

УДК 004.655.2

Сравнение парадигм моделей данных для описания модели мира информационных систем

Шишкин Д. А., аспирант

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационные системы и телекоммуникации»*

Научный руководитель: Девятков В.В., д.т.н, профессор

Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана

deviatkov@bmstu.ru

1. Введение

В настоящее время остро стоит вопрос в создании качественных информационных систем, которые позволяют пользователю быстрее и правильнее принимать решения. Качество информационной системы напрямую зависит от качества данных, которыми оперирует эта система, поэтому одним из важных вопросов для построения информационной системы является выбор парадигмы модели данных предметной области.

По аналогии с термином парадигмы программирования [13] под парадигмой модели данных будем понимать совокупность идей и понятий, определяющих стиль и подход к созданию модели данных.

Остановливаясь на определенной парадигме, мы тем самым выбираем синтаксическую и семантическую части языка описания данных. Синтаксическая часть языка определяет то, *как* описать данные, то есть то, какими структурами будут описаны данные. Семантическая же часть определяет то, *что* описывать, то есть, как будут интерпретироваться структуры памяти в объекты реального мира во время работы системы. Синтаксис и семантика языка связаны друг с другом и могут ограничивать друг друга. Например, определившись с синтаксисом языка, некоторые высказывания могут быть недостижимы из-за ограничений в выразительности языка. С другой стороны, одну и ту же семантику можно выразить различными синтаксическими средствами.

Для создания качественной модели данных необходимо не только выбрать язык описания предметной области, но и всю парадигму модели данных. Таким образом, целью

настоящей статьи является сравнение различных парадигм моделей данных для описания предметной области.

Настоящая статья организована следующим образом. В разделе 2 выбираются основные критерии качества парадигмы модели данных. Раздел 3 посвящен описанию различных парадигм моделей данных и описание на основе критериев качества. В разделе 4 будет приведено итоговое сравнение описанных ранее парадигм. Раздел 5 является заключительным.

2. Критерии качества модели данных

Модель всегда отображает только некоторые аспекты системы, выбираемые исследователем для включения в модель в зависимости от того, насколько эти аспекты существенны с точки зрения решения основной задачи. Выбор цели, таким образом, определяет характер всей последующей работы, связанной с построением и использованием модели, а также полезность и надежность получаемого результата.

Для оценки различных парадигм описания модели мира необходимо определиться с основными критериями качества моделей данных. В качестве основных критериев были выбраны критерии, описанные в [15].

Адекватность

Под адекватностью модели данных понимается максимальное соответствие функций, параметров, характеристик модели объектов реального мира и соответствующих свойств моделируемого объекта. Адекватность оценивается по тем свойствам модели, которые для системы считаются существенными.

Система может считаться пригодной для работы только тогда, когда его модель мира является максимально адекватной. В обратном случае система не позволит принимать правильные решения.

Адекватность может достигаться использованием справочных данных – общепринятых на уровне мира, страны, отрасли и предприятия терминов и определений.

Точность

Под точностью понимается степень совпадения полученных в процессе моделирования результатов с заранее установленными или желаемыми. Здесь важной задачей является оценка потребной точности результатов и имеющейся точности исходных данных, согласование их как между собой, так и с точностью используемой модели.

Количественное значение точности невозможно определить у парадигмы моделирования, так как парадигма описывает только подход и не содержит конкретных величин. Но для парадигмы можно качественно оценить возможности внесения изменений, корректировок и поправок в модель мира пользователем для более точного описания мира, которые напрямую влияют на точность модели предметной области.

Универсальность

Применимость модели мира к планированию ряда однотипных систем в одном или нескольких режимах функционирования. Это позволяет расширить область применимости модели мира для решения большего круга задач.

Критерий универсальности может достигаться следующими способами:

- Поддержка разных предметных областей – возможность использования одной и той же модели мира для построения моделей данных разных предметных областей;
- Расширяемость модели мира - заключается в возможности пополнения справочных данных новыми типами объектов без изменения алгоритмов системы;
- Повторное использование результатов - заключается в возможности полного или частичного использования описаний в других системах;

Целесообразная экономичность

Точность получаемых результатов и общность решения задачи должны увязываться с затратами на моделирование. И удачный выбор модели, как показывает практика, — результат компромисса между отпущенными ресурсами и особенностями используемой модели.

Экономичность создания модели данных может достигаться за счет:

- Использования более распространенных языков для описания модели;
- Более эффективных инструментов создания и проверки модели мира.

Требования универсальности, точности, адекватности с одной стороны и экономичности с другой противоречивы. Это обуславливает существование множества подходов для описания моделей мира, отличающихся теми или иными свойствами.

Другие критерии

Качество модели мира в смысле ее соответствия своему назначению и практической полезности характеризуется также такими показателями как:

- Наглядность, обозримость основных свойств и отношений;

- Управляемость, предполагающая наличие в модели должна хотя бы одного параметра, изменениями которого можно имитировать поведение моделируемой системы в различных условиях;
- Поддержка промышленных стандартов;
- Доступность и технологичность для исследования или воспроизведения;
- Адаптивность, под которой понимается способность модели приспосабливаться к различным входным параметрам и воздействиям окружения;
- Способность к эволюции, т.е., к количественному и качественному развитию.

3. Основные парадигмы моделей данных

В данном разделе будут рассмотрены основные парадигмы описания модели предметной области.

Субстанциональная парадигма

Субстанциональная парадигма строится на основе парадигмы, созданной Аристотелем еще в 4 в. до н. э. Он утверждал, что любой объект характеризуется совокупностью свойств и, говоря, например, про яблоко, мы сразу вспоминаем его запах, цвет и форму, а не само вещество, из которого состоит яблоко. Так, вещество по Аристотелю является контейнером для свойств объекта и любой объект, состоящий из этого вещества, получает все его свойства.

На рис. 1 показана диаграмма модели данных предметной области, описанной выше. Стрелкой обозначено отношение классификации.

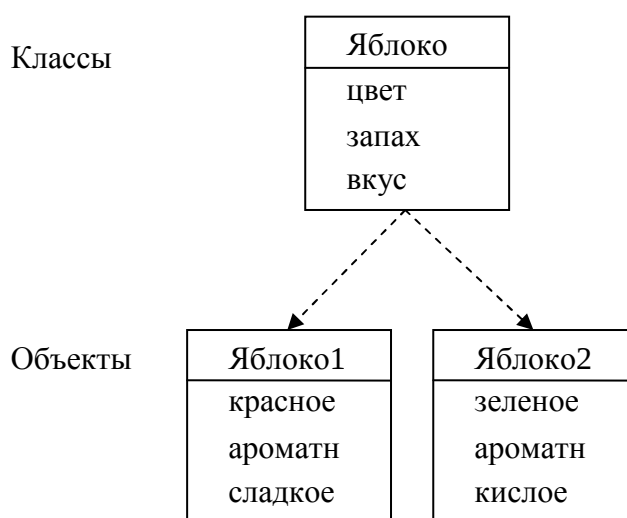


Рис. 1. Пример модели данных для субстанциональной парадигмы.

Реализацией субстанциональной парадигмы Аристотеля в настоящее время стал объектно-ориентированный подход, пользующийся в настоящее время большой популярностью среди программистов. Аналогично субстанциональной парадигме Аристотеля, в объектно-ориентированном подходе объект инкапсулирует в себе набор свойств, которые задают его текущее состояние.

Подход предполагает, что модель мира состоит из *объектов*, принадлежащих одному или нескольким *классам* и *аттрибуты*, значения которых являются свойствами конкретного объекта. Понятие класса появилось как необходимость группировки схожих объектов по каким-либо признакам и повторному использованию описаний схожих объектов.

Для построения модели данных с помощью ООП необходимо, чтобы в выбранном языке все сущности модели были *объектами первого класса* [10]. Это означает, что любая сущность модели данных:

- может быть сохранена в переменной или структурах данных;
- может быть передана в функцию как аргумент;
- может быть возвращена из функции как результат;
- может быть создана во время выполнения программы;
- внутренне самоидентифицируема (независима от именования).

Объект в таких языках как C, C++, Java удовлетворяет описанным выше свойствам, что означает, что он является объектом первого класса. С другой стороны, описание класса во время выполнения программы создать невозможно в этих языках, поэтому описания классов не являются объектами первого класса [7].

В некоторых источниках считается, что набор классов в реальном мире не изменяется. Однако знания человека об этих классах является неточным и неполным, поэтому неизбежна эволюция модели предметной области путем добавления новых классов и внесением уточнений в существующие. Но по причине того, что классы ООП являются объектами второго рода, то необходимо полностью определить их на стадии разработки. Такое ограничение оправдано, когда речь идет о бизнес-логике, когда не должно появляться классов с описанием "чужих" классов, чтобы избежать неправильного или даже опасного поведения системы. Тем не менее, когда речи идет о модели данных реального мира, то такая особенность является скорее ограничением и носит негативное влияние на возможность модели к эволюции.

Таким образом, невозможность определения всего множества классов предметной области на стадии разработки и невозможность добавления и изменения их во время

работы системы приводит к **гносеологической проблеме** использования ООП для описания модели предметной области - модель на уровне классов должна быть полностью создана во время разработки.

Другой проблемой в описании модели данных с помощью ООП является проблема **адекватности**.

В [6] утверждается, что объектно-ориентированный подход к проектированию информационных систем, который автор назвал сущностным (entity), не позволяет строить сложные модели реальных систем, и не позволяет прямо соотнести возникающие сущности с реальностью. Это подтверждается другим примером [1], в котором высказывание о том, что мир состоит из объектов и операций (взаимно-однозначно соответствующих объектам и операциям в модели) разрушается простым примером – когда мы пьем чай, мы не вызываем операцию “*пить*”, принадлежащую объекту “*чашка*” – и вообще, представить это простое действие с помощью ООП очень сложно.

Следующей проблемой в объектно-ориентированном подходе является **проблема объект – атрибут**, которая заключается в том, что одну и ту же сущность реального мира разработчики могут реализовать либо в виде объекта, либо в виде атрибута. Это может возникнуть даже в рамках одного проекта, если разработка ведется большой группой разработчиков.

Использование модели в других предметных областях ограничена тем, что в различных предметных областях одна и та же сущность может быть объектом, а в другой атрибутом, что делает невозможным интегрирование разных предметных областей и использование их совместно в одной системе.

Сильное развитие объектно-ориентированных языков программирования привело к развитию сообщества программистов и развитию инструментария, что является хорошим фактором для снижения стоимости создания модели с помощью такого подхода. Кроме того, в последнее время появилось множество систем для автоматизации создания модели данных на языках ООП. К ним относятся Unified Modeling Language [5], Eclipse Modeling Framework [9], которые позволяют создавать код классов на основе описания модели данных (мета-модели).

Разделения объектов на уровни абстракции в ООП нет. При этом понятие абстрактного класса не предполагает, что объектами этого класса являются некоторые абстрактные объекты реального мира, а лишь означает, что у этого класса не может быть инстансов. Поэтому часто абстрактные классы используют для группировки дочерних

классов, а не для обозначения каких-либо реальных классов сущностей. Эта особенность парадигмы снижает наглядность и читаемость модели данных.

Достоинством данной парадигмы является то, что, основываясь на императивных языках ООП, в ней возможны включения некоторых функций, которые могут изменять модель в зависимости от входных данных, делая модель максимально адаптивной.

Из всего вышесказанного можно сделать вывод, что объектно-ориентированный подход лучше всего подходит для описания бизнес-логики и объектов самой информационной системы, а не для создания модели предметной области информационной системы.

Логически-ориентированная парадигма.

Большинство систем работы со знаниями использует логический подход, в котором модель мира представляется с помощью конъюнкции предикатов первого порядка. Причем это могут быть системы, созданные на объектно-ориентированных языках.

Основой логической парадигмы является *n-местный предикат*, который может принимать значения истинности или ложности. Свойства объектов предметной области описываются с помощью *фактов* и *правил*. В свою очередь, выполнение программы – это доказательство теоремы, в ходе которого строится объект с описанными свойствами.

За счет того, что основой парадигмы является предикат, такой подход обладает хорошими возможностями расширяемости - разные части модели предметной области могут храниться в разных частях хранилища. В этом случае, возможно реализовать федеративное хранение данных, при котором владельцами разных частей модели предметной области могут разные лица и управлять их жизненным циклом независимо.

Одной из главных особенностей парадигмы является возможность порождения новых предикатов по определенным правилам, содержащимся в модели предметной области, что влияет на способность модели к эволюции и адаптивности. Однако это может приводить к **проблеме с производительностью**, когда для моделей данных большого размера вывод может потребовать больших вычислительных ресурсов.

Из вышесказанного, можно сделать вывод, что данная парадигма хорошо подходит для моделей данных, где необходим вывод неявных фактов о предметной области.

Парадигма моделей данных Semantic Web.

Подход сообщества Semantic Web отличается от объектно-ориентированного подхода. Этот подход основан на графовых моделях. Официальным языком является OWL, основанный на RDF.

Основной RDF является триплет и совокупность всех триплетов представляет собой модель ориентированного графа, где узлами графа являются *субъекты* и *объекты* триплетов, а дугами – *предикаты*. Направление предиката имеет значение и всегда идет от субъекта к объекту.

На рис. 2 показана модель предметной области для парадигмы Semantic Web. Для большей наглядности, предметная область выбрана той же, что и для субстанционального подхода.

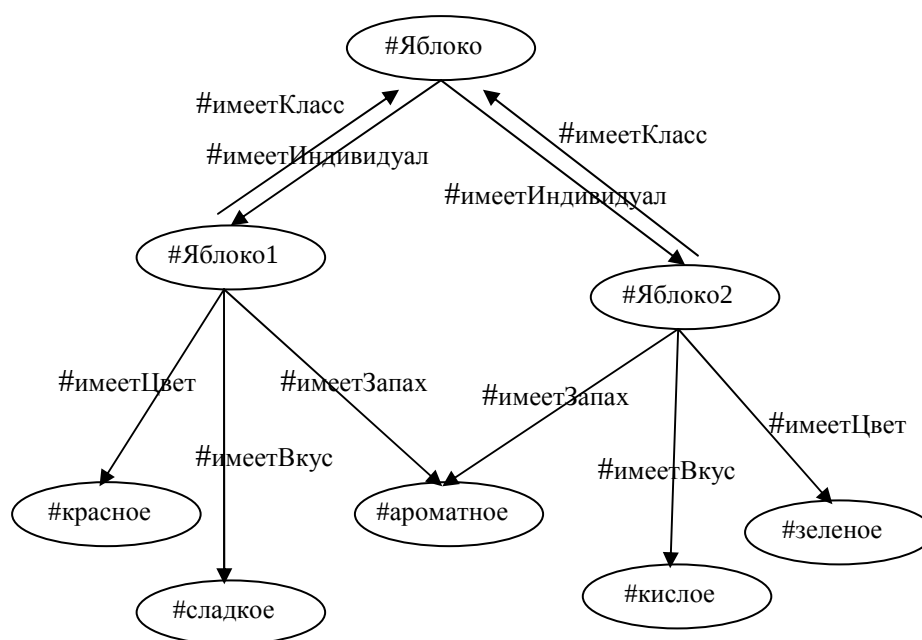


Рис. 2. Пример модели предметной области для парадигмы Semantic Web.

Узлами графа могут быть либо объекты, которые идентифицируются с помощью URI, либо Литералы, содержащие типизированное значение или строка, комбинированная с дополнительным языковым тегом. Такое разделение может приводить к такой же проблеме объект – атрибут, что и в случае объектно-ориентированного подхода.

Проанализировав граф, показанный на рис. 2, можно сделать вывод, что у ребер могут быть инверсные ребра, например у ребра «имеетИндивидуал» существует инверсное ребро «имеетКласс» и при внесении изменений в одно ребро, необходимо вносить соответствующие изменения в инверсное.

По сравнению с субстанциональной парадигмой, свойства объектов не инкапсулированы в объектах, поэтому могут быть добавлены вне объявления объекта, что увеличивает возможности по расширению и модульности модели предметной области.

На уровне языка RDF предикат не является объектом первого класса, так как другой предикат не может указывать на него. Для того, чтобы обойти эту особенность языка, необходимо создать новый объект, который будет представлять этот предикат. Однако появление служебных сущностей в модели будет влиять на адекватность и наглядность высказываний.

Минусом также является то, что в модели RDF все отношения между сущностями являются бинарными, однако при создании модели предметной области часто необходимо введение многомерных отношений. Например, отношение "находится между" является трехместным и не может быть выражено с помощью одного трипла.

Поверх RDF строится язык OWL, который, однако, придерживается той же парадигмы, что и все объектно-ориентированные языки. OWL расширяет RDF таким сущностями, как: *класс (class)*, *индивидуал (individual)*, *объектное свойство (object property)*, *свойство данных (datatype property)*.

Вследствие того, что OWL построен на основе RDF, а не ООП, классы онтологии являются объектами первого класса языка OWL. Это полностью решает решить гносеологическую проблему объектно-ориентированного подхода, а значит, позволяет создавать и модифицировать классы и отношения во время исполнения.

Из вышесказанного, можно сделать вывод, что парадигма Semantic Web позволяет строить более сложные и модульные модели предметных областей, по сравнению с субстанциональной парадигмой.

Парадигма моделей данных ISO 15926.

В международном стандарте ISO 15926 была предложена парадигма описания модели мира, сильно отличающаяся от описанных выше подходов. Эта парадигма берет свое начало в [12], где автор, анализируя объектно-ориентированную парадигму, приходит к выводу, что для описания сложных систем она не подходит и пытается построить собственный «объектный» подход.

Основой парадигмы была взята теория множеств, а основой является онтология верхнего уровня, содержащая 201 понятие, описанных в части 2 стандарта [3]. Такое количество понятий делает эту парадигму более выразительной, по сравнению с другими, описанными ранее. Основными сущностями верхнего уровня являются *сущность (Thing)*, *абстрактный объект (AbstractObject)*, *возможный индивидуал (PossibleIndividual)* и другие.

Главной особенностью парадигмы является поддержка 4D-онтологии [11]. В 4D-онтологии индивидуал описывается с помощью, так называемого, *пространственно-*

временного экстенста. При этом, когда 2 пространственно-временных экстенста совпадают, они образуют один объект. С помощью этого механизма в парадигме вводятся состояния объектов.

Таким образом, в отличие от 3D-парадигмы, в 4D-парадигме есть вся информация о прошлых или будущих запланированных состояниях объекта, на основе которых могут быть сделаны более сложные выводы и принятие более правильных решений. Например, если запланированы плановые технические работы над оборудованием, то планировать какие-то другие действия с оборудованием уже нельзя.

Экстенсионализм этой парадигмы приводит к тому, что в рамках нее можно составлять композиции из объектов, а так же описывать один и тот же физический объект с разных функциональных сторон.

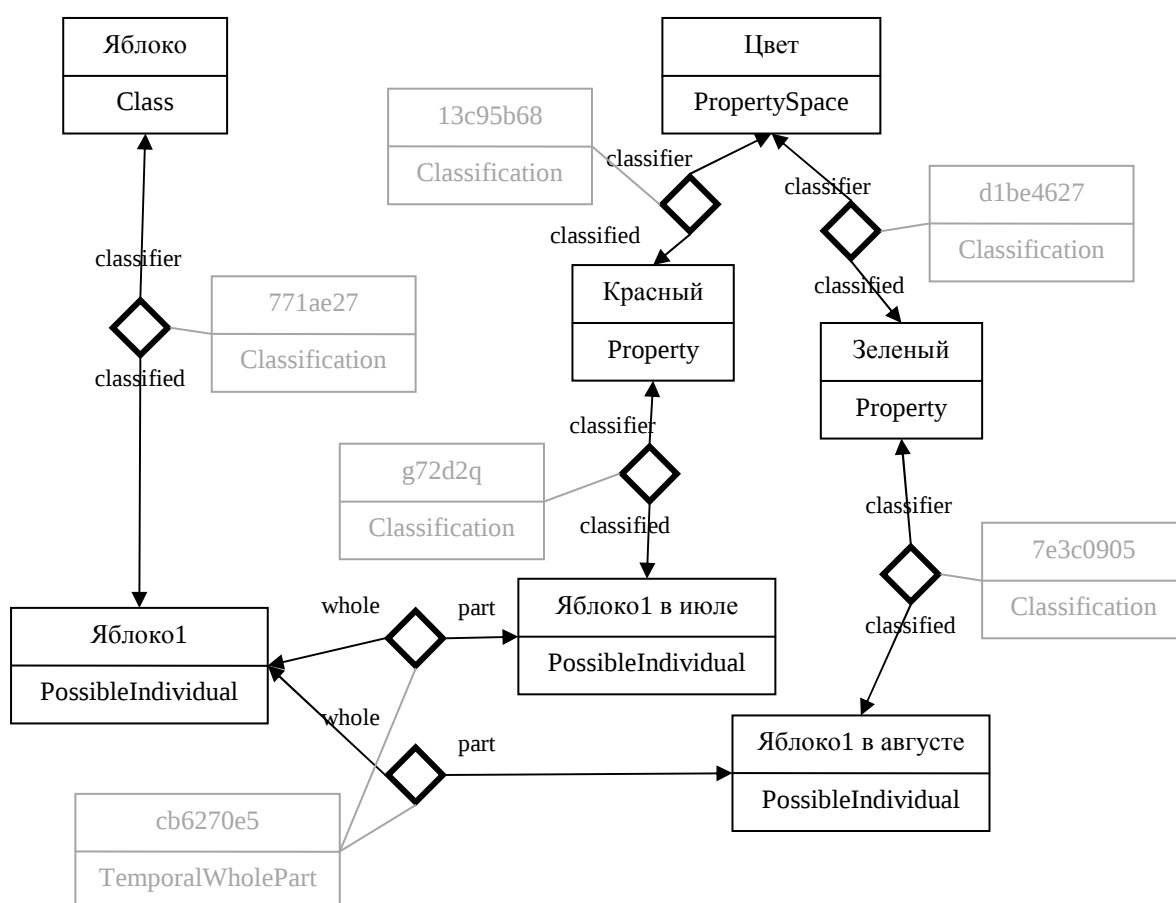


Рис. 3. Пример модели предметной области для парадигмы ISO 15926

На рис. 3 показана диаграмма модели предметной области в парадигме ISO 15926. В этой диаграмме показан один объект «яблоко» с двумя экстенстами – «яблоко в июле» и «яблоко в августе», которые имеют различные свойства цвета. Как видно из диаграммы,

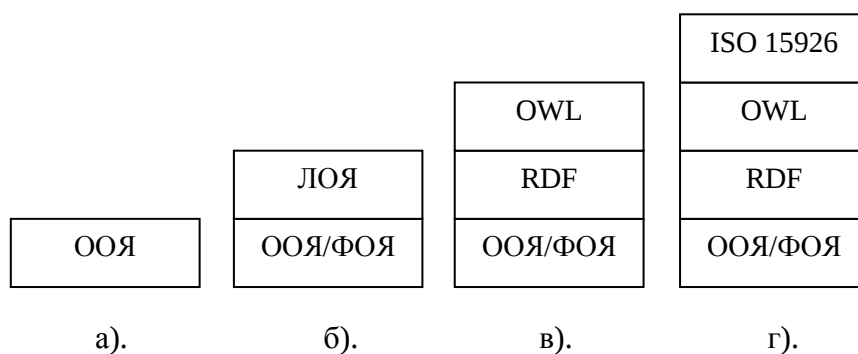
каждое отношение является отдельным объектом модели, а значит объектом первого класса. С одной стороны на такой объект можно ссылаться другими отношениями, с другой стороны для такого отношения необходимо выделение уникального идентификатора, что иногда нецелесообразно при большом их количестве. В зависимости от семантики, в отношения могут входить пространственно-временные экстенды (стадия жизненного цикла объекта), объекты (на всех стадиях), классы (множество объектов), а так же классы классов (множества множеств).

Другие парадигмы моделирования для реализации 4D-онтологии используют искусственные предикаты. При этом выбор, каким именно способом это будет реализовано, ложится полностью на плечи разработчика модели предметной области.

Минусом данной парадигмы является плохое развитие языков описания модели предметной области в соответствии с данной парадигмой, а также малое количества инструментариев для работы.

4. Сравнение парадигм моделей данных

На рис. 4 показаны стеки технологий для построения моделей данных на основе различных парадигм, где ООЯ - объектно-ориентированный язык; ФОЯ - функционально-ориентированный язык; ЛОЯ - логически-ориентированный язык.



а - объектно-ориентированная парадигма; б - логически-ориентированная парадигма; в - парадигма Semantic Web; г - парадигма ISO 15926

Рис. 4. Стек технологий для парадигм описания моделей мира

Существуют инициативы по построению парадигмы ISO 15926 на основе логически-ориентированных языках [4].

Парадигмы оценивались по пятибалльной шкале от «1» – по данному критерию требования не удовлетворяются до «5» – по данному критерию требования полностью удовлетворяются.

Веса для критериев выбраны для систем работы со знаниями (Knowledge-Based Systems), для которых важными критериями является адекватность и универсальность модели, тогда как распространенность инструментария имеет меньший вес.

Оценка критериев качества, описанных в разделе 2, приведена в Таблице ниже.

Критерий	Вес	OOP	LOP	SW	ISO
Адекватность модели					
Использование справочных данных	0.15	2	3	3	5
Точность модели					
Возможность внесения изменений	0.1	2	5	5	5
Универсальность					
Использование в других предметных областях	0.1	2	4	4	5
Расширяемость модели мира	0.1	2	5	5	5
Повторное использование результатов	0.15	3	5	5	5
Возможность федерализации моделей	0.05	3	4	4	5
Целесообразная экономичность					
Простота разработки модели мира	0.05	5	3	2	1
Распространенность языка описания модели	0.05	5	3	2	1
Эффективность инструментария	0.05	5	3	2	1
Другие критерии					
Наглядность	0.05	2	3	4	5
Поддержка промышленных стандартов	0.05	3	3	4	5
Адаптивность	0.05	5	4	3	2
Способность к эволюции	0.05	3	5	4	4
Сумма баллов	1	42	50	47	49
Средний балл		3,23	3,85	3,62	3,77
Взвешенный балл		2,90	4,00	3,85	4,20

где OOP - объектно-ориентированная парадигма; LOP - логически-ориентированная парадигма; SW - парадигма Semantic Web; ISO - парадигма ISO 15926.

На оценку экономичности влияет также тот факт, что построение моделей некоторых парадигм предполагает большего стека технологий (см. Рис. 4).

5. Заключение

В статье были проанализированы основные парадигмы для описания модели мира их основные особенности, а также методика оценки. Варьируя весами критериев возможно выбрать парадигму моделирования для каждого конкретного проекта.

Исходя из среднего балла, максимальную оценку имеет логический подход, однако после взвешивания критериев, ISO 15926 получил больше баллов, поэтому для создания высококачественных моделей данных предпочтительнее использовать модель данных согласно ISO 15926.

При построении парадигмы ISO 15926 на основе стека технологий Semantic Web появляется избыточность - сущность "класс" появляется и на уровне ООП, и на уровне OWL, и в самом ISO 15926. Использование модели гиперграфов позволяет избежать дублирование сущность "класс" в OWL и ISO 15926, а также упростить структуру многомерных отношений, за счет использования многомерных отношений гиперграфов, вместо направленных бинарных в RDF.

Список литературы

1. Cook S., Daniels J. Object-Oriented Methods and the Great Object Myth // Object in Europe. Vol. 1. 1994. P. 45-66.
2. Heller M. Philosophy in Science: An Historical Introduction. Berlin: Springer Berlin-Heidelberg, 2011. 167 p.
3. ISO 15926: Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas production facilities - Part 2: Data model. Available at: http://www.iso.org/iso/ru/home/store/catalogue_tc/catalogue_detail.htm?csnumber=29557, accessed 27.07.2014.
4. ISO 15926: Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas production facilities - Part 7: Implementation methods for the integration of distributed systems: Template methodology. Available at: http://www.iso.org/iso/ru/catalogue_detail.htm?csnumber=52455 accessed 27.07.2014.
5. OMG Unified Modeling Language TM (OMG UML). Available at: <http://www.omg.org/spec/UML/2.5/Beta2/PDF>, accessed 20.04.2014.
6. Partridge C. Business Objects: Re-Engineering for Re-Use, 2nd Edition. Oxford: Butterworth-Heinemann, 2010. 566 p.

7. Scott M. Programming Language Pragmatics. San Francisco, CA: Morgan Kaufmann Publishers, 2010. 944 p.
8. Sider T. Four-Dimensionalism. An Ontology of Persistence and Time. Oxford: Oxford University Press, 2003. 288 p.
9. Steinberg D., Budinsky F., Paternostro M., Merks E. EMF: Eclipse Modeling Framework, 2nd Edition. Upper Saddle River, NJ: Addison-Wesley Professional, 2008. 744 p.
10. Strachey C. Fundamental Concepts in Programming Languages, 2000. Available at: <https://www.itu.dk/courses/BPRD/E2009/fundamental-1967.pdf> accessed 12.10.2014.
11. West M. An introduction to 4 dimensionalism in data models. Available at: <http://www.matthew-west.org.uk/publications/AnIntroductionTo4DimensionalismInDataModels.pdf> accessed 13.05.2014.
12. West M. Developing high quality data models. San Francisco, CA: Morgan Kaufmann Publishers, 2011. 408 p.
13. Большакова Е.И., Баева Н.В., Груздева Н.В., Горячая И.В. Парадигмы и языки в обучении информатике и программированию. // Материалы международной научно-практической конференции “Перспективные инновации в науке, образовании, производстве и транспорте 2012”. Режим доступа: http://istina.msu.ru/media/publications/articles/a49/a24/1745046/statya_Bolshakova.doc (дата обращения 10.11.2012).
14. Тараров Я. В. Аристотелевская онтология как «онтологическая парадигма» классической физики и космологии // Эпистемология и философия науки. 2009. Т. 21. № 3. С. 122-137.
15. Прокимнов Н. Н. Моделирование систем: учебное пособие, М., 2007. Режим доступа: <http://www.e-biblio.ru/cgi-bin/lib.pl?func=book&level=6&book=2524> (дата обращения 12.10.2012).