

# Семантические предметно-ориентированные ЯЗЫКИ

Vitaly S. Gumirov  
Founder and CEO of Eyeline.mobi  
Group  
Novosibirsk, Russian Federation  
[vit@eyeline.mobi](mailto:vitaly@eyeline.mobi)

Peter Y. Matyukov  
Co-founder KIRIK.io and VP  
Innovation Eyeline.mobi Group  
Novosibirsk, Russian Federation  
[peter@kirik.io](mailto:peter@kirik.io)

Dmitry E. Palchunov  
Laboratory of Computability Theory  
and Applied Logic  
Sobolev Institute of Mathematics,  
Novosibirsk State University  
Novosibirsk, Russian Federation  
[palch@math.nsc.ru](mailto:palch@math.nsc.ru)

**Abstract**—A rationale for the introduction of semantic domain specific languages (sDSL) is discussed from the point of view of problems faced by IT industry. Goals and technological approaches for the development of sDSL are described. As well as mathematical model and foundation for such a concept is presented.

**Keywords**— *semantic modelling, artificial intelligence, machine learning, semantic smart contracts, semantic model,  $\Delta_0$ -formula, fragment of atomic diagram*

## I. ВВЕДЕНИЕ

Традиционный подход к разработке ПО основанный на применении алгоритмических языков программирования состоит из следующих основных этапов:

1. Спецификация требований (т.н. тех. задание или ТЗ)
2. Разработка архитектуры
3. Кодирование алгоритмов
4. Тестирование

Эти этапы не обязательно должны следовать один за другим во времени, но могут выполняться в общем случае итеративно. Основная проблема, которая при этом возникает и которую можно в общем виде сформулировать следующим образом:

**Разрушение смыслов (семантики) при преобразовании спецификаций в код.**

## II. ПОСТАНОВКА ПРОБЛЕМЫ

Что это означает на практике? На практике это означает появления целого комплекса проблем:

1. Повышение стоимости поддержки/владения с течением времени

Для исправления ошибок в системе или внесения изменений требуется найти тот кусок исходного кода, который соответствует требованиям по смыслу (семантике), в то же время код не содержит явной привязки к исходным спецификациям, и эта информация часто содержится только в голове кодировщика, писавшего соответствующий кусок кода системы.

2. Высокая стоимость разработки и внедрения

Для управления требованиями, качеством и архитектурой разработчикам приходится применять специальные процедуры для управления привязкой требований к участкам кода, и процедуры управления изменениями и требованиями, которые бы позволяли проносить связь от семантики (требований) через архитектуру, дизайн, к коду. Все эти процедуры не только удорожают процесс, но и требуют отдельных процедур контроля за исполнением и контроля качества.

3. Снижение качества за счет несоответствия ТЗ и систем

Невозможность прямой проверки соответствия кода исходным требованиям (семантики с точки зрения предметной области) специалистами в предметной области (например, заказчиком системы) приводит к снижению качества систем при наличии несоответствия.

4. Повышение стоимости обеспечения качества

В свою очередь необходимость обеспечения качества и в первую очередь соответствия кода систем исходным спецификациям требует внедрения специальных процессов, которые очевидным образом удорожают процесс разработки.

5. Снижение эффективности систем

Со временем в системах накапливаются свойства, которые были разработаны, но документация по ним утеряна или стала неактуальна. Эти недокументированные свойства приводят к снижению эффективности как самих систем, так и к снижению эффективности их использования.

6. Снижение безопасности с течением времени.

Такие недокументированные свойства могут приводить к непредсказуемым сценариям использования систем и в частности к рискам с точки зрения безопасности.

С другой стороны, цифровая трансформация, свидетелями которой мы в настоящее время являемся, по сути предполагает все большую степень автоматизации различных сторон нашей жизни. С точки зрения рынка труда, спрос на программистов не уменьшается, а наоборот растет<sup>1</sup>. С другой стороны, ограниченность этого ресурса ведет с одной стороны к

<sup>1</sup> According to the [U.S. Bureau of Labor Statistics](https://www.bls.gov/news.release/archives/wage01.htm), software developer jobs are expected to grow 17% from 2014 till

2024. <https://www.forbes.com/sites/quora/2017/01/20/will-the-demand-for-developers-continue-to-increase/#3dccc636033ee>

росту средних зарплат в этой области и с другой стороны к снижению среднего профессионального уровня разработчиков.

Это также приводит к перекосам в различных индустриях, когда спрос на ИТ специалистов в некоторых регионах просто не может быть удовлетворен в силу глобальной мобильности рабочей силы<sup>2</sup>.

### III. СЕМАНТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И СЕМАНТИЧЕСКИЕ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫЕ ЯЗЫКИ СПЕЦИФИКАЦИЙ

Ответом на вышеозначенные проблемы и вызовы является все более широкое внедрение и применение семантического моделирования и в частности применение семантических предметно-ориентированных языков спецификаций (*semantic domain specific languages*, далее для краткости далее *sDSL*).

Концепция семантического моделирования (семантического программирования) была предложена и разработана в 1980-х годах С.С. Гончаровым, Ю.Л. Ершовым и Д.И. Свириденко [1,19].

В основе концепции семантического моделирования (программирования) лежит идея *исполнимых спецификаций*, когда специалист в той или иной предметной области может задать спецификации на семантическом языке спецификаций, и эти спецификации будут автоматически исполнены и/или преобразованы в работающий код. В семантическом моделировании используется  $\Delta_0$ -подмножество формул логики предикатов первого порядка

**Определение.** Под *семантическими предметно-ориентированными языками спецификаций* мы будем понимать такие предметно-ориентированные языки, которые могут быть транслированы в язык  $\Delta_0$ -формул.

Под нужды той или иной предметной области могут быть разработаны соответствующие семантические предметно-ориентированные языки спецификаций как некое синтаксическое представление объектно-ориентированного расширения  $\Delta_0$ -формул. Возможность создания таких объектно-ориентированных расширений см. в [2, 18].

В качестве примеров успешного применения такого подхода можно привести следующие системы:

1. Eyeline SDP для отраслей телекоммуникаций и финтех, которая с 2008 года применяется в компаниях МТС, Билайн, Т -mobil, SMART Philippines, YellowPages для управления мобильными и финтех услугами

2. LibrettoLabs Ontobox, применяемая для отрасли розничной торговли

Другие перспективные области применения семантических предметно-ориентированных языков:

- Автоматизация бизнес процессов между различными экономическими субъектами (семантические смарт-контракты)<sup>3</sup>
- Управление интеллектуальной собственностью
- Автоматизация юридических услуг
- FinTech, банки, страхование
- Автоматизация управления проектами
- Управление инвестициями портфелями
- Здравоохранение
- Телекоммуникации
- Робототехника
- Беспилотные технологии
- RegTech

### IV. ЦЕЛИ РАЗРАБОТКИ СЕМАНТИЧЕСКИХ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ЯЗЫКОВ

При разработке и внедрении *sDSL* для конкретной предметной области преследуются следующие цели:

1. Снижение затрат на обучение специалистов предметной области
2. Снижение стоимости внедрения и владения системами созданными на основе *sDSL*
3. Предсказуемость спецификаций на основе *sDSL*
4. Возможность локализации
5. Возможность автоматизации управления качеством систем на основе *sDSL*

### V. КОНЦЕПЦИЯ СЕМАНТИЧЕСКИХ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ЯЗЫКОВ

Разработка и применение семантических предметно ориентированных языков происходит на основе **семантической модели предметной области**, которая будет подробно описана в следующем параграфе.

Язык  $\Delta^\circ$ , основанный на логическом языке  $\Delta_0$  - формул, может исполняться на Семантической Машине, а *sDSL* нужен для того, чтобы специалист в предметной области мог создавать спецификации на понятном ему языке.

Ключевым здесь является требование **автоматической** трансляции в язык  $\Delta^\circ$ , без участия человека, так как это исключает возможность ошибок и разночтений при интерпретации смыслов, а также значительно снижает трудозатраты.

В настоящее время нет устоявшегося определения, что такое предметно ориентированный язык (DSL). Мы будем трактовать это понятие в самом широком

<sup>2</sup> Например, эксперты в области энергетики сетуют, что цифровизация приводит просто невозможности нанять ИТ специалистов на электростанции в удаленных регионах, в силу того, что адекватный уровень оплаты для этого сектора и региона просто

не соответствует среднему уровню оплаты подобного рода ИТ специалистов по стране вообще.

<sup>3</sup> См например платформу семантических смарт-контрактов KIRIK.io [20]

смысле. В частности семантический предметно-ориентированный язык (sDSL) -- это такой способ представления знаний о предметной области, который облегчает специалисту в предметной области создание исполнимых семантических спецификаций. В частности, sDSL может содержать графику, текст и даже голосовые/звуковые элементы.

```
SUPPLY AGREEMENT ID KIRIK_111
PLACE China
EFFECTIVE DATE 01st day of August 2018

BUYER
  Company Name Bees Against Honey Inc.
  Company State California
  Company Address White House, Washington, DC, USA

SUPPLIER
  Company Name Horns and Hooves Inc.
  Company State Alabama
  Company Address White House, Washington, DC, USA

DEFINITIONS
  var Product = "Horns and hooves 18253-72 "Raw materials from horns and hooves""
  var Supply Address = "White House, Washington, DC, USA"

SUBJECT
  var agreesQuantity = 345
  var productCost = 123.45

  def isRiskOnBuyer() means
    BASIC_SUPPLY_AGREEMENT.Product is in Buyer's warehouse()
  end def

  def isRiskOnSupplier() means
    check all
      not isRiskOnBuyer()
    end check
  end def
```

Диаграмма 1: Пример sDSL для типовых юридических контрактов.

```
contract EXCHANGE def
  use SYSTEM
  use Escrow_BTC
  use Escrow_ETH

  rules
    var btcUser = ""
    var ethUser = ""
    if Escrow_BTC.isHoldBTC(btcUser, ethUser) and Escrow_ETH.isHoldETH(ethUser, btcUser) then
      Exchange(ethUser, btcUser)
    end

  end

  def Exchange(ethUser : string, btcUser : string) means
    Escrow_ETH.releaseETH(btcUser) and Escrow_BTC.releaseBTC(ethUser)
  end def

  def StartExchangeETH(toUser : string, eth : numeric, timeout : numeric) means
    check all
      btcUser := toUser
      ethUser := SYSTEM.USER
      Escrow_ETH.holdETH(toUser, eth, timeout)
    end check
  end def

  def StartExchangeBTC(toUser : string, btc : numeric, timeout : numeric) means
    check all
      ethUser := toUser
      btcUser := SYSTEM.USER
      Escrow_BTC.holdBTC(toUser, btc, timeout)
    end check
  end def

end EXCHANGE
```

Диаграмма 2: Пример спецификации на языке  $\Delta^\circ$

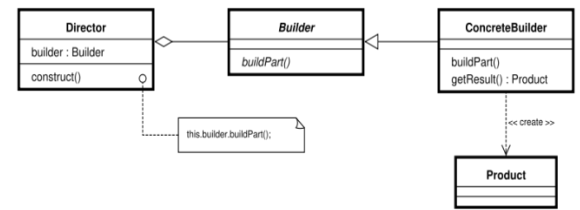


Диаграмма 3: Пример спецификации в графическом виде на языке UML

Отдельно стоит упомянуть возможность использования NLP (Natural Language Processing) и AI (Artificial Intelligence) при создании спецификаций на DSL.

Например, специалист может диктовать голосом, а AI создавать sDSL спецификацию в режиме диалога, уточняя у человека пропущенные параметры.

Или AI может анализировать тексты юридических контрактов на естественном языке и строить автоматически sDSL спецификацию.

VI. СЕМАНТИЧЕСКАЯ МОДЕЛЬ

Семантическая модель состоит из онтологии предметной области и содержательного описания данной предметной области. Формальное представление онтологии и теории предметной области производится на языке конечных фрагментов атомарных диаграмм или на его расширении – языке  $\Delta_0$ -формул.

Для разработки семантического предметно-ориентированного языка мы используем четырехуровневую модель представления знаний, реализованную в виде семантической модели предметной области [3-6]. Семантическая модель содержит онтологию и ещё три уровня представления знаний о предметной области: общие (теоретические) знания, эмпирические знания и оценочные (вероятностные) знания [4, 6] (рис. 1).

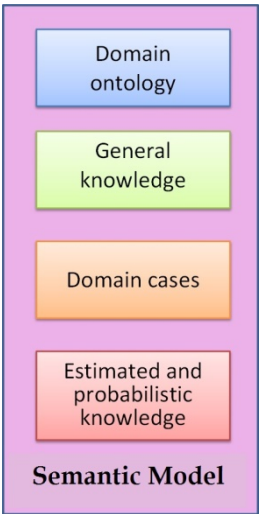


Рис. 1. Семантическая модель

1. Онтология предметной области содержит определения, спецификацию смысла ключевых понятий, на языке которых описывается данная предметная область. В случае необходимости целесообразно рассматривать не одну онтологию, а некоторую иерархию онтологий. Каждая из онтологий в таком случае будет иметь свой набор ключевых понятий (т.е., свою сигнатуру) и множество определений, задающих смысл ключевых понятий.

В рамках развиваемого нами теоретико-модельного подхода к формализации знаний мы используем следующее определение онтологии:

**Определение** [7, 8]. Онтологией предметной области  $SD$  мы называем пару  $O = \langle SA, \sigma \rangle$ , где  $\sigma$  – множество ключевых понятий предметной области, а  $SA$  – множество аналитических предложений, описывающих смысл этих ключевых понятий.

Аналитические предложения – это предложения, истинность которых определяется только смыслом входящих в них понятий (сигнатурных символов) [7-9].

При помощи онтологий мы можем пополнять фрагмент атомарной диаграммы модели, которую мы строим. Для этого, например, можно использовать представленные в онтологии отношения между понятиями: синонимию, «общее-частное» и т.д.

С другой стороны, поскольку понятия, содержащиеся в тексте, имеют определённый смысл, описываемый онтологией, на модели, фрагменты атомарной диаграммы которой мы строим, должна выполняться онтология рассматриваемой предметной области. То есть, на модели должно быть истинно множество аналитических предложений  $SA$ .

Содержательное описание предметной области, представленное в её семантической модели, содержит ещё три уровня представления знаний:

2. Универсальные, общие утверждения – законы и постулаты рассматриваемой предметной области. Это теоретические знания о предметной области: общих принципах, законах и закономерностях и пр., которые на данный момент времени считаются полностью достоверными. Общие знания являются известной нам частью теории предметной области. Эти знания являются синтетическими, в то время как в онтологии представлены аналитические знания. Синтетические знания, в отличие от аналитических, не вытекают из смысла употребляемых понятий, их истинность зависит от состояния реального мира. Поэтому эти знания могут изменяться (то есть, истинные утверждения могут становиться ложными, а ложные утверждения, наоборот, становиться истинными), при этом смысл употребляемых понятий может оставаться полностью неизменным. Эти утверждения носят общий, универсальный характер. В отличие от следующего уровня представления – эмпирических знаний, они описывают не отдельную ситуацию, а предметную область в целом, считаются истинными для всех ситуаций.

3. Эмпирические знания содержат описания конкретных ситуаций, прецедентов предметной области. Формально описание каждого прецедента

представляется в виде фрагмента атомарной диаграммы алгебраической системы. Знания о прецедентах предметной области формально представляются в виде прецедентной модели, которая является булевозначной моделью [10-12].

4. Вероятностные, нечёткие и оценочные знания. Эти знания либо берутся из внешних источников, либо порождаются на основе знаний, представленных на первых трёх уровнях семантической модели. В результате фазификации булевозначной прецедентной модели [11, 12], указанной в предыдущем пункте, получается нечёткая модель, в которой значениями истинности формул являются числа из интервала  $[0; 1]$ . На основе анализа имеющихся в семантической модели эмпирических данных и сопоставления их с общими знаниями и с онтологическими знаниями, представленными в семантической модели, порождаются вероятностные и оценочные знания. Формально вероятностные и оценочные знания представляются в виде множества предложений  $\Gamma \subseteq S(\sigma_A)$  и отображения  $\mu: \Gamma \rightarrow [0, 1]$ , ставящего в соответствие каждому предложению  $\varphi \in \Gamma$  его нечёткое значение истинности  $\mu(\varphi) \in [0, 1]$  [11, 12].

Далее нам потребуются некоторые определения и обозначения. Сведения по теории моделей можно найти в [13, 14].

Мы рассматриваем модели

$\mathfrak{A} = \langle A; \sigma \rangle = \langle A; P_1, \dots, P_n, c_1, \dots, c_l \rangle$  – сигнатуры  $\sigma = \langle P_1, \dots, P_n, c_1, \dots, c_l \rangle$ , где  $A$  – основное множество (универсум) модели  $\mathfrak{A}$ ,  $P_1, \dots, P_n$  – символы предикатов, а  $c_1, \dots, c_l$  – символы констант. Через  $S(\sigma)$  обозначим множество предложений, т.е. формул без свободных переменных, сигнатуры  $\sigma$ .  $\mathfrak{A} \models \varphi$  означает, что на модели  $\mathfrak{A}$  истинно предложение  $\varphi$ .

Для модели  $\mathfrak{A}$  сигнатуры  $\sigma$  обозначим  $\sigma_A = \sigma \cup \{c_a \mid a \in A\}$ ; при этом считаем, что  $c_a \notin \sigma$  при  $a \in A$ . Через  $\mathfrak{A}_A$  обозначим модель сигнатуры  $\sigma_A$ , обеднение которой до сигнатуры  $\sigma$  совпадает с моделью  $\mathfrak{A}$  и при этом выполнено  $c_a^{\mathfrak{A}_A} = a$  для  $a \in A$ .

Обозначим:

$D(\mathfrak{A}) = \{\varphi \in S(\sigma_A) \mid \mathfrak{A}_A \models \varphi \text{ и предложение } \varphi \text{ – бескванторное}\}$  – элементарная диаграмма модели  $\mathfrak{A}$ .

Напомним, что мы рассматриваем сигнатуру  $\sigma$ , состоящую только из символов предикатов и констант. Предложение  $\varphi$  назовем атомарным, если  $\varphi = P(c_1, \dots, c_n)$ , где  $P, c_1, \dots, c_n \in \sigma_A$ .

Пусть  $\mathfrak{A}$  – модель сигнатуры  $\sigma$ . Обычно атомарной диаграммой модели  $\mathfrak{A}$  называется множество предложений

$\{\varphi \in S(\sigma_A) \mid \mathfrak{A}_A \models \varphi \text{ и предложение } \varphi \text{ – атомарное}\}.$

Поскольку мы рассматриваем фрагменты (то есть, подмножества) атомарной диаграммы модели, целесообразно использовать несколько иное определение.

**Определение.** Атомарной диаграммой модели  $\mathfrak{A}$  мы называем множество предложений

$AD(\mathfrak{A}) = \{\varphi \in S(\sigma_A) \mid \mathfrak{A}_A \models \varphi \text{ и либо предложение } \varphi \text{ – атомарное, либо } \varphi = \neg\psi \text{ и предложение } \psi \text{ – атомарное}\}.$

Заметим, что атомарная диаграмма  $AD(\mathfrak{U})$  определяет модель  $\mathfrak{U}$  с точностью до изоморфизма. Это, в частности, означает, что атомарная диаграмма  $AD(\mathfrak{U})$  однозначно определяет элементарную диаграмму  $D(\mathfrak{U})$ ; при этом выполнено  $AD(\mathfrak{U}) \vdash D(\mathfrak{U})$ .

Поэтому задачу построения модели можно свести к задаче построения её атомарной диаграммы.

В рамках теоретико-модельного подхода к представлению знаний о предметной области можно выделить два способа формализации: семантический и синтаксический. Знания о предметной области, в том числе и онтологические знания, можно представлять как при помощи алгебраической системы, так и в виде множества предложений логики предикатов первого порядка.

Семантические методы основаны на использовании алгебраических систем. Синтаксические методы основаны на использовании теорий, в частности элементарных теорий алгебраических систем. И тот и другой подход обладает своими преимуществами и недостатками.

Мы используем синтез семантического и синтаксического подходов [15]. Для формализации знаний, извлекаемых из текстов естественного языка и представляемых в семантической модели, мы используем конечные фрагменты (подмножества) атомарных диаграмм моделей.

Атомарная диаграмма модели – это множество истинных на ней атомарных предложений – предикатов от констант, и отрицаний атомарных предложений; при этом сигнатура модели пополняется дополнительными константами – именами каждого элемента. Как было отмечено выше, своей атомарной диаграммой модель определяется с точностью до изоморфизма. Поэтому, с одной стороны, совокупность конечных фрагментов атомарных диаграмм модели полностью задаёт эту модель, то есть полностью определяет семантику. С другой стороны, как сама атомарная диаграмма модели, так и её фрагменты являются некоторыми множествами предложений, то есть описание знаний при помощи фрагментов атомарных диаграмм лежит в русле синтаксического подхода. Таким образом, представление знаний при помощи фрагментов атомарных диаграмм моделей является интеграцией семантического и синтаксического подходов.

Кроме того, использование в качестве базового объекта конечных фрагментов атомарных диаграмм позволяет нам решить следующие две проблемы.

Во-первых, мы избавляемся от бесконечности. Действительно, с одной стороны, рассматриваемые алгебраические системы, как правило, бесконечны. С другой стороны, любая теория в логике предикатов первого порядка также бесконечна, так как бесконечным является уже множество тождественно истинных предложений. Таким образом, конечными множествами предложений – конечными фрагментами атомарных диаграмм – мы можем аппроксимировать как бесконечные модели, так и бесконечные теории.

Во-вторых, использование фрагментов атомарных диаграмм моделей позволяет нам решить ещё более важную и сложную проблему – необходимость работать с алгебраическими системами разной сигнатуры. В классической теории моделей, при изучении вопросов аксиоматизации, истинности предложений на классах моделей, рассматриваются только классы алгебраических систем, имеющих одинаковую сигнатуру. С другой стороны, в практических приложениях возникает необходимость работать с классами моделей разной сигнатуры [16]. Более того, часто приходится иметь дело с ситуацией, когда сигнатура рассматриваемой модели заранее не известна – в том случае, если модель ещё только строится.

Такая ситуация возникает, например, когда мы извлекаем знания из текстов естественного языка: сигнатура алгебраической системы, которую мы строим, может быть в любой момент времени расширена. В частности, если мы объединяем знания, извлечённые из разных документов, мы должны объединять их сигнатуры, то есть множества понятий, представленных в этих документах. Эту проблему мы решаем при помощи атомарных диаграмм моделей: мы объединяем несколько конечных фрагментов диаграмм в один.

Для формального логического представления знаний о предметной области в виде семантической модели мы используем язык  $\Delta_0$ -формул. Для целей данной работы мы будем использовать следующее определение  $\Delta_0$ -формул.

#### Определение.

1. Если  $P$  – предикатный символ,  $P \in \sigma$ , а  $t_1, \dots, t_n$  – символы констант сигнатуры  $\sigma$  или переменных, то  $P(t_1, \dots, t_n)$  –  $\Delta_0$ -формула.
2. Если  $\varphi$  и  $\psi$  –  $\Delta_0$ -формулы, то  $\varphi \& \psi$ ,  $\varphi \vee \psi$ ,  $\varphi \rightarrow \psi$ , и  $\neg \varphi$  –  $\Delta_0$ -формулы.
3. Если  $\varphi$  –  $\Delta_0$ -формула,  $x$  – переменная и  $l$  – конечный список, то  $(\forall x \in l)\varphi$  и  $(\exists x \in l)\varphi$  –  $\Delta_0$ -формулы.
4. Других  $\Delta_0$ -формул нет.

Несложно заметить, что любая атомарная формула является  $\Delta_0$ -формулой. Более того, конъюнкция всех формул, входящих в произвольный конечный фрагмент атомарной диаграммы, также является  $\Delta_0$ -формулой. Поэтому можно считать, что язык  $\Delta_0$ -формул является расширением языка конечных фрагментов атомарных диаграмм. В частности, определённая выше семантическая модель предметной области может быть полностью описана на языке  $\Delta_0$ -формул.

Важно также отметить, что язык  $\Delta_0$ -формул, как и язык конечных фрагментов атомарных диаграмм, является алгоритмически разрешимым. Одной из наших задач является создание эффективных разрешающих алгоритмов для языка  $\Delta_0$ -формул и для языка конечных фрагментов атомарных диаграмм.

Опишем процесс построения фрагмента атомарной диаграммы модели. Атомарные предложения порождаются на основе автоматизированной обработки предложений текстов естественного языка,

описывающих данную предметную область. Строится фрагмент атомарной диаграммы модели – совокупность атомарных предложений. При этом возможна интеграция нескольких фрагментов атомарных диаграмм, соответствующих разным текстам естественного языка. Используется программная система LogicText [17], которая осуществляет порождение фрагмента атомарной диаграммы модели по тексту естественного языка.

## VII. ПРОЦЕСС РАЗРАБОТКИ sDSL

При разработке sDSL предлагается руководствоваться следующим правилом: в отличие от языков программирования, языковые конструкции sDSL должны отвечать на вопрос ЧТО, а не КАК. Например, sDSL может содержать команду “Перевести деньги со счета А на счет Б”, но не должен включать подробные алгоритмические команды, реализующие это действия. То есть реализация не должна выноситься на уровень спецификации.

Предлагаемый процесс разработки sDSL:

1. Строится объектно-ориентированная модель предметной области
2. Выделяется подмножество объектно-ориентированной модели (данные, предикаты, методы классов), которое будет выноситься на уровень спецификации (и иметь представление в нотации/синтаксисе sDSL)
3. Реализуются базовые предикаты/базовые функции, которые будут использоваться в sDSL. Это могут быть системы работы с базами данных, системы на основе машинного обучения и нейронных сетей и/или другие системы, реализованные с помощью обычных языков программирования.
4. Выбирается тип нотации sDSL: текстовый, графический, или система на основе NLP (Natural Language Processing).
5. Реализуется автоматическая трансформация из sDSL в язык  $\Delta^0$ .

## REFERENCES

- [1] Гончаров С.С., Ершов Ю.Л., Свириденко Д.И. Методологические аспекты семантического программирования // Научное знание: логика, понятия, структура. - Новосибирск, Наука 1987. - с. 154-184.
- [2] В.Ш. Гумиров, "Объектно-ориентированный вариант языка  $\Sigma$ -спецификаций," [Online]. Available: <https://goo.gl/UjvUrp>
- [3] D.Palchunov, G.Yakhyaeva, O.Yasinskaya. Software system for the diagnosis of the spine diseases using case-based reasoning // The Siberian Scientific Medical Journal. 2016. Vol. 36, No. 1, p. 97-104.

- [4] C.Naydanov, D.Palchunov, P.Sazonova. Development of automated methods for the critical condition risk prevention, based on the analysis of the knowledge obtained from patient medical records. In: Proceedings of the International Conference on Biomedical Engineering and Computational Technologies (SIBIRCON / SibMedInfo — 2015), 2015, Novosibirsk, p. 33-38.
- [5] Palchunov D., Yakhyaeva G., Dolgusheva E. Conceptual Methods for Identifying Needs of Mobile Network Subscribers // Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications, Moscow, Russia, July 18-22, 2016, p. 147-160.
- [6] Naidanov C. A., Palchunov D. E., Sazonova P. A. Model-theoretic methods of integration of knowledge extracted from medical documents // Vestnik NSU Series: Information Technologies. - 2015. - Volume 13, Issue No 3. - P. 29–41. - ISSN 1818-7900. (in Russian).
- [7] Palchunov D.E. Modeling of reasoning and formalization of reflection I: Model theoretical formalization of ontology and reflection. Filosofiya nauki, No. 4 (31), 2006, p. 86-114 (in Russian).
- [8] Palchunov D.E. The solution of the problem of information retrieval based on ontologies // Bisnes-informatika, No. 1, 2008, — p. 3–13 (in Russian).
- [9] Carnap, R. Meaning and Necessity. A Study in Semantics and Modal Logic. —Chicago, 1956. — 220 p.
- [10] Pal'chunov D.E., Yakhyaeva G.E. Interval Fuzzy Algebraic Systems // Mathematical Logic in Asia. Proceedings of the 9th Asian Logic Conference'05. World Scientific Publishers, 2006. Pp. 191-202.
- [11] Pal'chunov D.E., Yakhyaeva G.E. Fuzzy algebraic systems // Vestnik NGU. Seriya: Matematika, mexanika, informatika, vol. 10, no. 3, 2010, p. 75-92 (in Russian).
- [12] Pal'chunov D.E., Yakhyaeva G.E. Fuzzy logics and fuzzy model theory // Algebra and Logic, vol. 54, no. 1, 2015, p. 74-80.
- [13] Chang C.C., Keisler H.J. Model theory. — Moscow: Mir, 1977. — 615 p. (in Russian).
- [14] Ershov Yu. L., Palyutin E. A. Mathematical Logic. — Moscow: Nauka, 1979. — 317 p. (in Russian).
- [15] Makhasoeva O.G., Palchunov D.E. Semi-automatic methods of a construction of the atomic diagrams from natural language texts // Vestnik NSU: Information Technologies - 2014. Vol. 12, No. 2, p.64–73. - ISSN 1818-7900 (in Russian)
- [16] Kleschev A.S., Artemeva I.L. Mathematical models of subject domain ontologies. Parts 1-3 // Nauchno-tekhnicheskaya informatsiya. Seriya 2. 2001, No. 2, p. 20-27, No. 3, p.19-29, No. 4, p. 10-15. (In Russian).
- [17] Makhasoeva O.G., Palchunov D.E. Program system for the construction of the atomic diagram of a model from natural language texts. (in Russian). Certificate of the State Registration of the computer program. No. 2014619198, registered 10.09.2014.
- [18] Гумиров В.Ш. Объектно-ориентированный вариант языка  $\Sigma$ -спецификаций // Теория вычислений и языки спецификаций, Новосибирск, 1995 - Вып. 152: Вычислительные системы, стр. 3-19.
- [19] S.S. Goncharov, D.I. Sviridenko  $\Sigma$ -Programming Amer. Math. Soc. Transl. (2) Vol. 142, 1989
- [20] KIRIK.io whitepaper, 2018 // [Online]. Available: <http://bit.ly/2Gjn2wE>

We suggest tat you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.