



Серия «Математика»
2017. Т. 22. С. 71–89

Онлайн-доступ к журналу:
<http://mathizv.isu.ru>

ИЗВЕСТИЯ
Иркутского
государственного
университета

УДК 004.82, 510.62

MSC 68T30

DOI <https://doi.org/10.26516/1997-7670.2017.22.71>

Построение локально-простых моделей: методология и практика*

И. А. Казаков, И. А. Кустова, Е. Н. Лазебникова, А. В. Манцивода
*Иркутский государственный университет,
Институт математики им. С. Л. Соболева*

Аннотация. Одним из самых заметных трендов, связанных с разворачивающейся на наших глазах четвертой промышленной революцией, является существенное усиление роли семантических методов, ориентированных на управление смыслами и знаниями. Это касается и средств искусственного интеллекта, и добычи смыслов из огромных потоков больших данных, и роботизации, и появления умных городов и домов, других вещей. Появились и умные контракты, хотя над «интеллектом» контрактов еще следует серьезно поработать. Данные тренды должны с неизбежностью вести к усилению роли логических методов, работающих с семантикой, и существенно расширить сферу их применения на практике. Однако здесь есть ряд проблем, которые тормозят процесс.

Нами разрабатывается подход к использованию логического моделирования в ряде важных предметных областей, который позволяет обойти возникающие препятствия. Подход основан на концепции локально-простых моделей и прежде всего ориентирован на решение задач в сфере управления предприятиями, организациями, органами власти. Важнейшей особенностью локально-простых моделей является их способность заменить программные системы. Замена программирования на моделирование дает колоссальные преимущества: резко снижаются издержки на разработку и поддержку систем, моделирование в отличие от программирования сохраняет явную семантику моделей, оставляя возможности работы для средств искусственного интеллекта и роботов. Кроме того, модели намного более понятны людям, чем программы.

В настоящей работе рассматривается реализация концепции локального-простого моделирования на основе документных моделей, разработанных нами ранее. Показывается, что локально-простое моделирование реализуется через документные модели с конечными подмодельными покрытиями. Во второй части работы описывается пример использования документных моделей для решения управленческой задачи реальной сложности.

* Исследование выполнено за счет гранта Российского научного фонда (проект №17-11-01176).

Ключевые слова: локально-простая модель, документная модель, семантическое программирование, бизнес-процесс.

1. Введение

В данной работе продолжается рассмотрение концепции локально-простых моделей (ЛПМ), введенных в [2]. Обсуждается внутренняя структура ЛПМ, а также уточняются методологические аспекты их использования. Показывается, каким образом абстрактная архитектура локально-простых моделей находит свое конкретное воплощение в рамках документных моделей, а также формулируются соответствующие математические определения.

Одной из важнейших особенностей локально-простых моделей является их применимость к решению задач реальной сложности. Логическая технология, основанная на ЛПМ, не только позволяет решать серьезные задачи, но и имеет значительные конкурентные преимущества по сравнению со стандартными подходами.

На сегодняшний день главное направление приложений ЛПМ — решение управленческих задач на предприятиях, в учреждениях, в бизнесе. Однако предлагаемая нами методология подходит для самого широкого спектра предметных областей. Например, мы планируем использовать ЛПМ для проекта «Цифровой Байкал», который посвящен построению единой модели мультидисциплинарных данных об озере с дальнейшим автоматическим поиском междисциплинарных корреляций, основанном на логико-вероятностных методах машинного обучения.

Если говорить о решении задач управления на предприятиях, то здесь наш подход конкурирует с такими системами как Oracle ERP, SAP, Microsoft Dynamics AX (Ахapta) и российские 1С, которые сегодня доминируют в данной области.

Все эти продукты представляют собой программные системы, управляющие бизнес-процессами. Но программирование имеет один крайне неприятный изъян, который мы называем лоботомией смыслов [5]. Единая цельная семантика взаимодействующих бизнес-процессов в результате программирования расчленяется и растворяется в базах данных и программном коде. В результате мы теряем прямой контроль над бизнес-процессами. Без доступа к семантике невозможно использовать средства искусственного интеллекта и управленческих роботов — им просто не с чем работать.

Использование локально-простых моделей позволяет исключить программирование, заменив его на семантическое моделирование. Правильно построенные семантические модели могут напрямую действо-

вать как информационные системы, делая ненужной работу программистов. При этом, в отличие от программного кода, семантические модели сохраняют смыслы бизнес-процессов, и открыты для инструментов искусственного интеллекта и роботов.

Свойство локально-простых моделей быть непосредственно готовыми к действию называется *исполняемостью* (это свойство они наследуют из концепции семантического программирования [13; 14], которое соединяет моделирование с вычислимостью). Исполняемость означает, что построенные ЛПМ сразу начинают функционировать как прикладные системы, действующие в моделируемой предметной области. Такое качество ЛПМ позволяет получить прорывные конкурентные преимущества относительно систем, основанных на программировании:

- отказ от программирования многократно сокращает время разработки, принципиально снижает издержки на поддержку и модернизацию,
- моделирование задает явную семантику предметной области, необходимую для средств искусственного интеллекта и роботов (в программном коде семантика растворяется),
- моделированием (в отличие от программирования) может заниматься широкий круг работников — менеджеров, консультантов-аналитиков, юристов,
- моделирование позволяет строить *семантические* смарт-контракты, которые в отличие от популярных сегодня программных смарт-контрактов [8], могут интерпретироваться как обычные документы и пониматься не только программистами.

Для решения прикладных задач реальной сложности нами разработана bSystem — веб-платформа управления документными моделями промышленных масштабов и решения задач локально-простого моделирования. В настоящее время bSystem используется для реализации прикладных управленческих проектов на предприятиях, решающих задачи бюджетирования, управления товародвижением, производством и персоналом, логистики, взаимодействия с клиентами, управления паями в с/х кооперативе и других. Во второй части работы нами описывается один из таких проектов — построение технологии расчета заработной платы крупной торговой компании на основе моделирования системы мотиваций персонала предприятия.

2. Архитектура локально-простой модели

В этом параграфе рассмотрим более подробно внутреннюю структуру локально-простых моделей (ЛПМ). Как следует из определения ЛПМ, она собирается из достаточно простых компонент, но сама модель

может быть сколь угодно сложной. Каждая такая компонента представляет собой подмодель, описывающая относительно замкнутую и четко очерченную область, например компетенцию сотрудника, отдельную компоненту управления предприятием и т. д.

Локальная простота компоненты неформально означает, что её функционал не требует для своего понимания чрезвычайных интеллектуальных усилий (которые, скажем, необходимы при доказательстве сложной математической теоремы) и может, например, соответствовать некоторой рутинной работе.

Чтобы такие компоненты могли совместно действовать, они должны быть каким-то образом объединены друг с другом при сохранении их обособленности. Поэтому общая архитектура ЛПМ многокомпонентна и неоднородна. Рассмотрим основные виды компонент ЛПМ на концептуальном уровне. Этими компонентами являются локаль, отчет, компетенция и оракул.

2.1. Локаль

Локаль — это подмодель ЛПМ, описывающая относительно замкнутую и обособленную компоненту глобальной модели. В случае управленческих моделей это может быть локаль товародвижения, локаль управления персоналом, локаль взаимодействия с клиентами и т. д. Локаль является основным блоком построения ЛПМ. Вычислительная сложность локали должна быть невысокой, например не превышать уровня сложности конечных автоматов. Именно невысокий уровень сложности локалей должен обеспечивать локальную простоту глобальной модели.

2.2. ОТЧЕТ

ЛПМ строится через интеграцию локалей. Но прямое «смешение» двух локалей — плохая идея (хотя и естественная математически), поскольку, она, во-первых, отягощает сложность получившегося объединения и, во-вторых, совершенно не соответствует практике (никто не смешивает мозг бухгалтера с мозгом товароведа). Локали не должны иметь полный доступ друг к другу — их взаимодействие должно быть строго регламентировано в рамках процессов, происходящих в моделях. Этот принцип реализуется через понятие *отчёта*. Локали взаимодействуют через общие для них подмодели, которые называются отчетами. Неформально в подмодели отчета одна из локалей генерирует логические описания («отчеты»), которые используются в жизненном цикле другой локали. Отчеты можно охарактеризовать как «логические интерфейсы» между моделями. Отчёты обеспечивают также уровни безопасности, делая доступной только ту информацию, которая явно оговорена.

2.3. КОМПЕТЕНЦИЯ

В локально-простой модели функционируют акторы, действия которых формируют жизненный цикл модели. В роли таких акторов могут выступать, например, люди, действующие на своих рабочих местах, или роботы, выполняющие определенные функции. *Компетенция* — подмодель ЛППМ, которая описывает то, какой видит актор локально-простую модель со своей позиции, а также какие действия он может производить в рамках жизненного цикла ЛППМ. Компетенции позволяют описывать и интерпретировать одну и ту же информацию разными способами. Например, данные о приобретаемой компьютерной технике будут совершенно по-разному отражаться в компетенции технического директора (подробная информация по спецификациям оборудования) и финансиста (обобщенная информация по технике, но подробное описание финансовой составляющей закупок).

Компетенция позволяет локализовать изменения в модели. Например, если исполнение некоторой рутинной работы поручается вместо человека роботу, то компетентностная модель данной работы не изменится, и поэтому модель в целом не заметит изменений. Компетенции используются и при подключении интернета вещей (IoT). Для подключаемой вещи (например, автоматической метеостанции) формируется компетенция, в рамках которой данная вещь будет взаимодействовать с локально-простой моделью.

2.4. ОРАКУЛ

Акторы действуют в модели, опираясь на данные из самой модели. Но они могут также использовать информацию из внешних источников, либо запускать внешние процедуры. В этом случае актор выступает для ЛППМ в роли *оракула*. Именно оракулы обеспечивают взаимодействие модели с внешним миром. Реализации оракулов могут быть самыми разнообразными, например, ввод данных с клавиатуры, результат деятельности интеллектуальных роботов, интернет вещей (данные автономной метеостанции, работа кассового аппарата) и еще множество вариантов.

Ключевым качеством оракула является то, что по механизмам взаимодействия оракул ничем не отличается от локалей. Как и у локалей, его коммуникация заключается в получении и отправлении отчетов другим компонентам модели. Поскольку локали взаимодействуют с другими компонентами через отчеты, они неспособны в общем случае отличить, с кем они коммуницируют в настоящий момент — с другой локалью или оракулом.

Вторая важная функция оракулов заключается в том, что они играют роль «модельной заглушки» в процессе развития ЛППМ. Если какая-

то из локалей ещё не готова, то вместо неё вставляется оракул, который взаимодействует с окружающими компонентами аналогично отсутствующей модели. Таким временным оракулом может служить ручной ввод данных или импорт из других информационных систем. Главное, чтобы он снабжал окружающие модели такими же отчетами, какие они будут получать от модели, когда она будет добавлена. Технологически это очень важно: как только недостающая модель будет готова, она бесповоротно волеется в единое семантическое пространство, другие компоненты даже не заметят разницы.

Последнее свойство задаёт принципиально новый процесс внедрения семантических моделей по сравнению с программными системами. Как известно, встраивание новой программной системы в существующее информационное пространство часто связано с тяжелой интеграционной работой. С другой стороны, наша практика показала, что семантические модели распространяются «тихой сапой» – как спрут, постепенно захватывая всё новые и новые сегменты бизнес-процессов организации.

2.5. ПРИМЕР

В процессе моделирования деятельности комбината питания, который кормит десятки тысяч детей в день в 350 школах и детских садах, среди прочих нами были построены следующие локалы:

- Локаль технолога меню, которая логически описывает знания, связанные с питанием – калорийность, раскладка блюд на определенный период времени, учёт возраста, описание обобщенных данных, направляемых в контролирующие органы и т. д.
- Локаль контингентов, описывающая структуру учебных заведений, количество учащихся, их расслоение по возрасту, полу и другим особенностям, которые нужно принимать во внимание при организации питания.
- Локаль контрактной деятельности, формализующая виды контрактов и взаимодействие предприятия со своими контрагентами (поставщиками).

Синергия совместного использования этих локальных моделей позволяет автоматизировать процесс бюджетного планирования и логистики. Умножение объемов базовых ингредиентов блюд (локаль технолога) на количество учащихся (локаль контингентов), распределенных по времени, задает динамику потребления ингредиентов. Это позволяет нам, во-первых, строить расходную часть бюджета (с учётом стоимостей ингредиентов, описываемой в локали контрактной деятельности), а во-вторых, роботизировать логистику предприятия, обеспечивающую поддержку необходимых остатков ингредиентов на складах учебных заведений (автоматический заказ поставок у контрагентов).

3. Локальная простота документных моделей

Как уже было показано выше, локально-простые модели имеют неоднородную структуру и скомпонованы из подмоделей, каждая из которых играет определенную роль. В данном параграфе мы определяем, каким образом свойство локальной простоты формализуется на документных моделях [2].

Прежде всего нам необходимо расширить определение документа из [2], добавив в документы *функциональные поля*. Мы будем работать с одноместными функциями двух типов:

- *Регулярные функции* – полиномиально-вычислимые функции над локально-простой моделью.
- *Внешние функции*, являющиеся для модели черным ящиком и реализующие влияние внешнего мира.

Будем обозначать множества этих функций *Reg* и *Ext*, соответственно. Полиномиальная вычислимость регулярных функций – ограничение, обеспечивающее локальную простоту модели на функциональном уровне. Это ограничение может варьироваться (например, можно потребовать линейную вычислимость в случаях, особо чувствительных к эффективности).

Добавим в документы два типа полей, работающих с функциями. Пусть ϕ — документ.

- *Функциональное поле* — поле документа ϕ , содержащее определение регулярной функции $fun \in Reg$ (внешние функции из *Ext* в модели не определяются). Значением поля является сама определяемая функция fun .
- *Вычисляемое поле* — поле документа ϕ , содержащее определение регулярной функции $fun \in Reg$, либо ссылку на внешнюю функцию $ext \in Ext$. Значением поля является значение $fun(\phi)$ в случае регулярной функции, и значение $ext(\phi)$ в случае функции внешней.

Чтобы формально ввести в определение документа поля, работающие с функциями, предпримем следующие шаги. Во-первых, расширим определение описания поля документа из [2]. Поле документа описывается там как тройка $\mathbf{d} = \langle d, \mathbf{g}, \mathbf{c} \rangle$, где $d \in \mathbb{I}_D$ – имя поля, $\mathbf{g} \in \mathbb{B} \cup \mathbb{I}_F$ – его тип, и $\mathbf{c} \in \mathbb{C}$ – его кардинальность. Расширим множество допустимых типов значений поля функциональными типами вида

$$\mathbf{f} \rightarrow \langle \mathbf{g}, \mathbf{c} \rangle$$

где $\mathbf{f} \in \mathbb{I}_F$ — форма, определяющая тип аргумента функции (это всегда документ), $\mathbf{g} \in \mathbb{B} \cup \mathbb{I}_F$ — тип результата функции, а $\mathbf{c} \in \mathbb{C}$ — его кардинальность. Данные типы описывают одноместные функции, получающие в качестве аргумента документы формы \mathbf{f} , и возвращающие

значения типа \mathbf{g} и кардинальности \mathbf{c} . Определением функционального поля назовём выражение

$$\mathbf{d} = \langle d, \mathbf{f} \rightarrow \langle \mathbf{g}, \mathbf{c} \rangle \rangle$$

где $d \in \mathbb{D}$ – имя поля. Значениями такого поля будут функции типа $\mathbf{f} \rightarrow \langle \mathbf{g}, \mathbf{c} \rangle$. Мы не уточняем способ задания функций, лишь фиксируя ограничения, которые накладываются на функции из-за локальной простоты модели (полиномиальная вычислимость).

Таким образом, если ϕ – документ формы \mathbf{f} , а fun – функция типа $\mathbf{f} \rightarrow \langle \mathbf{g}, \mathbf{c} \rangle$, то результат применения функции $fun(\phi)$ должен иметь тип \mathbf{g} и кардинальность \mathbf{c} .

Дадим теперь формальное определение *вычисляемого поля*. Как и функциональные поля, вычисляемые поля определяются функциями, но их значением является значение функции в контексте документа, в которое входит поле. Приведем пример. Пусть документ ϕ описывает прямоугольник, и в нем имеются поля *длина* и *ширина*, задающие размеры геометрической фигуры. В такой документ можно добавить вычисляемое поле *площадь*, которое базируется на регулярной функции

$$площадь = \lambda doc. \{ doc.длина \times doc.ширина \}$$

Значением поля *площадь* будет результат применения функции к самому документу ϕ

$$\lambda doc. \{ doc.длина \times doc.ширина \}(\phi).$$

Общий вид определения вычисляемого поля выглядит так:

$$\mathbf{d} = \langle d, f \rangle,$$

где $f \in Reg \cup Ext$. Отметим, что функция вычисляемого поля задается на уровне определения формы документа. Это означает, что для всех документов данной формы в поле будет действовать одна и та же функция.

В отличие от регулярных функций, которые определяются внутри документной модели, внешние функции являются для модели черным ящиком. Интуитивно, они снабжают модель эмпирической информацией из предметной области, в которую погружена модель. Документы, содержащие поля с внешними функциями, будем называть *документами-оракулами*. Примером внешней функции может выступить курс доллара к рублю на определенный момент времени или данные о текущей температуре овощей в умном холодильнике.

Теперь мы готовы формализовать понятие *локальной простоты для документных моделей*. Это определение будет опираться на графовую природу документных моделей.

Пусть

$$\nu : \mathbb{N} \rightarrow \mathbb{D}$$

нумерация документов [2]. Документы ϕ_1, \dots, ϕ_m назовем *непосредственно доступными* из документа ϕ , если id_1, \dots, id_m — множество всех ссылок на документы, входящих в качестве значений всех полей документа ϕ , и для каждого $i = \overline{1, m}$: $\nu(id_i) = \phi_i$. Обозначим через $Ac(\phi)$ множество всех документов, непосредственно доступных из ϕ .

Определение 1. *Множество документов*

$$\mathbb{O} = \{\phi_1, \dots, \phi_m\}$$

назовем замкнутым относительно нумерации ν , если для каждого документа $\phi_i \in \mathbb{O}$ выполняется $Ac(\phi_i) \subseteq \mathbb{O}$.

Будем говорить, что множество документов $\mathbb{O} = \{\phi_1, \dots, \phi_m\}$ *индуцирует модель* относительно нумерации ν , если существует модель нумерации ν , содержащая данное множество документов и только их.

Предложение 1. *Множество документов $\mathbb{O} = \{\phi_1, \dots, \phi_m\}$ индуцирует модель тогда и только тогда, когда \mathbb{O} замкнуто относительно нумерации.*

Доказательство. Доказательство «туда» очевидно, поскольку модель должна быть замкнутой относительно нумерации. В обратную сторону, ясно, что такой моделью будет модель

$$\mathcal{M} = \langle \{\phi_1, \dots, \phi_m\}, \nu \rangle$$

сигнатуры \mathcal{M} , содержащей ровно те формы документов, которые входят в множество \mathbb{O} . Легко проверить, что на \mathcal{M} выполняются все свойства документной модели [2]. Замкнутость множества документов гарантирует, что все документы, на которые ссылается документ модели, также принадлежат модели. \square

Определение 2. Пусть $\mathcal{M}_1 = \langle \mathbb{O}_1, \nu \rangle$ и $\mathcal{M}_2 = \langle \mathbb{O}_2, \nu \rangle$ — две модели с общей нумерацией ν . \mathcal{M}_1 назовем *подмоделью модели \mathcal{M}_2* (обозначается $\mathcal{M}_1 \subseteq \mathcal{M}_2$), если $\mathbb{O}_1 \subseteq \mathbb{O}_2$.

Предложение 2. Пусть

$$\mathcal{M} = \langle \mathbb{O}, \nu \rangle$$

— модель и $\mathcal{M}_1 = \langle \mathbb{O}_1, \nu \rangle$ и $\mathcal{M}_2 = \langle \mathbb{O}_2, \nu \rangle$ — её подмодели ($\mathcal{M}_1 \subseteq \mathcal{M}$ и $\mathcal{M}_2 \subseteq \mathcal{M}$). Тогда пересечение $\mathbb{O}_1 \cap \mathbb{O}_2$ и объединение $\mathbb{O}_1 \cup \mathbb{O}_2$ множеств документов \mathcal{M}_1 и \mathcal{M}_2 индуцируют подмодели модели \mathcal{M} .

Доказательство. Очевидно, что документы из пересечения могут непосредственно ссылаться только на документы, также принадлежащие пересечению (в противном случае по крайней мере одна из подмоделей не будет замкнута относительно нумерации). Поскольку пересечение определяет множество документов, замкнутое относительно нумерации, то в соответствии с предложением 1 оно индуцирует модель. Аналогично рассматриваем для объединения. \square

Обозначим модель, индуцируемую пересечением множеств документов моделей M_1 и M_2 через $M_{1 \cap 2}$. Такую модель будем называть *пересечением моделей* M_1 и M_2 . Аналогично, через $M_{1 \cup 2}$ обозначим модель, индуцируемую объединением множеств документов, которую будем называть *объединением моделей*.

Определение 3 (Отчёт). Пусть M_1 и M_2 — две модели, а $M_{1 \cap 2}$ — их пересечение. Пересечение $M_{1 \cap 2}$ назовем отчетом, если для любого документа такого, что $\phi \in O_1 \cup O_2$ и $\phi \notin O_1 \cap O_2$, все непосредственно достижимые из ϕ документы принадлежат той же модели, что и сам документ ϕ .

Другими словами, все документы, которые взаимодействуют с обеими моделями одновременно, должны быть сконцентрированы в отчете.

Множество M_1, \dots, M_n возможно пересекающихся подмоделей документной модели $M = \langle O, \nu \rangle$ назовём *покрытием* M , если $O = \bigcup_{i=1}^n O_i$.

Определение 4. Пусть M — некоторая модель, а

$$M_1, \dots, M_n$$

её покрытие. Тогда покрытие M_1, \dots, M_n назовем *локально-простым*, если все непустые попарные пересечения подмоделей из M являются отчетами.

Подмодели M_1, \dots, M_n локально-простого покрытия будем называть *локалями*.

Определение 5. Подмодель, содержащую документы-оракулы назовем *оракулом*.

Компетенция в документных моделях определяется как частный случай оракула. Таким образом, мы определили на документной модели с локально-простым покрытием все необходимые компоненты ЛПМ — локаль, отчет, оракул и компетенцию. Легко видеть, что формальные определения этих компонент в документных моделях соответствуют их концептуальным характеристикам в предыдущем параграфе. Это

позволяет нам использовать документные модели вместе с локально-простыми покрытиями в качестве конкретного представления концептуального понятия ЛПМ. Документные модели послужили основой для создания bSystem – платформы для локально-простого моделирования.

4. bSystem

На базе документного моделирования нами была построена технология решения управленческих задач на предприятиях, в учреждениях и органах власти. Эта технология была реализована как веб-платформа управления документными моделями. Реализация bSystem была осуществлена на веб-фреймворке [3; 4], базирующемся на языке Libretto [7; 18].

bSystem имеет многослойную архитектуру, в которую входят:

- Libretto Web Framework,
- графовая база данных хранения документных моделей,
- интерфейс управления документными моделями,
- конструктор формирования и редактирования документных моделей,
- модуль управления транзакциями и журналами,
- конструктор веб-интерфейсов и сценариев работы с документными моделями.

Веб-платформа обеспечивает облачные механизмы управления документными моделями в удаленном режиме. В настоящее время нами ведется разработка новых методов хранения документных моделей и компиляции языка определения регулярных функций в документах. Хотя и текущая версия bSystem умеет эффективно решать задачи промышленного уровня в реальном времени, мы надеемся, что нововведения позволят существенно повысить эффективность системы как по времени исполнения, так и по памяти. Сегодня bSystem умеет управляться с десятками миллионов документов. Мы планируем вывести её на работу в реальном времени с документными моделями, содержащими десятки и сотни миллиардов документов.

5. Моделирование мотиваций персонала

На базе bSystem сегодня решен и решается ряд прикладных управленческих задач для разных видов предприятий. В этом параграфе мы продемонстрируем применение документного моделирования к решению задачи расчета заработной платы на примере крупной торговой компании. Предприятие является крупнейшим региональным ритейлером, владеющим 150 супермаркетами и дискаунтерами, в которых

работают около 5000 человек. Менеджментом компании была разработана собственная уникальная система мотивации работы сотрудников, состоящая из десятков различных видов стимулирующих вознаграждений – как индивидуальных, когда результат зависит непосредственно от работы сотрудника, так и коллективных, когда результат зависит от работы всего коллектива.

Система мотиваций, в принципе, содержит всю информацию, необходимую для расчета заработной платы на рабочих местах. Но для использования этой информации нужно учесть сложные логические связи между мотивациями, должностями и позициями сотрудников, видами торговли и вспомогательной деятельности, общей структурой предприятия. Идея использования документных моделей состояла в том, чтобы, отказавшись от программирования, смоделировать все эти логические связи в виде набора локалей и оракулов таким образом, чтобы процесс вычисления заработной платы формировался автоматически как операционная семантика этой модели. В течение двух месяцев такая локально-простая модель была построена. Она включала

- локаль заработной платы,
- локаль моделирования мотиваций,
- локаль моделирования рабочих мест и структуры предприятия,
- набор оракулов для тех компонент, которые пока еще не смоделированы в виде локалей (оракул товародвижения, оракул инвентаризации, и еще несколько десятков).

Заметим, что в дальнейшем планируется моделирование в виде локалей тех компонент, которые сегодня реализованы в виде оракулов. Это будет означать постепенный «захват» документной моделью управленческой инфраструктуры компании в целом.

Локально-простая модель системы мотиваций и расчета зарплаты была внедрена в компании, и использовалась для контроля расходов по зарплате в 2017 году, а также для формирования бюджетного плана компании по заработной плате на 2018 год.

Для построения ЛПМ системы мотиваций и расчета заработной платы был использован конструктор моделей и транзакций bSystem. Данный конструктор позволяет при помощи графического интерфейса создавать, редактировать и просматривать формы документов, используемых для описания бизнес-процессов, а также описывать транзакции, которые должны автоматически производиться при достижении заданных условий.

При помощи конструктора можно создавать и дополнять модели вне зависимости от их сложности и имеющихся особенностей. При этом bSystem автоматически генерирует стандартные интерфейсы наполнения моделей данными. Созданная при помощи конструктора модель является исполняемой и не требует дополнительного программирования. Используя конструктор, мы имеем возможность не только собирать

модель требуемого бизнес-процесса, но и расширять ее, соединять с другими собранными моделями, в результате чего мы получим не только сколь угодно детализированные локалы, но и бесповное их соединение в глобальную ЛПМ.

Конструктор позволил нам постепенно «вырастить» модель расчета заработной платы, начиная с задания основных форм документов, таких как штатное расписание, график работы и зарплатная ведомость в их стандартном виде. По мере понимания процесса расчета зарплаты на предприятии, модель дополнялась и видоизменялась. Наряду со стандартными вещами появлялись детали, свойственные только данному предприятию. Достаточно было задать в конструкторе новое поле, добавить новую форму или внести изменения в транзакции, и измененную модель можно было дальше наполнять данными и тестировать, что ускорило процесс ее построения и отладки. После завершения этого процесса, модель была передана в практическую эксплуатацию.

В таблице приведена статистическая информация по расчету заработной платы торговой компании на весь 2018 год.

Параметр	Значение
Всего рабочих мест	3 576
Операционных подразделений	243
Видов вознаграждений	63
Систем корректировки (усилений)	10
Форм документов	70 (з/п – 60)
Штатных расписаний	100
Ведомостей сотрудников (всего)	81 552
Графиков инвентаризации (оракул)	87
Порожденных документов	1 159 113

Зарплата рассчитывалась для 3576 рабочих мест. Рассчитывались магазины — от линейного персонала до директоров магазинов, центральный офис не учитывался. Операционные подразделения — это, в первую очередь, отделы магазинов («Гастроном», «Молоко» и т. д.). На самом деле было смоделировано 94 вознаграждения, но при расчете заработной платы были «включены» только 63 из них. Усиления — это особые режимы расчета, например для работы в новогоднюю ночь. Сигнатура документной модели состоит из 70 форм, из них 60 форм непосредственно связаны с вычислением заработной платы. Количество документов, задействованных в расчете зарплаты, — 1 159 113.

В текущей версии bSystem время расчета зарплаты для всех магазинов на двенадцать месяцев 2018 года составляет примерно 40 минут.

Мы надеемся, что с введением компилятора регулярных функций время будет снижено на порядок – до 4–5 минут, что будет значительно превосходить по производительности существующие сегодня системы расчета. Но дело не только в эффективности расчетов. Опыт внедрения и эксплуатации документной модели показал, что

- Модель умеет «объяснять» проведенный расчет, а также помогает находить ошибки при расчетах, сделанных другими способами.
- С помощью моделирования удалось формально описать очень сложную систему мотиваций, работающую в компании. Существующие коммерческие системы расчета заработной платы базируются на программировании массовых стандартных подходов, поэтому неспособны поддерживать уникальные механизмы, задействованные в компании. В результате те процессы, которые осуществляются bSystem в автоматическом режиме, ранее делались сотрудниками компании вручную.
- Сотрудники, ответственные за расчет зарплаты, непосредственно понимают документы ЛПМ и могут самостоятельно работать с ними (в отличие от программных систем, требующих посредничества программистов). Опросы сотрудников показали, что им bSystem представляется как «логический мега-эксель», в котором вместо табличек и ячеек можно работать с произвольными документами.
- Моделирование оказалось исключительно эффективным с точки зрения реализации и снижения издержек: разработку такой сложной модели осуществили два человека «с нуля под ключ» в течение пары месяцев, причем построили не «стандартизированную» систему расчета, а уникальную, соответствующую конкретной компании.
- Моделирование позволяет резко сократить потребность в ручном труде в рамках исполнения рутинных управленческих операций.
- Модели (локалы), разработанные для одной компании, могут быть легко перенесены и адаптированы для другой компании.

Все эти особенности управленческих технологий, базирующихся на документном моделировании, являются их сильнейшими конкурентными преимуществами по сравнению с системами, доминирующими сегодня на рынке.

6. Заключение

Нами планируется продолжение исследований в рамках парадигмы локально-простого моделирования, в первую очередь, в области документных моделей. Будут исследоваться механизмы сочетания денотационной и операционной семантик документных моделей, жизненный цикл документных моделей, разворачивающийся во времени. Кроме

того, будет осуществляться интеграция документных моделей с логико-вероятностным методом машинного обучения [1; 17], для которого ЛПМ будут играть роль онтологий. Также будут исследоваться возможности сочетания документных моделей с блокчейном [20; 21] и построения на этой основе семантических смарт-контрактов (в отличие от программных смарт-контрактов, наиболее популярных сегодня [8], но критикуемых как излишне низкоуровневые для традиционных бизнесов [19]).

В рамках прикладных исследований особое внимание будет уделяться двум направлениям. Во-первых, совместно с бизнес-партнерами будут решаться конкретные задачи управления бизнес-процессами. Основной целью данных усилий является доводка документного моделирования как технологии решения масштабируемых прикладных задач в области управления, а также конверсия технологии в коммерческий продукт.

Во-вторых, будут исследоваться возможности использования локально-простого моделирования в качестве новой основы для «семантизации» интернета. Идея использования логики для повышения управляемости и автоматизации интернета принадлежит создателю веба Тиму Бернерсу-Ли [10]. В рамках этой идеи, сформулированной почти двадцать лет назад, возникло целое направление исследований – Semantic Web [11; 15; 16]. Математической основой семантического веба стали дескриптивные логики [9]. К сожалению, за прошедшие годы семантический веб так и не смог оказать того глобального влияния, на которое рассчитывал Бернерс-Ли, и в конце концов превратился в еще одну узкоспециализированную математическую дисциплину, не играющую никакой роли в междисциплинарном пространстве интернета. Глобальное визионерское целеполагание Тима Бернерса-Ли было просто забыто.

Мы считаем, что стремительная «семантизация» информационных пространств в контексте Четвертой промышленной революции [6] дает нам шанс вернуться к глобальной идее Бернерса-Ли, но уже с использованием существенно более адекватных математических инструментов, чем дескриптивные логики. Наша гипотеза состоит в том, что локально-простые модели являются хорошим кандидатом для решения этой задачи, а концепция семантического программирования [12–14], соединяющая моделирование с вычислимостью, является наиболее подходящей методологической основой для этого. Поэтому мы планируем начать работу в этом направлении.

Список литературы

1. Витяев Е. Е. Семантический вероятностный вывод предсказаний // Изв. Иркут. гос. ун-та. Сер. Математика. – 2017. – Т. 21. – С. 33–50. <https://doi.org/10.26516/1997-7670.2017.21.33>
2. Малых А. А. Документное моделирование / А. А. Малых, А. В. Манцивода // Изв. Иркут. гос. ун-та. Сер. Математика. – 2017. – Т. 21. – С. 89–107. <https://doi.org/10.26516/1997-7670.2017.21.89>
3. Малых А. А. Объектные теории над списочными надстройками / А. А. Малых, А. В. Манцивода // Изв. Иркут. гос. ун-та. Сер. Математика. – 2012. – Т. 4. – С. 27–44.
4. Малых А. А. Система Libretto: разработка веб-ресурсов в единой модели данных и знаний / А. А. Малых, А. В. Манцивода // 6-я Всероссийская конференция по проблемам управления (МКПУ-2013). Геленджик, 30 сент. – 5 окт. 2013 г. – Геленджик, 2013. – С.73-75.
5. Малых А. А. Ontobox: смарт-контракты как документные модели [Электронный ресурс] / А. А. Малых, А. В. Манцивода. – URL: <https://ontobox.io/ppr> (дата обращения: 30.11.2017).
6. Шваб К. Четвертая промышленная революция / К. Шваб. – М.: Эксмо, 2016. – 138 с.
7. Язык Libretto [Электронный ресурс]. – URL: <http://librettolabs.com> (дата обращения: 30.11.2017).
8. Antonopoulos A. Mastering Ethereum: Building Smart Contracts and Dapps / A. Antonopoulos, G. Wood. – O'Reilly Media, 2017. – 220 p.
9. The Description Logic Handbook: Theory, Implementation, and Applications / eds.: Baader F., Calvanese D., McGuinness D. L., Nardi D., Patel-Schneider P. F. – Cambridge University Press, 2003.
10. Berners-Lee T. The Semantic Web / T. Berners-Lee, J. Hendler, O. Lassila // Scientific American. – May, 2001. <https://doi.org/10.1038/scientificamerican0501-34>
11. Semantic Web activity [Electronic resource]. – URL: <https://www.w3.org/standards/semanticweb/>
12. Goncharov S. S. Conditional terms in semantic programming // Siberian Mathematical Journal. – 2017. – Vol. 58, N 5. – P. 794–800. <https://doi.org/10.1134/S0037446617050068>
13. Goncharov S. S. Semantic foundations of programming / S. S. Goncharov, Yu. L. Ershov, D. I. Sviridenko // Lecture Notes in Computer Science. – 1987. – Vol. 278. – P. 116-122. https://doi.org/10.1007/3-540-18740-5_28
14. Goncharov S. S. Semantic programming / S. S. Goncharov, Yu. L. Ershov, D. I. Sviridenko // Information processing, Proc. IFIP 10-th World Comput. Congress. Vol. 10. – Dublin, 1986. – P. 1093–1100.
15. Horrocks I. From SHIQ and RDF to OWL: The making of a Web Ontology Language / I. Horrocks, P. Patel-Schneider, F. Van Harmelen // Journal of Web Semantics. – 2003. – Vol. 1, Issue 1. – P. 7–26. <https://doi.org/10.1016/j.websem.2003.07.001>
16. Horrocks I. Practical reasoning for expressive description logics / I. Horrocks, U. Sattler, U. Tobies // Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99), number 1705 in Lecture Notes in Artificial Intelligence / eds.: H. Ganzinger, D. McAllester, A. Voronkov. – Springer-Verlag, 1999. – P. 161–180. https://doi.org/10.1007/3-540-48242-3_11

17. Kovalerchuk B. Data Mining in finance: Advances in Relational and Hybrid Methods / B. Kovalerchuk, E. Vityaev. – Kluwer Academic Publishers, 2000. – 456 p.
18. Malykh A. A Query Language for Logic Architectures / A. Malykh, A. Mantsivoda // Lecture Notes in Computer Science 5947. – Springer-Verlag, Berlin Heidelberg, 2010. – P. 294–305. https://doi.org/10.1007/978-3-642-11486-1_25
19. Schumpeter J. Not-so-clever contracts // The Economist. The new political divide. – Issue July 31, 2016. <https://www.economist.com/news/business/21702758-time-being-least-human-judgment-still-better-bet-cold-hearted>.
20. Swan M. Blockchain: Blueprint for a New Economy / M. Swan. – O'Reilly Media, 2015. – 127 p.
21. Wattenhofer R. The Science of the Blockchain / R. Wattenhofer. – Inverted Forest Publishing, 2016. – 320 p.

Казаков Илья Анатольевич, кандидат физико-математических наук, научный сотрудник, Иркутский государственный университет, 664003, Россия, г. Иркутск, ул. К. Маркса, 1; Институт математики им. С. Л. Соболева, 630090, Россия, г. Новосибирск, пр. Академика Коптюга, 4,

Кустова Ирина Александровна, научный сотрудник, Иркутский государственный университет, 664003, Россия, г. Иркутск, ул. К. Маркса, 1; Институт математики им. С. Л. Соболева, 630090, Россия, г. Новосибирск, пр. Академика Коптюга, 4,

Лазебникова Елена Николаевна, научный сотрудник, Иркутский государственный университет, 664003, Россия, г. Иркутск, ул. К. Маркса, 1,

Манцивода Андрей Валерьевич, доктор физико-математических наук, профессор, Иркутский государственный университет, 664003, Россия, г. Иркутск, ул. К. Маркса, 1; Институт математики им. С. Л. Соболева, 630090, Россия, г. Новосибирск, пр. Академика Коптюга, 4, (e-mail: andrei@baikal.ru)

**I. A. Kazakov, I. A. Kustova, E. N. Lazebnikova,
A. V. Mantsivoda**

Locally Simple Models Construction: Methodology and Practice

Abstract. One of the most notable trends associated with the Fourth industrial revolution is a significant strengthening of the role played by semantic methods. They are engaged in artificial intelligence means, knowledge mining in huge flows of big data, robotization, and in the internet of things. Smart contracts also can be mentioned here, although the 'intelligence' of smart contracts still needs to be seriously elaborated. These trends should inevitably lead to an increased role of logical methods working with semantics, and significantly expand the scope of their application in practice. However, there are a number of problems that hinder this process.

We are developing an approach, which makes the application of logical modeling efficient in some important areas. The approach is based on the concept of locally

simple models and is primarily focused on solving tasks in the management of enterprises, organizations, governing bodies. The most important feature of locally simple models is their ability to replace software systems. Replacement of programming by modeling gives huge advantages, for instance, it dramatically reduces development and support costs. Modeling, unlike programming, preserves the explicit semantics of models allowing integration with artificial intelligence and robots. In addition, models are much more understandable to general people than programs.

In this paper we propose the implementation of the concept of locally simple modeling on the basis of so-called document models, which has been developed by us earlier. It is shown that locally simple modeling is realized through document models with finite submodel coverages. In the second part of the paper an example of using document models for solving a management problem of real complexity is demonstrated.

Keywords: locally simple modeling, document model, semantic programming, business process

References

1. Vityev E. Semantic Probabilistic Inference of Predictions. *Izv. Irkutsk. Gos. Univ. Ser. Mat.*, 2017, vol. 21. (in Russian). <https://doi.org/10.26516/1997-7670.2017.21.33>
2. Malykh A., Mantsivoda A. Document modeling. *Izv. Irkutsk. Gos. Univ. Ser. Mat.*, 2017, vol. 21, p. 33-55 (in Russian). <https://doi.org/10.26516/1997-7670.2017.21.89>
3. Malykh A., Mantsivoda A. Object Theories over List Superstructures. *Izv. Irkutsk. Gos. Univ. Ser. Mat.*, 2012, vol. 4, pp. 27-44. (in Russian)
4. Malykh A., Mantsivoda A. Sistema Libretto: razrabotka web-resursov v edinoi modeli dannykh i znanii [Libretto System: Web Resources Development Based On a Holistic Data and Knowledge Model]. *Proceedings of The 6th All-Russian Conference on Control Problems*. Gelendzhik, September, 2013, pp. 73-75.
5. Malykh A., Mantsivoda A. Ontobox: Smart Contracts as Document models. URL: <https://ontobox.io/ppe>.
6. Schwab K. The Fourth Industrial Revolution. World Economic Forum, 2015. 120 p.
7. Programming Language Libretto. - URL: <http://librettolabs.com>.
8. Antonopoulos A., Wood G. Mastering Ethereum: Building Smart Contracts and Dapps. O'Reilly Media, 2017. 220 p.
9. Baader F., Calvanese D., McGuinness D.L., Nardi D., Patel-Schneider P.F. (eds.) The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003.
10. Berners-Lee T., Hendler J., Lassila O. The Semantic Web *Scientific American*. May, 2001. <https://doi.org/10.1038/scientificamerican0501-34>
11. Semantic Web activity. URL: <https://www.w3.org/standards/semanticweb/>
12. Goncharov S.S. Conditional terms in semantic programming. *Siberian Mathematical Journal*, vol. 58, no 5, pp. 794-800. <https://doi.org/10.1134/S0037446617050068>
13. Goncharov S.S., Ershov Yu.L., Sviridenko D.I. Semantic foundations of programming. *Lecture Notes in Computer Science*. 1987, vol. 278, pp. 116-122. https://doi.org/10.1007/3-540-18740-5_28
14. Goncharov S.S., Ershov Yu.L., Sviridenko D.I. Semantic programming. *Information processing, Proc. IFIP 10-th World Comput. Congress*. Dublin, 1986, vol. 10, pp. 1093-1100.

15. Horrocks I., Patel-Schneider P., Van Harmelen F. From SHIQ and RDF to OWL: The making of a Web Ontology Language. *Journal of Web Semantics*, 2003, vol. 1, Issue 1, pp. 7-26. <https://doi.org/10.1016/j.websem.2003.07.001>
16. Horrocks I., Sattler U., Tobies U. Practical reasoning for expressive description logics. In: H. Ganzinger, D. McAllester and A. Voronkov (eds.). *Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99)*, number 1705 in *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 1999, pp. 161-180. https://doi.org/10.1007/3-540-48242-3_11
17. Kovalerchuk B., Vityaev E. Data Mining in finance: Advances in Relational and Hybrid Methods. Kluwer Academic Publishers, 2000. 456 p.
18. Malykh A., Mantsivoda A. A Query Language for Logic Architectures. *Lecture Notes in Computer Science 5947*. Springer-Verlag, Berlin Heidelberg, 2010, pp. 294–305. https://doi.org/10.1007/978-3-642-11486-1_25
19. Schumpeter J. Not-so-clever Contracts. *The Economist*. The new political divide. Issue July 31, 2016. URL: <https://www.economist.com/news/business/21702758-time-being-least-human-judgment-still-better-bet-cold-hearted>.
20. Swan M. *Blockchain: Blueprint for a New Economy*. O'Reilly Media, 2015. 127 p.
21. Wattenhofer R. *The Science of the Blockchain*. Inverted Forest Publishing, 2016. 320 p.

Kazakov Ilya Anatolyevich, Candidate of Science (Physics and Mathematics), Researcher, Irkutsk State University, 1, K. Marx, Irkutsk, 664003, Russian Federation; Sobolev Institute of Mathematics, 4, Academic Koptyug avenue, Novosibirsk, 630090, Russian Federation.

Kustova Irina Alexandrovna, Researcher, Irkutsk State University, 1, K. Marx, Irkutsk, 664003, Russian Federation; Sobolev Institute of Mathematics, 4, Academic Koptyug avenue, Novosibirsk, 630090, Russian Federation.

Lazebnikova Elena Nikolaevna, Researcher, Irkutsk State University, 1, K. Marx, Irkutsk, 664003, Russian Federation.

Mantsivoda Andrei Valeryevich, Doctor of Science (Physics and Mathematics), Professor, Irkutsk State University, 1, K. Marx, Irkutsk, 664003, Russian Federation; Sobolev Institute of Mathematics, 4, Academic Koptyug avenue, Novosibirsk, 630090, Russian Federation, (e-mail: andrei@baikal.ru)