрементный подход).

Метод описания систем, именуемый NSL (Natural System Language, природный язык систем), включает:

- NSL-T – текст;
- NSL-F – формулы;
- NSL-G – графика.

Дальнейшие описания даются без развёрнутых пояснений для сокращения объёма материала.

IV. NSL-T – ТЕКСТ (TEXT)

В разделе предлагается текстовый вариант описания – NSL-T. Язык соответствует естественному.

Термины вводятся на русском и английском, при этом выбраны наиболее употребительные и близкие варианты на этих языках. В ряде случаев термин, возможно, не

самый общепринятый или не идеально соответствующий – это сделано во избежание повторов и конфликтов терминов языка между собой и с терминами специальных областей знания.

4.1 Базис (Basis)

На первом этапе определяются базовые группы терминов – понятия и утверждения.

4.1.1 Понятие (Notion)

Вводятся базовые понятия действительности:

- Субстанция (Substantion):
 - Сущность (Entity)
 - Взаимодействие (Interaction)
- Атрибут (Attribute):
 - Свойство (Property)
 - Локализация (Location)
 - Множество (Set)
 - Порядок (Order)
- Пространство/Область/Размерность (Space/Region/Dimension)
- Время/Период/Последовательность (Time/Period/Sequence)

- Отношение (Attitude)

4.1.2 Утверждение (Statement)

Утверждения используются для описания отношений и являются понятиями познания. Утверждения записываются с помощью выражений.

4.1.2.1 Выражение (Expression)

Выражения включают:

- Операция (Operation)
- Операнд (Operand)
 - Аргумент (Argument)
 - Результат (Result)

4.1.2.2 Вид (View)

Утверждения имеют виды:

- Объявление (Declaration)
- Сочетание (Combination)
- Логика (Logic)
- Определение (Definition)

4.1.2.3 Объявление (Declaration)

Объявления связывают локализации с логикой (см. далее) и включают:

- Существование (Existence):
 - Наличие (Presence)
 - Отсутствие (Absence)
- Сравнение (Comparison)
- Тождество (Identity)
- Отличие (Difference)
- Указание (Indication):
 - Уникальный (Unique)
 - Любой (Any)

4.1.2.4 Сочетание (Combination)

Сочетания являются отношениями между локализациями:

- Пересечение (Intersection)
- Объединение (Union)
- Исключение (Exclusion)
- Вложение (Inclusion)

4.1.2.5 Логика (Logic)

Вводятся базовые логические сущности, которые применяются к рассуждениям и используются в объявлениях и сочетаниях:

- Логическая сущность (Logical Entity):
 - Истина (Truth)
 - Ложь (False)
- Логические отношения (Logical Operation):
 - Или (Or)
 - И (And)

4.1.2.6 Определение (Definition)

Определения используются для этапов описания новых понятий и могут включать:

- Вложение (Inclusion)
- Заключение (Conclusion)

Введённых групп достаточно для построения терминология языка NSL-T.

4.2 Основные объективные понятия (Main objective notions)

Вводятся понятия на основе базовых.

- Мир (World)

Объединений всех сущностей. Эквивалентно утверждению Р.Бартини: мир – это тотальный уникальный экземпляр [20].

- Точка (Point)

Локализация, локализации внутри которой .

- Расположение (Position)

Точка в пространстве.

- Момент (Moment)

Точка во времени.

и т.д.

Далее инкрементно вводятся понятия системного моделирования в разделах:

- Основные объективные понятия (Main objective notions)
- Основные субъективные понятия (Main subjective notions)
- Производные понятия (Derived notions)
- Оценки (Grades)
- Деятельности (Activities)
- Смыслы (Meanings)
- Системы (Systems)
- Программы (Programs)

В том числе даются определения обсуждавшийся ключевых терминов:

- Система (System)

Комплекс, имеющий модель поведения.

- Архитектура (Architecture)

Компоненты объекта и их связи/соединения (как статические, так и динамические) с внешними объектами и между собой (если они локализируются окружением).

По названиям разделов можно заключить, что в язык включен указанный ранее объект «цели» спецификаций TOGAF и ArchiMate. Интересно, что определение «смысла жизни» также удалось дать, и определение подсказывает, что ИИ может при определённом развитии определить смысл своего существования и поставить целью его обеспечение.

Для сравнения дадим сравнительный анализ нескольких терминов языков UML и NSL-T (из [21] –

из-за отсутствия прямого определения в спецификации UML, из [22]– немного адаптировано) в Табл. 1.

Таблица 1: Сравнение ряда терминов UML и NSL

UML		NSL-T	
Наименование	Объяснение	Наименование	Объяснение
Диаграмма вариантов использования (UCD – Use Case Diagram)	A diagram that shows the relationships among actors and use cases within a system [21]. Функциональное назначение системы. Представляет функциональность системы в виде отношений использования внешними сущностями (т.н. actors) для достижения определенных целей [22].	Диаграмма случая использования (Use case Diagram)	Диаграмма алгоритма случая использования.
Диаграмма кооперации (Collaboration Diagram)	A diagram that shows interactions organized around the structure of a model, using either classifiers and associations or instances and links. Unlike a sequence diagram, a collaboration diagram shows the relationships among the instances. Sequence diagrams and collaboration diagrams express similar information, but show it in different ways [21]. Статическое представление взаимодействия. Взаимосвязи структурного характера между классами, не зависящие от времени или реакции системы на внешние события [22].	Диаграмма кооперации (Cooperation Diagram)	Диаграмма связей компонентов сервисов.
Диаграмма состояний (Statechart Diagram или SMD – State Machine Diagram)	A behavior [diagram] that specifies the sequences of states that an object or an interaction goes through during its life in response to events, together with its responses and actions [21]. Процесс изменения состояний системы и подсистем. Представляет поведение сущности в виде переходов между состояниями, провоцируемых событиями [22].	Диаграмма состояний (State Diagram)	Диаграмма алгоритма с состояниями сервиса и сигналами их изменений.

Можно заметить, что предлагаемые NLS определения близки к определениям UML, но лаконичнее, при этом они логически строго выстроены.

V. NSL-F – ФОРМУЛЫ (FORMULA)

В разделе предлагается формульный вариант описания – NSL-F. Предложен язык описаний, позволяющий передать текстовый вариант в формализованном виде.

5.1 Структура описаний

Как правило, описания включают два раздела:

- обозначения;
- утверждения.

Обозначения включают, как правило, две половины (одна или две колонки):

- левая – субстанции;
- правая – атрибуты или операции.

Для описания обозначений применяются два варианта:

- термин и пиктограмма;
- термин в кавычках.

Пиктограммы применяются в первых подразделах (основные объективные и субъективные термины) – применение пиктограмм здесь компактно и наглядно. В дальнейших разделах, связанных в первую очередь с субъективными результатами (оценки, деятельность и т.п.), используются термины в кавычках, т.к. терминов много, и описание пиктограммами теряет очевидность.

Для наглядности поддерживается следующее различие пиктограмм:

- сущности – закрашенные, темные пиктограммы;
- атрибуты или операции – незакрашенные, светлые пиктограммы.

Таблица 2: Пример пиктограмм

Субстанция (Substation)	Пиктограмма	Атрибут (Attribute)
Сущность (Entity)		Свойство (Property)
Взаимодействие (Interaction)		Локализация (Location)
Пространство (Space)		Множество (Set)
Время (Time)		Порядок (Order)

5.2 Утверждение (Statement)

Утверждение, описанное формулой, представляет собой строку обозначений и отношений, при этом оно может содержать вложенные утверждения.

5.2.1 Обозначения (Symbol)

Пример обозначений приведён в Табл. 3.

Таблица 3: Примеры обозначений

Обозначение	Пиктограмма	Обозначение	Пиктограмма
Операция (Operation)		Определение (Definition)	:
Аргумент (Argument)		Операнд (Operand)	\Rightarrow
Уникальный (Unique)		Последствие (Consequence)	\Rightarrow
Любой (Any)		Список (List)	,
Вложение (Inclusion)		Продолжение (Continuation)	—
И (And)		Фраза (Phrase)	;

5.2.2 *Отношение (Attitude)*

Отношение является формулой, которая используется для записи отношений между понятиями и для определения других понятий.

Опуская детализацию языка формул, приведём примеры шаблонов ключевых формул, участвующих в определении нового понятия:

- Отношение-операция:

$$\text{Понятие1} = \text{Операция (Аргумент)}$$

где Аргумент – любой список понятий.

Смысл: Понятие1 локализуется тем, что зафиксировано в применении Операция к Аргумент.

- Вложение-понятие:

$$\text{Понятие1} \subset \text{Операция (Аргумент)}$$

Смысл: Понятие1 (как локализация) вложено в Понятие2 (как локализацию).

- Вложение-операция:
 - Одинарное

$$\text{Операция1} \subset \text{Операция2}$$

Смысл: в этом случае типы аргументов и результатов для Операция1 и Операция2 совпадают.

- Бинарное

$$\text{Операция1} \subset (\text{Понятие1} \Rightarrow \text{Понятие2})$$

Смысл: в этом случае аргумент для Операция – Понятие1, результат для Операция – Понятие2.

- Заключение:
 - Заключение-отношение
Форма как для Отношение-операция.
 - Заключение-вывод:

$$\text{Отношение1} \Rightarrow \text{Отношение2}$$

Смысл: при условии выполнения условий в Отношение1, понятие локализуется тем, что зафиксировано в Отношение2 с участием понятия.

- Определение:

$$\text{Вложение} : \text{Заключение}$$

Смысл: понятие определяется через его Вложение и Заключение.

Пример:

$$\boxed{1} \subset \boxed{2} : \boxed{1} = \diamond(\boxed{2})$$

или, опуская скобки:

$$\boxed{1} \subset \boxed{2} : \boxed{1} = \diamond\boxed{2}$$

Смысл: Локализация1 входит в Локализация2 и определяется применением операции \diamond к Локализация2.

Определение применяется в том числе для операций:

$$\text{Операция1} \subset \text{Операция2} : \text{Заключение}$$

Смысл: определяемое Операция1 входит в Операция2 и локализуется тем, что зафиксировано в Заклчении с участием Операция1.

Пример:



$$\diamond 1 \subset \diamond 2 : \diamond 1 (\boxed{1}) = \diamond 3 \diamond 2 (\boxed{1})$$

В данном случае возможно применение Операция1 к Локализация1 (в правой части отношения) ввиду вложенности Операция1 в Операция2. Аргумент Операция1 совпадает с аргументом Операция2. Результат Операция1 совпадает с результатом Операция3 после Операция2.

5.3. *Основные объективные понятия (Main objective notions)*

На этой основе с помощью языка формул NSL-F последовательно, в соответствии с определениями в NSL-T, вводятся формульные определения с применением пиктограмм, пример которых приведён в 0.

Таблица 4: Обозначения (Symbols)

Понятие	Пиктограмм а	Атрибут/Операци я	Пиктограмм а
Мир (World)		Переход (Transition)	\rightarrow
Точка (Point)	\bullet	Перемещение (Movement)	\Rightarrow
Расположени е (Position)		Эволюция (Evolution)	\mapsto

В этой записи введённые ранее термины записываются следующим образом:

- Мир (World)
 $\blacksquare \subset \text{☀} : \blacksquare = U(\forall \text{☀})$

или (в определении Бартини)

$$\blacksquare \subset \text{☀} : (\forall \text{☀} \subset \blacksquare) \wedge ((\blacksquare \subset \text{☀}) \Rightarrow (\text{☀} = \blacksquare))$$

- Точка (Point)

$$\bullet \subset \square : \square \bullet = \bullet$$

- Расположение (Position)

$$\square = \bullet \oplus$$

- Момент (Moment)

$$\odot = \bullet \odot$$

и т.д.

VI. NSL-G – ГРАФИКА (GRAPHICS)

В языке NSL-G используются следующие графические элементы (образы) – для наглядности они приводятся сразу с терминами, для которых применяются).

Таблица 5: Примеры образов базовых понятий

Субстанция (Substation)	Образ (Image)	Атрибут (Attribute)	Образ (Image)
Сущность (Entity)		Свойство (Property)	
Взаимодействие (Interaction)		Локализация (Location)	
Пространство (Space)		Множество (Set)	
Время (Time)		Порядок (Order)	

Таблица 6: Примеры образов сочетаний

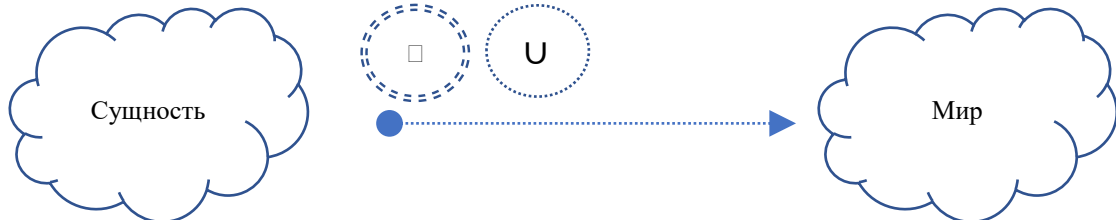
Сочетание	Образ	Сочетание	Образ
Объединение (Union) / Или (Or)		Исключение (Exclusion) / Не (Not)	
Пересечение (Intersection) / И (And)		Вложение (Inclusion) / В (In)	

Таблица 7: Примеры образов связи

Связь	Образ	Связь	Образ
Сочетание (Combination)		Наследование (Inheritance)	

Как делалось ранее, приведём примеры записи первых четырёх определений:

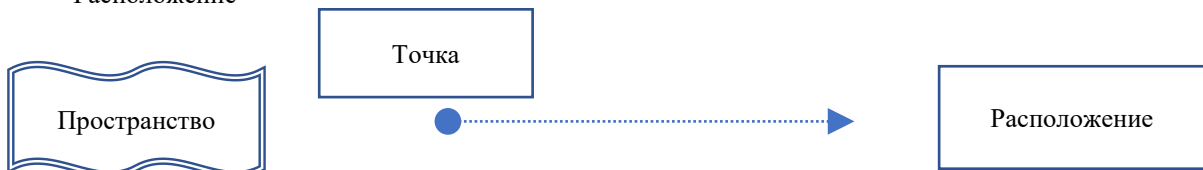
- Мир



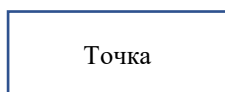
- Точка



- Расположение



- Момент



VII. ЗАКЛЮЧЕНИЕ

Таким образом, реализуемость предложенной концепции описания систем, базирующейся на фундаментальных естественно-научных принципах, подтверждается практически: на языках NSL-T, NSL-F, NSL-G в текстовом, формульном и графическом изложениях удаётся получить основные термины системного описания и моделирования в широком понимании (включая аспекты бизнес-процессов и технической реализации). Предложены применения языка NSL для моделирования систем, построения терминологии и искусственного интеллекта.

Полная спецификация языка планируется к публикации в течение ближайшего времени.

БЛАГОДАРНОСТИ

Автор признателен Филиппову В.Е. за помощь с переводом словаря на английский, Зорину П.А. за замечания по тексту и Костогрызову А.И. за оценку материала и рекомендации по терминологии.

КОНФЛИКТ ИНТЕРЕСОВ

Конфликт интересов: автор заявляет об отсутствии конфликта интересов. Работа выполнена автором самостоятельно, вне интересов каких-либо иных лиц или организаций.

БИБЛИОГРАФИЯ

- [1] OMG Unified Modeling Language (OMG UML) Version 2.5.1 // <https://www.uml.org/>
- [2] Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. Язык UML. Руководство пользователя = The Unified Modeling Language user guide. 2-е изд. // М., СПб.: ДМК Пресс, Питер, 2004. 432 с. ISBN 5-94074-260-2.
- [3] OMG Systems Modeling Language Version 1.5 // <https://www.omgsysml.org/>
- [4] Friedenthal S., Moore A., Steiner R. A Practical Guide to SysML: The Systems Modeling Language. 2nd ed. // Elsevier, 2011. ISBN 978-0-12-385206-9.
- [5] Business Process Model and Notation (BPMN) Version 2.0.2 // <https://www.omg.org/spec/BPMN/2.0.2/PDF>
- [6] И.Г. Фёдоров. Моделирование бизнес-процессов в нотации BPMN2.0 / МЭСИ, 2013. ISBN 978-5-7764-0772-7.
- [7] Welcome to TOGAF. Version 9.1 Enterprise Edition // <https://www.opengroup.org/togaf/>
- [8] ArchiMate® 3.1 Specification. The Open Group Standard // ISBN 1-947754-30-0
- [9] OMG Meta Object Facility (MOF) Core Specification. Version 2.5 // <https://www.omg.org/spec/MOF/2.5>
- [10] OWL Web Ontology Language. Use Cases and Requirements // <https://www.w3.org/TR/2004/REC-webont-req-20040210/>
- [11] ANSYS SCADE Suite. User Manual. Revision SCS-UM-19 - DOC/rev/21560-07 // Esterel Technologies SAS, 2019.
- [12] IEC 61513 Edition 2.0 2011 «Nuclear power plants – Instrumentation and control important to safety – General requirements for systems» // IEC, Geneva, 2011.

- [13] ГОСТ Р МЭК 61513-2020 «Системы контроля и управления, важные для безопасности атомной станции. Общие требования» // Москва, Стандартинформ, 2020.
- [14] IAEA Nuclear Safety and Security Glossary, 2022 (Interim) Edition // IAEA, Vienna, 2022.
- [15] ГОСТ Р 57100-2016 «Системная и программная инженерия. Описание архитектуры» // Москва, Стандартинформ, 2016.
- [16] IEEE Std 1471-2000 «IEEE Recommended Practice for Architectural Description of Software-Intensive Systems» // The Institute of Electrical and Electronics Engineers, Inc., 2020.
- [17] Integrated computer-aided manufacturing (ICAM). Architecture part 11. Volume IV - Function modeling manual (IDEF0) // SofTech, Inc., 1981.
- [18] ГОСТ Р ИСО 9000-2015 (ISO 9000:2015) «Системы менеджмента качества. Основные положения и словарь» // Москва, Стандартинформ, 2015.
- [19] Розенблатт Фрэнк. Принципы нейродинамики. Перцептроны и теория механизмов мозга // М.: Мир, 1965.
- [20] Роберт Орос Ди Бартини. Некоторые соотношения между физическими константами // Доклады Академии наук СССР. Физика. 1965. Том 163, N. 4. С.861-864.
- [21] Meta Object Facility (MOF) Specification. Version 1.4 // <https://www.omg.org/spec/MOF/1.4/About-MOF>
- [22] Леоненков А. В. Самоучитель UML. — 2-е изд., перераб. и доп. // СПб.: БХВ-Петербург, 2004. ISBN 5-94157-342-1

Natural Principles and Methods for Describing Systems

Aleksandr Zorin

Abstract – The paper presents a new system language, which is characterized by a well-founded design based on fundamental concepts.

A number of system description and modeling languages are analyzed, in particular, UML, SysML, BPMN, TOGAF, etc., currently used in designing systems for various purposes. A number of problems with these languages are identified, including focus on special areas of application, a large number of elements and diagrams with no justification for them, poor reflection of time aspects, etc.

A critical analysis of terminological difficulties in large projects, mainly arising from contradictions in the glossaries of the standards used, is carried out.

The aim of the work is defined as the creation of a system language being based on fundamental concepts (and therefore with a minimum of internal contradictions), having, along with a graphical representation (as a method of visual description), also a textual one (for the preparation and machine processing of documents), as well as formalized rules for linking textual constructions together (as an instrument for justifying and constructing terms).

A method for describing systems is proposed based on the basic notions of natural sciences: essence, interaction, space, time, attribute, attitude. The system description language NSL (Natural System Language) is presented in text (NSL-T), formula (NSL-F) and graphical presentation (NSL-G). A consistent incremental creation of the glossary is demonstrated in the sections Basic Objective Notions, Basic Subjective Notions and further to the sections Systems and Programs.

The results of the proposed approach are demonstrated by comparing the names and descriptions of the most commonly used UML diagrams. Possible applications of the developed method and NSL language for modeling systems and constructing terminology, as well for improving an artificial intelligence are presented.

Keywords – systems analysis, modeling, documenting, language, artificial intelligence.

REFERENCES

- [1] OMG Unified Modeling Language (OMG UML) Version 2.5.1 // <https://www.uml.org/>
- [2] Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modeling Language user guide. 2nd ed. // Moscow, St. Petersburg: DMK Press, Piter, 2004. 432 p. ISBN 5-94074-260-2.
- [3] OMG Systems Modeling Language Version 1.5 // <https://www.omg-sysml.org/>
- [4] Friedenthal S., Moore A., Steiner R. A Practical Guide to SysML: The Systems Modeling Language. 2nd ed. // Elsevier, 2011. ISBN 978-0-12-385206-9.
- [5] Business Process Model and Notation (BPMN) Version 2.0.2 // <https://www.omg.org/spec/BPMN/2.0.2/PDF>
- [6] I.G. Fedorov. Modeling Business Processes in BPMN2.0 Notation / MESI, 2013. ISBN 978-5-7764-0772-7.
- [7] Welcome to TOGAF. Version 9.1 Enterprise Edition // <https://www.opengroup.org/togaf/>
- [8] ArchiMate® 3.1 Specification. The Open Group Standard // ISBN 1-947754-30-0
- [9] OMG Meta Object Facility (MOF) Core Specification. Version 2.5 // <https://www.omg.org/spec/MOF/2.5>
- [10] OWL Web Ontology Language. Use Cases and Requirements // <https://www.w3.org/TR/2004/REC-webont-req-20040210/>
- [11] ANSYS SCADE Suite. User Manual. Revision SCS-UM-19 - DOC/rev/21560-07 // Esterel Technologies SAS, 2019.
- [12] IEC 61513 Edition 2.0 2011 «Nuclear power plants – Instrumentation and control important to safety – General requirements for systems» // IEC, Geneva, 2011.
- [13] GOST R IEC 61513-2020 "Instrumentation and control systems important to safety of nuclear power plants. General Requirements" // Moscow, Standartinform, 2020.
- [14] IAEA Nuclear Safety and Security Glossary, 2022 (Interim) Edition // IAEA, Vienna, 2022.
- [15] GOST R 57100-2016 "Systems and Software Engineering. Description of Architecture" // Moscow, Standartinform, 2016.
- [16] IEEE Std 1471-2000 "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems" // The Institute of Electrical and Electronics Engineers, Inc., 2020.
- [17] Integrated computer-aided manufacturing (ICAM). Architecture part 11. Volume IV - Function modeling manual (IDEF0) // SofTech, Inc., 1981.
- [18] GOST R ISO 9000-2015 (ISO 9000:2015) "Quality management systems. Fundamentals and vocabulary" // Moscow, Standartinform, 2015.
- [19] Rosenblatt Frank. Principles of neurodynamics. Perceptrons and the theory of brain mechanisms // Moscow: Mir, 1965.
- [20] Robert Oros Di Bartini. Some relationships between physical constants // Reports of the USSR Academy of Sciences. Physics. 1965. Vol. 163, N. 4. P.861-864.
- [21] Meta Object Facility (MOF) Specification. Version 1.4 // <https://www.omg.org/spec/MOF/1.4/About-MOF>
- [22] Leonenkov A. V. Self-study Guide UML. 2nd edition, revised and enlarged. // St. Petersburg: BHV-Petersburg, 2004.