

Семантическая интеграция и адаптация данных в гетерогенных корпоративных информационных системах

М.А. Лаверычев, И.О. Саркисова, А.А. Соколов

Московский государственный технологический университет «Станкин»

Аннотация: Статья посвящена проблемам интеграции и обработки разнородных данных в условиях цифровой трансформации как внутри одной компании, так и при организации взаимодействия между различными участниками бизнес-процессов. Основное внимание уделено вопросам совместной работы компаний-производителей оборудования и промышленных предприятий, подчеркивается важность согласования и преобразования данных при взаимодействии гетерогенных информационных систем. Рассматривается проблема интеграции исторических данных, сложности, возникающие при переходе на новую инфраструктуру, и предлагается способ решения по аналогии с принципами работы открытых стандартов, таких как OpenCL. Особое внимание уделяется необходимости предоставления полных и непротиворечивых данных, разработке эффективных механизмов семантической интеграции и использованию онтологических подходов для устранения трудностей при сопоставлении и интерпретации разнородных данных. Подчеркивается важность постоянного обновления словаря метаданных и формирования связей между различными источниками данных для обеспечения качественной и надежной интеграции. Рассматриваемые подходы направлены на создание устойчивых механизмов обмена информацией между различными участниками бизнеса для принятия управленческих решений.

Ключевые слова: цифровая трансформация, гетерогенные системы, онтология, семантическая интеграция, метаданные, маппинг данных.

В современном мире глобальные и отечественные компании и предприятия активно проходят этап цифровой трансформации, переводя все больше привычных процессов в цифровую среду [1, 2]. Вслед за цифровизацией неизбежно увеличиваются объемы создаваемых и обрабатываемых данных. Также создаются новые технологии и инструменты для работы с данными, совершенствуются методы их хранения и обработки.

Несомненно, очевидной целью таких изменений является улучшение и повышение эффективности работы с большим объемом данных. Тем не менее, появление нового инструментария и/или форматов данных может повлечь за собой ряд проблем, способных усложнить и замедлить работы с различными данными.

Одна из ключевых сложностей – необходимость выстраивания процессов преобразования и форматирования данных при переходе на новый инструментарий. Зачастую бывает затруднительно использовать информацию, накопленную в устаревших системах, например, из-за проблем с совместимостью форматов.

Такие исторические данные остаются ценным ресурсом для организаций, помогая принимать более взвешенные управленческие решения и учитывать накопленный опыт. В связи с этим могут возникнуть трудности как при работе с такими данными в целом, так и при попытке извлечь из них полезную информацию.

При этом источники данных редко ограничиваются пределами только самой организации – информация поступает от иных предприятий, организаций-партнеров, клиентов, контрагентов и др. Разнообразие технологических решений и специфики работы различных организаций приводит к высокой степени разрозненности и разноформатности данных. Это усложняет их обработку, снижая эффективность и качество анализа. Конечно, можно определить четкие стандарты обмена данными, но реализация такого подхода требует значительных организационных и временных затрат.

Дополнительно, по мере роста бизнеса и расширения продуктовой линейки компании появляются новые бизнес-направления и растет число подразделений внутри организации. Это ведет к росту объемов обрабатываемых данных и появлению новых источников информации. Разные отделы компании работают с различными задачами, из-за чего скорость, качество и объем поступающих данных могут существенно различаться. При этом хранение информации может осуществляться как централизованно – в корпоративном хранилище данных, так и на уровне

отдельных подразделений. Это повышает риск дублирования данных, а их актуализация разными сотрудниками может приводить к противоречиям.

Несмотря на разнообразие бизнес-направлений внутри компании, существуют задачи, требующие консолидации разрозненной информации. Это могут быть кросс-функциональные проекты, подготовка отчетности на уровне всей организации, а также выполнение требований аудиторов и регуляторов. Разнообразие источников данных, отличающихся по структуре и содержанию, становится серьезным вызовом для выполнения подобных задач.

Данные, которым свойственны разрозненность и различия в типах данных, моделях данных, языках описания процедур, структурах данных и пр., характеризуются как гетерогенные. Невозможность отказаться от работы с такими данными, а также их особенности предъявляют повышенные требования к системам, обеспечивающим их хранение и обработку.

Таким образом, существующая в настоящее время потребность в повышении эффективности обработки информации с учетом роста скорости ее генерации и объемов дополняется увеличением ее разрозненности. Помимо этого, общий тренд на повышение динамичности бизнеса предъявляет существенные требования к скорости анализа информации. Возможность оперативно предоставить наиболее полную и качественную информацию для принятия обоснованного управленческого решения является необходимостью для современных организаций.

Различие в форме и структуре данных не всегда следует рассматривать как недостаток, поскольку оно часто обусловлено спецификой процессов в отдельных предприятиях и их целями. Разнообразие форматов и содержания данных во взаимодействующих организациях требует разработки современных методов и технологий, обеспечивающих их согласованную обработку и эффективное использование. Важно четко

определять критерии интеграции данных, чтобы они соответствовали задачам, для решения которых применяются.

Работая над концепцией сервиса производственных услуг, суть которого состоит в агрегации свободного станочного времени на разных предприятиях в одну систему [3], возникает потребность интеграции данных от разных предприятий в единую платформу [4]. Эта платформа должна обеспечить взаимодействие производственных предприятий, производителей оборудования и клиентов в рамках единого контекста. Если общение клиента с системой (в рассматриваемой концепции системой владеет производитель оборудования) стандартизируется в рамках одной системы, то общение с производственными предприятиями стандартизовать и формализовать на уровне одной системы проблематично. Ведь степень цифровизации предприятий, а также конкретный выбор программных решений на каждом предприятии может отличаться.

Стандартизация общения клиентов с системой осуществляется посредством единой платформы, однако интеграция производственных предприятий требует гибкости и учета особенностей их ИТ-инфраструктур. Создавая всю систему сверху вниз с самого начала, можно добиться унификации и интероперабельности на уровне единых подходов, унификации программных решений и т.д. Но когда система создается путем объединения уже существующих предприятий поменять всю ИТ-инфраструктуру предприятия будет сложно и дорого. Более того, необходимость таких преобразований будет воспринята негативно большинством предприятий, так как потребует инвестиций времени, ресурсов. Поэтому необходимо создавать систему и интерфейсы взаимодействия с предприятиями как с гетерогенными системами. Т.е. должен быть некий сервисный уровень, который будет заниматься приведением данных от предприятия к требуемому главной системе виду и

наоборот, будет пересылать данные от главной системы к подчиненной в понятном ей виде.

Похожий подход к работе с подчиненными системами можно встретить в программном пакете OpenCL, который используется для решения сложных вычислительных задач [5, 6]. Платформа OpenCL состоит из хоста, соединенного с устройствами, поддерживающими OpenCL. Каждое OpenCL-устройство состоит из вычислительных блоков, которые далее разделяются на один или несколько элементов-обработчиков (processing elements, PE).

OpenCL-приложение выполняется на хосте в соответствии с нативными моделями его платформы. OpenCL-приложение отправляет с хоста команды устройствам на выполнение вычислений на PE. В рамках вычислительного блока PE выполняют один поток команд как одиночный поток команд и множественный поток данных (single instruction, multiple data, SIMD) блоки (одна инструкция выполняется всеми одновременно, обработка следующей инструкции не начнется, пока все PE не завершат исполнение текущей инструкции), либо как единую программу и множество данных (single program, multiple data, SPMD) блоки (у каждого PE собственный счетчик инструкций).

OpenCL обрабатывает команды, поступающие от хоста, при этом приложение не связано жестко с OpenCL, а значит всегда можно подменить реализацию OpenCL, не нарушив работоспособность программы. Даже если будет создано такое устройство, которое не укладывается в модель «OpenCL-устройства», для него можно будет создать реализацию OpenCL, транслирующую команды хоста в более удобный для устройства вид.

Для координации работы всех этих устройств в гетерогенной системе всегда есть одно «главное» устройство, которое взаимодействует со всеми остальным с помощью OpenCL API.

Поэтому OpenCL исходит из наиболее общих предпосылок, дающих представление об устройстве с поддержкой OpenCL: так как это устройство предполагается использовать для вычислений – в нем есть некий «процессор» в общем смысле этого слова. Нечто, что может исполнять команды. Так как OpenCL создан для параллельных вычислений, то такой процессор может иметь средства параллелизма внутри себя. Также элементарным способом наращивания производительности параллельных вычислений является установка нескольких таких процессоров на устройстве. И естественно в гетерогенной системе может быть несколько таких OpenCL-устройств с различной архитектурой.

OpenCL предоставляет программисту низкоуровневый интерфейс программирования приложения (application programming interface, API) (рис.1), через который он взаимодействует с ресурсами устройства. OpenCL API может либо напрямую поддерживаться устройством, либо работать через промежуточный API (как в случае NVidia), это зависит от конкретной реализации и не описывается стандартом.

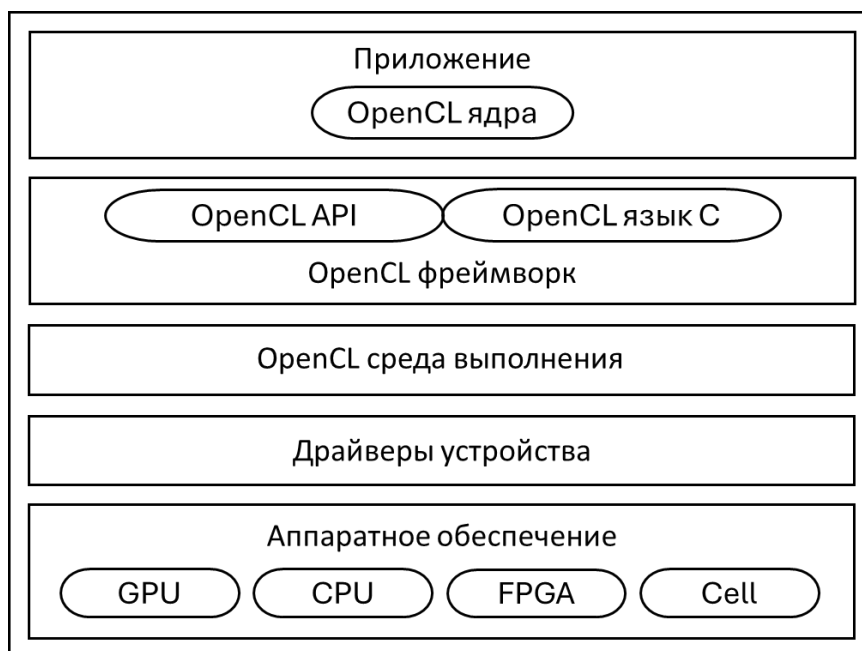


Рис.1 – Иерархическая структура OpenCL.

Аналогично платформе OpenCL, разрабатываемая система должна обеспечивать возможность работы с различными информационными системами без необходимости полной замены их внутренней инфраструктуры. Проводя прямые аналогии: OpenCL устройства – это информационные системы предприятия, хост – это платформа, а OpenCL драйверы – сервисный уровень, обеспечивающий подготовку данных для обработки платформой. Платформа – система производителя оборудования (аналог хоста), через которую клиенты оформляют заказы (аналог вычислительных задач). Есть предприятия (аналог OpenCL-устройств), которые должны выполнить задачу. Однако у каждого предприятия могут быть свои системы для работы с производством (MES/ERP/CSRP и другие), свои принципы ведения учета. Помимо этого, необходима и обратная связь, от предприятия к системе. Например, для передачи производственного плана с указанием свободных слотов времени и доступного оборудования, отчета о ходе выполнения заказа и т.д. Все эти данные должны быть адекватно обработаны системой, чтобы внутри себя работать с формализованными данными стандартизированного вида, вне зависимости от того, как они выглядят в момент передачи от производственного предприятия системе. Все это требует разработки уникального интерфейса для подключения конкретного предприятия к системе. Это решение позволяет главной системе функционировать корректно и не требует перестройки всей IT-инфраструктуры подчиненных предприятий.

Для анализа жизненного цикла заказа была построена модель в нотации диаграммы последовательности на языке UML (рис.2.).

На основании представленной диаграммы последовательности процесса выполнения заказа в системе можно построить математическую модель, которая описывает взаимодействия между участниками системы и их состояния.

Основные элементы:

1. Участники:

- Заказчик (З)
- Система (С)
- Отдел технологической подготовки (ОТП)
- Производственное предприятие (ПП)

Диаграмма последовательности процесса выполнения заказа в системе

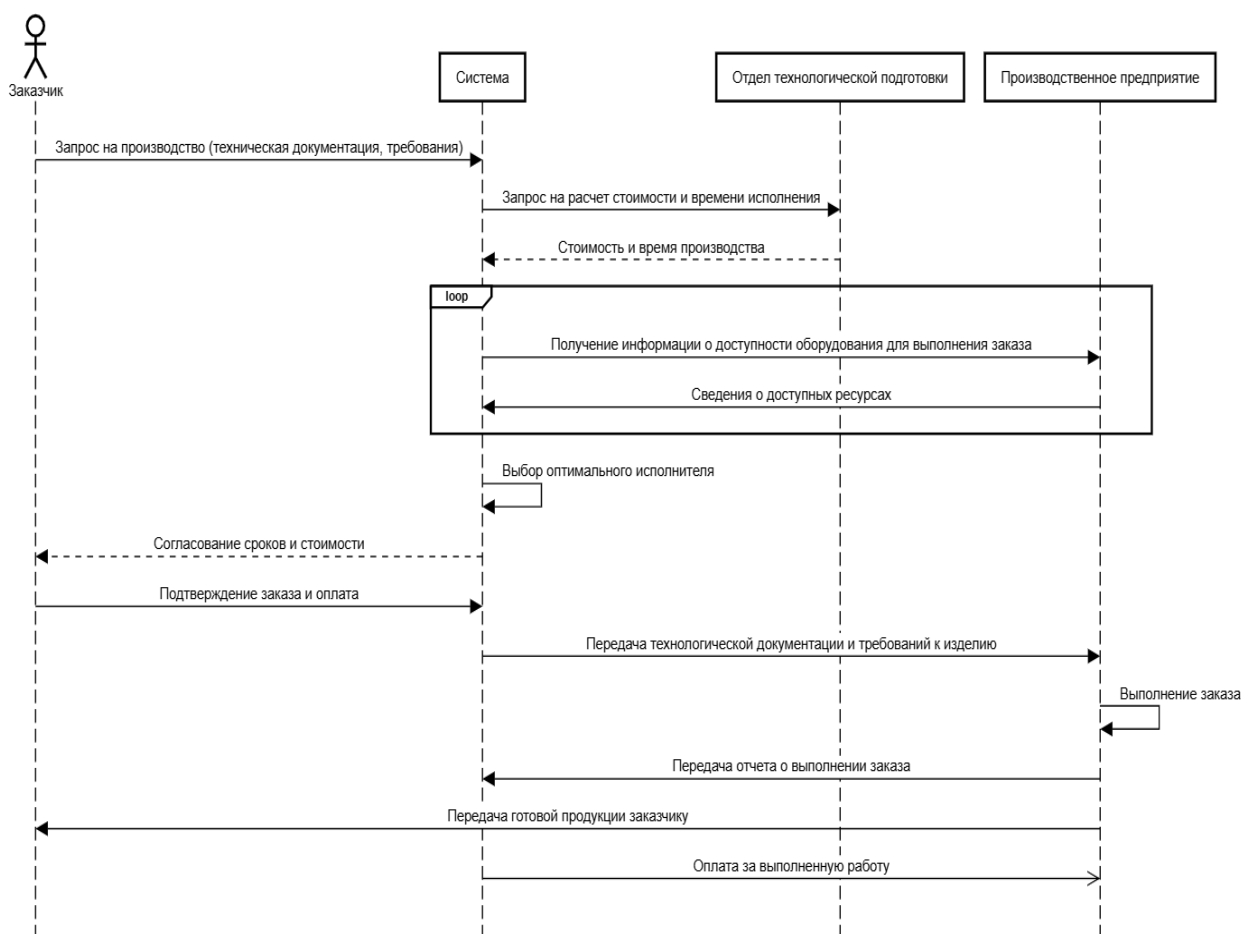


Рис. 2. – Диаграмма последовательности

2. Состояния участников.

Заказчик:

$S_3 = \{\text{idle (начальное состояние), waitingConfirmation (ожидание подтверждения), confirmed (заказ подтвержден), paid (заказ оплачен), receivedProduct (заказ получен)}\}$

Система:

$S_C = \{\text{idle (начальное состояние), awaitingCostTime (ожидание расчета времени и стоимости выполнения заказа), choosingExecutor (выбор исполнителя), awaitingApproval (ожидание подтверждения от исполнителя), awaitingPayment (ожидание оплаты), orderInProgress (выполнение заказа), orderCompleted (заказ выполнен)}\}$

Отдел технологической подготовки:

$S_{ОТП} = \{\text{idle (начальное состояние), calculating (расчет времени и стоимости выполнения заказа)}\}$

Производственное предприятие:

$S_{ПП} = \{\text{idle (начальное состояние), checkingResources (оценка свободных ресурсов), executingOrder (производство заказа), completedOrder (заказ выполнен)}\}$

3. Действия участников:

Заказчик:

$A_3 = \{\text{requestProduction (запрос на производство), confirmOrder (подтверждение заказа), payOrder (оплата заказа)}\}$

Система:

$A_C = \{\text{requestCostTime (запрос времени и стоимости выполнения заказа), requestResourceAvailability (запрос доступности ресурсов на предприятие), chooseExecutor (выбор исполнителя), sendOrderDetails (отправка комплекта документации на выбранное предприятие), receiveReport (получение отчета), transferProduct (отправка заказа), payForWork (оплата выполненного заказа)}\}$

Отдел технологической подготовки:

$A_{отп} = \{ \text{calculateCostTime} (\text{расчет времени и стоимости выполнения заказа}) \}$

Производственное предприятие:

$A_{пп} = \{ \text{checkResources} (\text{проверка доступных ресурсов}), \text{executeOrder} (\text{выполнение заказа}), \text{reportCompletion} (\text{отправка отчета о завершении производства}), \text{deliverProduct} (\text{отправка заказа}) \}$

Модель процесса выполнения заказа можно представить следующим образом: $M = (S, A, T)$, где:

- $S = S_3 \cup S_C \cup S_{отп} \cup S_{пп}$,
- $A = A_3 \cup A_C \cup A_{отп} \cup A_{пп}$,
- $T = \{ T_{зс}, T_{сотп}, T_{отпс}, T_{спп}, T_{ппс}, T_{сс}, T_{сз}, T_{зс}, T_{спп}, T_{пп}, T_{ппс}, T_{сз}, T_{спп} \}$

Наиболее сложными, с точки зрения работы с информацией, являются функции переходов между «Системой» и «Производственным предприятием» для проверки ресурсов ($T_{спп}$ ($\text{choosingExecutor}, \text{requestResourceAvailability}$) = checkingResources); между «Производственным предприятием» и «Системой» для передачи сведения о ресурсах ($T_{ппс}$ ($\text{checkingResources}, \text{checkResources}$) = choosingExecutor); между «Системой» и «Производственным предприятием» для начала работы по заказу ($T_{спп}$ ($\text{orderInProgress}, \text{sendOrderDetails}$) = executingOrder); между «Производственным предприятием» и «Системой» для передачи отчета о завершении ($T_{ппс}$ ($\text{completedOrder}, \text{reportCompletion}$) = orderCompleted).

Проверка доступности ресурсов предприятия (станков, персонала) подразумевает получение информации о количестве свободного станочного времени предприятия. Эти данные подготавливаются на стороне производственного предприятия с помощью MES системы, календарного планирования и других инструментов организации производства.

Передача сведений о ресурсах в главную систему происходит в виде выгрузки из базы данных предприятия, файла JSON, XML, CSV или NoSQL

файла. Конкретный формат и вид передаваемых данных может различаться, что и требует приведения входных данных к виду, понятному системе.

Пример выгрузки данных о загрузке ресурсов в формате XML из 1С MES может иметь вид:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResourceLoadReport date="2023-10-25T15:00:00">
  <Resources>
    <!-- Загрузка оборудования -->
    <Resource id="WC-001" type="WorkCenter">
      <Name>Токарный станок №1</Name>
      <Capacity unit="hours/day">16</Capacity>
      <Schedule>
        <Day date="2023-10-25">
          <PlannedLoad>8</PlannedLoad>
          <ActualLoad>6.5</ActualLoad>
          <Reserved>
            <Order id="PO-12345" hours="2"/>
          </Reserved>
        </Day>
        <Day date="2023-10-26">
          <PlannedLoad>10</PlannedLoad>
          <ActualLoad>0</ActualLoad>
        </Day>
      </Schedule>
    </Resource>
  </Resources>
</ResourceLoadReport>
```

Выполнения задания на стороне производственного предприятия начинается после получения всей необходимой документации. Задание на производство, включающее технологическую карту, исполняемые программы и чертежи, должно быть передано таким образом, чтобы все компоненты были доступны и понятны производственным предприятием. Есть несколько способов передачи таких заданий, каждый из которых имеет свои преимущества и недостатки, представленные в таблице 1.

Система может хранить комплект документации на файл-сервере или в базе данных, а в зависимости от возможностей производственного

предприятия передавать данные в виде архива или напрямую загружать данные на сервер предприятия или во внутренние производственные системы.

Таблица №1

Сравнение способов хранения и передачи заданий

	Электронный документ (PDF, DOCX и др.)	Архив (ZIP, RAR и др.)	Облачное хранилище (Google Drive, Яндекс.Диск и др.)	Спец. системы управления производством (ERP/MES)
Достоинства	Все документы собраны в одном месте. Легко пересылать и открывать на разных устройствах.	Упрощает передачу большого количества файлов. Позволяет сохранить структуру папок и файлов.	Доступность с любого устройства. Возможность совместной работы и контроля версий.	Автоматизация процессов. Контроль выполнения задач. Обеспечение безопасности данных.
Недостатки	Может возникнуть проблема с совместимостью программных средств для открытия файла. Возможен риск потери части информации при передаче.	Требуется наличие программного обеспечения для распаковки архива. Возможно повреждение архива при передаче.	Зависимость от интернет-соединения. Ограниченная безопасность данных.	Высокая стоимость внедрения и поддержки. Сложность настройки и обучения персонала.

По завершению выполнения заказа производственное предприятие должно отчитаться перед главной системой. Среди прочего она должна передать информацию о: отправке готового изделия, время выполнения, наличие/отсутствие брака, затраченное количество инструмента и т.д. Эти

данные генерируются на стороне производственного предприятия посредством MES системы, ERP, другой системой или вручную. Эти данные также должны прийти в базу данных главной системы и быть правильно интерпретированы, вне зависимости от специфики выгрузок разных систем.

В указанных условиях возникает необходимость обеспечения интеграции данных, обмен которыми осуществляется между производственными предприятиями и производителями оборудования. Интеграция данных предполагает объединение данных, полученных из различных источников, для объединенного представления их пользователю. Целью интеграции данных является предоставление полного и непротиворечивого набора данных, сформированного на основе множества разрозненной информации, поступившей из разных источников. В рамках данного процесса решаются задачи по интеграции значений, синтаксиса, структуры семантики данных, каждая из которых предполагает собственные методы реализации [7].

Одной из наиболее сложных и актуальных составных частей процесса интеграции является семантическая интеграция. Представляется, что задача по обеспечению программной совместимости может быть решена на техническом уровне, в то время как задача по корректной интерпретации полученных данных требует выработки и применения специализированных подходов. В ходе решения подобной задачи в первую очередь представляется необходимым учитывать контекст, в рамках которого осуществляется взаимодействие между системами, а также обеспечивать соответствие обрабатываемой информации такому контексту. Наличие единого контекста, в условиях которого осуществляется интеграция, позволяет выделить критерии семантической интеграции, а также сформировать определенную онтологию. Ключевая цель семантической интеграции заключается в обеспечении соответствия множества различных и разрозненных значений

единому контексту. В основе семантической интеграции находится необходимость учета семантического (смыслового) содержания обрабатываемых данных, а не только их структурных характеристик [8]. Семантическая интеграция способна повысить согласованность и качество обработки информации для ее дальнейшего эффективного применения.

Следует более подробно рассмотреть возможные применимые методы семантической интеграции для решения указанных ранее задач и проблем.

Одним из таких методов может являться интеграция на основе метаданных. Метаданные представляют собой структурированную информацию, описывающую характеристики данных, включая сведения об их содержании, формате, происхождении, условиях создания и изменениях, что обеспечивает возможность их эффективного поиска, управления и интеграции в информационных системах. Критерием эффективной интеграции в рамках данного подхода является совпадение значений метаданных по определенному количеству измерений. При этом, для эффективного построения системы метаданных следует исходить из единого формата таких данных, в то время как гетерогенность систем, между которыми осуществляется взаимодействие, не позволяет это сделать. Таким образом, при различии форматов данные могут обладать различными характеристиками, хранить больший или меньший объем информации. Так, например, в различных таблицах информация о дате может иметь как различный формат и/или синтаксис, так и вовсе храниться в нескольких столбцах/строках сразу.

В целом интеграция на основе метаданных предполагает сравнение содержания и состава метаданных различных наборов для определения возможности их семантической интеграции. Для эффективной реализации такого подхода может быть применен метод интеграции на основе онтологий [9, 10]. Как правило, выделяют формирование множества онтологий либо

единой онтологии. Наиболее применимым представляется подход, предполагающий формирование единой онтологии посредством интеграции нескольких исходных онтологий (рис.3). Такая онтология формируется посредством согласованного объединения словарей, используемых для определения семантики интегрируемых наборов данных. Сами по себе словари обеспечивают нормализацию терминологии и позволяют унифицировать обозначения, используемые в разных источниках. Так, например, в различных системах могут использоваться разные термины для описания схожих сущностей, что может усложнить обмен данными между организациями. Интерпретация таких терминов в условиях единого контекста является необходимостью в условиях высокой гетерогенности информационных систем. Данная задача решается посредством «отображения онтологий» (от англ. *Ontology Mapping*), которая предполагает нахождение семантических связей схожих элементов из разных онтологий, то есть установление соответствия между несколькими онтологиями [11].



Рис. 3 – Фрагмент онтологии предметной области.

Важным условием эффективного применения такого метода является постоянное обновление и актуализация используемого словаря. В условиях частого изменения задач анализа и/или части передаваемой информации необходимо обеспечивать наличие актуальной терминологии. При этом за счет наличия единого контекста можно обеспечить эффективность

применения такой терминологии, так как обмен информацией предполагается в рамках определенной бизнес-задачи или направления. Постоянное обновление словаря способствует расширению и обогащению семантического контекста, в том числе посредством сбора дополнительной информации об уже имеющихся терминах, например, их синонимах, гипонимах, гиперонимах.

Применение такого подхода может быть эффективно в условиях, когда осуществляется обмен информацией, которая должна соответствовать определенным четко заданным критериям. В таком случае следует исходить из того, что необходимо получить и/или передать определенный набор информации. В связи с этим, изначально ожидается получение конкретного перечня данных. Сложность же заключается в том, что такие данные могут различаться по формату и объему, в связи с чем необходимо корректное их сопоставление.

Для решения подобной задачи применяется маппинг (сопоставление) данных. Маппинг представляет собой процесс сопоставления и преобразования данных из разных систем посредством установления соответствия между понятиями и атрибутами. Кроме того, в случае двунаправленного обмена информацией наличие выстроенных связей между различными онтологиями позволяет обеспечивать постоянное взаимодействие различных систем (рис.4.).

Наличие сформированной онтологии и выстроенных в её рамках связей между семантиками различных источников также может обеспечить эффективную интеграцию новых источников. В случае, если семантика нового источника может быть сопоставлена с уже имеющейся, то задача интеграции такого источника может решаться вполне успешно. Кроме того, потеря актуальности какого-либо подхода в рамках интеграции также может быть учтена при построении новых связей, обеспечивая актуальность общего

контекста. Накапливаемый в рамках взаимодействия разрозненных систем опыт позволяет учитывать его для построения новых связей и обеспечения их согласованности.

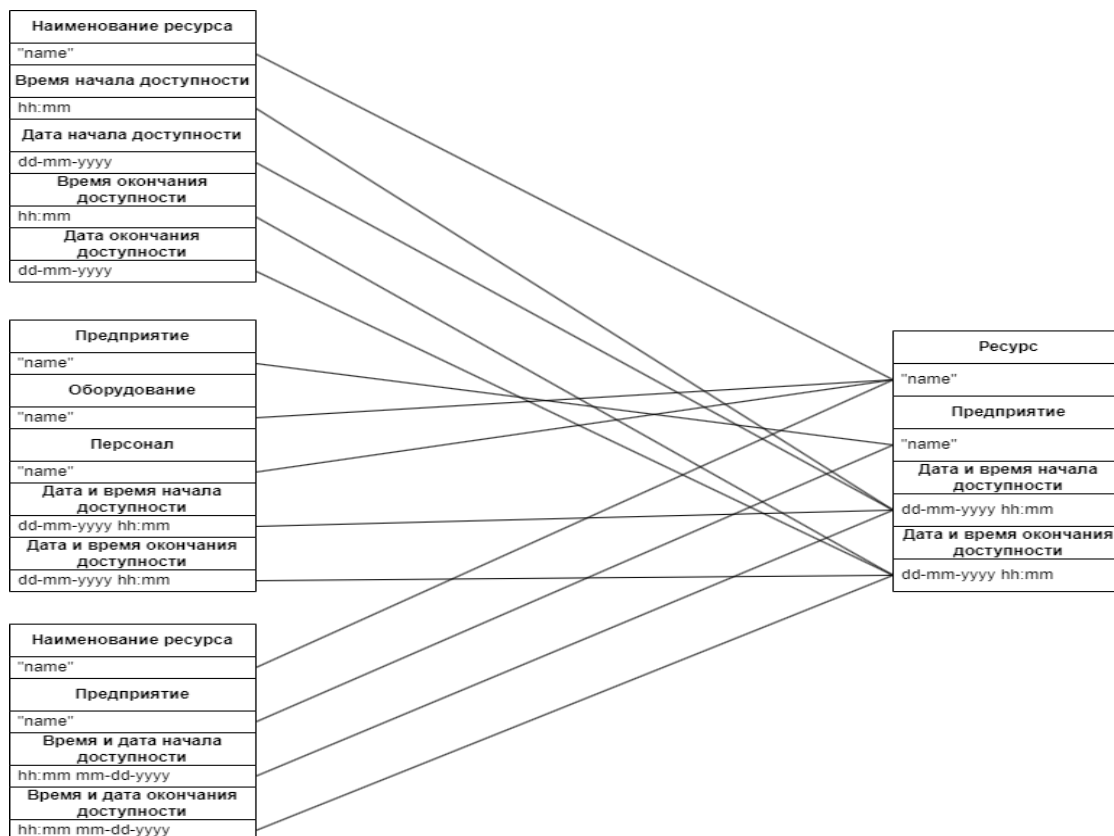


Рис. 4 – Пример маппинга данных в рамках передачи сведений о доступности ресурсов предприятия.

Предложенная методика помогает решать задачи сравнения и объединения данных, соответствующих различным стандартам и форматам. Такой подход направлен на создание устойчивых механизмов обмена информацией между различными участниками бизнеса, обеспечивая своевременное получение необходимых данных для качественного принятия управленческих решений.

Литература

1. Сигачева М. А., Кабанова С. М., Лаверычев М. А. Формализация возможности внедрения проблемно ориентированной системы управления // Известия Тульского государственного университета. Технические науки. 2024. №. 3. С. 83-87.
2. Хоронько Л.Я., Шеина С.Г., Аль-Згуль И.Х., Аль-Згуль С.Х. Цифровая трансформация образовательной среды технического университета: новые подходы к формированию профессиональных компетенций // Инженерный вестник Дона, 2025, №2. URL: ivdon.ru/ru/magazine/archive/n2y2025/9907.
3. Лаверычев М. А., Саркисова И. О. Метод управления информационными процессами при использовании коммерческой концессии в промышленности // Известия Тульского государственного университета. Технические науки. 2024. №3. С. 16-21.
4. Althati C., Tomar M., Shanmugam L. Enhancing Data Integration and Management: The Role of AI and Machine Learning in Modern Data Platforms // Journal of Artificial Intelligence General science (JAIGS). 2024. V. 2. №1. pp. 220-232.
5. Owaida M., Bellas N., Daloukas K., Antonopoulos C.D. Synthesis of platform architectures from OpenCL programs // 2011 IEEE 19th Annual International Symposium on Field-Programmable Custom Computing Machines. IEEE, 2011. pp. 186-193.
6. OpenGL Overview - The Khronos Group Inc. URL: khronos.org/opengl/ (дата доступа 10.05.2025).
7. Гочева Д., Георгиев Д., Батчкова И. Семантический подход к интероперабельности и интеграции данных в управлении производственными операциями // Проблемы машиностроения и автоматизации. 2016. № 4. С. 37-44.

8. Городецкий В.И., Тушканова О.Н. Семантические технологии для семантических приложений // Искусственный интеллект и принятие решений. 2018. № 4. С. 61-71.

9. Жилиев А.А. Онтологии как инструмент создания открытых мультиагентных систем управления ресурсами // Онтология проектирования. 2019. № 2 (32). С. 261-281.

10. Бадецкий А. П., Медведь О. А. Онтологический подход к разработке единой базы знаний мультимодальных перевозок // Известия Петербургского университета путей сообщения. 2023. Т. 20. №1. С. 182-193.

11. Карпенко А.П., Сухарь Р.С. Методы отображения онтологий. Обзор // Машиностроение и компьютерные технологии. 2009. № 1. URL: cyberleninka.ru/article/n/metody-otobrazheniya-ontologiy-obzor.

References

1. Sigacheva M.A., Kabanova S.M., Laverychev M.A. Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskie nauki. 2024. №3. pp. 83-87.

2. Khoron'ko L. Ya., Sheina S.G., Al'-Zgul' I.Kh., Al'-Zgul' S.Kh. Inzhenernyj vestnik Dona, 2025, №2. URL: ivdon.ru/ru/magazine/archive/n2y2025/9907.

3. Laverychev M. A., Sarkisova I. O. Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskie nauki. 2024. №3. pp. 16-21.

4. Althati C., Tomar M., Shanmugam L. Journal of Artificial Intelligence General science (JAIGS). 2024. V. 2. №1. pp. 220-232.

5. Owaida M., Bellas N., Daloukas K., Antonopoulos C.D. 2011 IEEE 19th Annual International Symposium on Field-Programmable Custom Computing Machines. IEEE, 2011. pp. 186-193.

6. OpenGL Overview - The Khronos Group Inc. URL: khronos.org/opengl/ (accessed 10.05.2025).

7. Gocheva D., Georgiev D., Batchkova I. Problemy mashinostroeniya i avtomatizatsii. 2016. № 4. pp. 37-44.



8. Gorodetskiy V.I., Tushkanova O.N. Iskusstvennyj intellekt i prinyatie resheniy. 2018. № 4. pp. 61-71.
9. Zhilyaev A.A. Ontologiya proektirovaniya. 2019. № 2 (32). pp. 261-281.
10. Badetskiy A. P., Medved' O. A. Izvestiya Peterburgskogo universiteta putey soobshcheniya. 2023. V. 20. №1. pp. 182-193.
11. Karpenko A.P., Sukhar' R.S. Mashinostroenie i komp'yuternye tekhnologii. 2009. № 1. URL: cyberleninka.ru/article/n/metody-otobrazheniya-ontologiy-obzor.

Дата поступления: 6.06.2025

Дата публикации: 25.07.2025