Применение универсальной модели данных в теоретическом материаловедении для хранения кристаллохимической информации

Д.Е. Яблоков[1]

[1]Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

Аннотация. Специалистам-исследователям в области теоретической кристаллохимии необходимо получать и обрабатывать достоверную и полную информацию о химических объектах различной природы и их исследуемых или прогнозируемых свойствах. Одной из основных проблем, возникающих при работе с неструктурированными или слабоструктурированными данными, является проблема потери соответствия между источником данных и приложениями, с которыми он взаимодействует. Чаще всего это выражается в наличии двух разных представлений данных на уровне системы хранения и программного слоя доступа, требующих взаимной трансляции. Рассматриваемый в статье уровень абстракции позволяет избежать подобного несоответствия и хранить в базе данных, поддерживающей универсальную модель, информацию любого типа и любого уровня сложности. Представленные в работе элементарные примитивы для описания объектов и взаимоотношений между ними формируют концептуальную метамодель. Они создают базовый каркас понятий общий для системы хранения данных и приложений, которые с ней работают. Например, понятийный аппарат в материаловедении тесно связан с определениями из теории графов посредством которых можно описывать кристаллохимические данные. Эта взаимосвязь позволяет, применяя взаимодополняющие понятия из области химии и дискретной математики, описывать как простые элементы кристаллохимических данных, такие как атомы и их связи, так и более сложные конструкции с возможностью проведения их последующей декомпозиции или классификации. Все это предоставляет возможность для использования различных методологических подходов при обработке и хранении данных в процессе исследований.

1. Введение

Для нормального функционирования любой информационной системы необходимо, чтобы ее модель данных адекватно отражала реалии той предметной области, для которой она разрабатывается. Например, в базе данных, созданной по принципу расширенного контекстуального паттерна [1], важно определить общие для большинства потребителей данных примитивы, основанные на априори определенных элементарных понятиях. Это позволит однозначно идентифицировать логически связанные с этими понятиями неструктурированные данные, в зависимости от контекста предметной области, а также обеспечит переносимость модели данных из проекта в проект. При этом отпадает

необходимость каждый раз модифицировать старую или разрабатывать новую схему хранения и проводить кардинальные изменения программ, которые взаимодействуют с базой данных. Степень детализации при использовании такого подхода обязывает разработчиков новых приложений продумывать на ранних этапах проектирования лишь наиболее важные вопросы. Все необходимые нюансы развития системы хранения могут быть учтены позже, когда концептуальное преставление о модели данных, по мере углубления знаний, касающихся проблемных участков целевой области знаний, разовьется до необходимого уровня.

Описание данных производится с использованием реляционного подхода и элементов из теории объектно-ориентированного программирования [2]. Основной акцент делается на том, что при сохранении реляционного ядра системы хранения она наращивается наиболее удачными объектными надстройками. В качестве таких надстроек могут выступать и расширяемая пользователем система типов, и средства описания иерархически взаимосвязанных данных, такие как наследование и композиция [3], которые позволяют представлять отношения между сущностями по принципам «подобного поведения» или «является частью» соответственно. Объектно-ориентированный подход делает возможным представление данных в виде совокупности взаимодействующих объектов, каждый из которых является экземпляром сущности определенного класса. Это способствует правильному и более эффективному структурированию хранимой информации, а также создает предпосылки для проведения объектно-ориентированной декомпозиции при формировании концептуальных границ модели данных [4].

2. Кристаллохимические данные

Многие вычислительные приложения для экспериментальных исследований часто используют некоторый набор элементов взаимосвязанных между собой определенным набором соединений. Например, экземпляр какой-либо абстрактной структуры данных может содержать в своем описании набор сущностей, обладающих семантикой поведения вершин графа. Эти сущности могут объединяться в пары с помощью связей, обладающих семантикой поведения ребра графа. Предположим, что для проведения расчетов или обработки результатов эксперимента в базу данных, поддерживающую универсальную модель хранения [5], нужно загрузить небольшой сегмент кристаллографических данных, касающихся представления информации о химическом соединении и его свойствах. И пусть эти данные будут касаться содержимого элементарной ячейки [6, 7], характеризующей структуру этого соединения и соответствующего этому содержимому некоторого набора свойств. Поскольку главной задачей объектно-ориентированного проектирования является правильный выбор совокупности используемых абстракций, то для выделения концептуальных границ модели данных необходимо сформировать перечень таких абстракций, связанных с уже определенным набором понятий. Данные об атомах и межатомных связях будут представлены в терминах объектов и отношений между объектами, а свойства атомов и связей – атрибутов объектов и атрибутов отношений соответственно. В виду того, что концептуально представление информации об атомах и межатомных связях очень похоже на представление информации о вершинах и ребрах графа, то в качестве основной идиомы хранения данных о содержимом элементарной ячейки целесообразно придерживаться выбора абстракции поддерживающей логику работы списка смежности графа [8]. В рамках этой абстракции можно легко отслеживать все дочерние элементы, включенные в связный список вершин обозначенных как корневые (рисунок 1). Для задач кристаллохимии, при ассоциации вершин и ребер с соответствующими объектами предметной области и связями между ними, свойства таких объектов и их связей можно легко отображать как свойства корневых или дочерних элементов и, соответственно, как свойства связей между этими элементами.

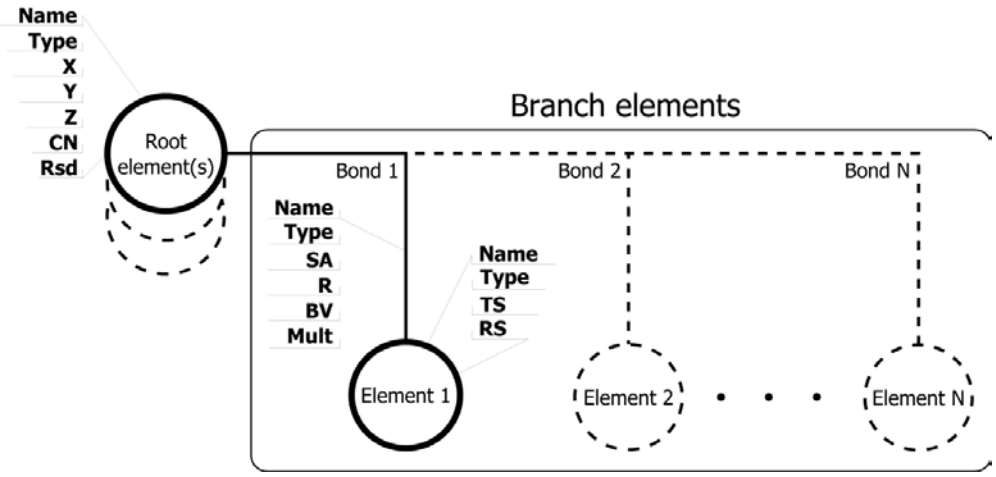

Рисунок 1. Атомы и связи между ними в виде списка смежности.

Например, все неэквивалентные атомы в элементарной ячейке, выступающие в роли корневых элементов («Root element(s)»), могут обладать некоторыми из следующих характеристик:

1. *Name* – символ химического элемента, который представляет атом;
2. *Type* – метрика выделяющая ключевые характеристики объекта для однозначной идентификации среди всех объектов других видов;
3. *X, Y, Z* – кристаллографические координаты;
4. *CN* – координационные числа для валентных и невалентных контактов;
5. *Rsd* – радиус сферического домена.

Все атомы, связанные с «корневыми» атомами, являющиеся дочерними элементами («Element 1», «Element 2», … , «Element N»), могут содержать в своем описании следующие данные:

1. *Name* и *Type* как и в описании корневых элементов;
2. *TS* – операции трансляционной симметрии;
3. *RS* – операции ротационной симметрии.

Все межатомные связи, интерпретируемые как связи между корневыми и дочерними элементами («Bond 1», «Bond 2», … , «Bond N»), могут быть описаны с использованием следующих свойств:

1. *SA* – телесный угол;
2. *R* – межатомное расстояние;
3. *BV* – валентная связь;
4. *Mult* – количество валентных связей одного типа в элементарной ячейке.

3. Способы представления данных

При работе с универсальной моделью данных [5] очень важен правильный подход к идентификации абстрактных сущностей. Это одна из самых сложных задач объектно-ориентированного анализа и проектирования [2] и в большинстве случаев ее решение фрагментарно содержит в себе элементы эвристики. Для этого необходимо уметь распознавать основные абстракции и механизмы, образующие терминологический аппарат или словарь предметной области, а также конструировать обобщенные абстракции и новые механизмы, определяющие способы взаимодействия для уже имеющихся объектов. Но, кроме того, для загрузки или извлечения данных необходимы запросы, к реляционной СУБД, обеспечивающие операции с данными объектов. Требуется загружать, сохранять, создавать новые данные, делать выборки по определённым критериям, удалять объекты. Все эти действия в конечном итоге сводятся к SQL запросам. Во время построения запроса нужно «знать всё» об отображаемых в строки таблиц объектах: идентификатор, класс сущности, имена и типы

атрибутов, средства для их чтения и записи, связи с другими объектами. Всю эту информацию предоставляет метамодель объектных данных.

Метамодель – это ключевое звено в процессе автоматизации хранения данных с учетом объектно-ориентированного способа описания информации. Обычно устанавливаются некоторые правила отображения метамодели на структуру реляционных таблиц. Используются фиксированные способы именования элементов модели, сопоставление конкретных типов данных описываемой предметной области и типов данных СУБД. Если же такие средства отсутствуют, то программистам, разрабатывающим уровень хранения, нужно самостоятельно сформулировать и ввести в эксплуатацию требуемые абстракции.

3.1. Объекты

Любой объект, как экземпляр сущности определенного класса, рассматривается как чистая абстракция, без привязки к какой-либо предметной области (рисунок 2). Спецификация свойств объектов, согласно используемым объектно-ориентированным надстройкам для реляционной модели хранения, производится на уровне типа объекта, (сущность «object_type»). Семантика хранения предполагает, что каждый тип объекта наследует какому-либо базовому типу («meta_type»), выраженному в терминах элементарных примитивов, определяющих смысловой контекст как признак для возможной классификации всех дочерних элементов данных.

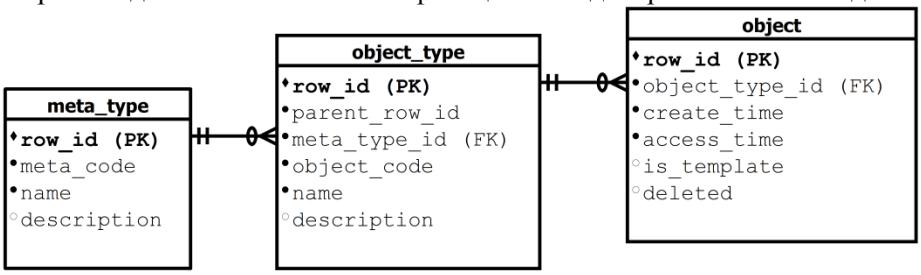

Рисунок 2. Универсальная модель данных. Объекты.

В качестве таких элементарных примитивов, используемых при описании мета-типов можно выделить следующий набор абстракций:

- Справочник (Dictionary) – специализированная логическая область структурированных, заранее подготовленных и сохраненных данных, определяющих совокупность элементов информации относящихся к какой-либо предметной области. Во избежание многократного повторного ввода одних и тех же данных, а также исключения возможных ошибок справочники, как каталоги различного назначения, содержащие справочную или служебную информацию, являются незаменимым инструментарием, группирующим эту информацию по определённым критериям. Включаемые в справочники данные используются при заполнении атрибутов-ссылок некоторых объектов, позволяя избежать ввода заведомо неверных сведений. Варианты типов объектов: справочник химических элементов (Periodic Table), справочник пространственных групп (Space Groups Table).
- Строка данных (Data Row) – абстракция, содержащая некоторый набор данных, доступ к которым может быть осуществлен через значения ассоциативно связанных с ее экземпляром атрибутов. Можно сказать, что строка данных выступает в роли объекта, обладающего семантикой строки абстрактной таблицы, в которой каждый столбец в точности соответствует атрибуту, ассоциированному со строкой данных. Некоторые из ассоциативно связанных типов объектов: запись справочника химических элементов (Periodic Table Record), запись справочника пространственных групп (Space Groups Table Record).
- Группа данных (Data Group) – логическая группа данных, объединенных по каким-либо критериям общности, обозначающим существенные особенности контекста предметной области и определяющим степень подобия для входящих в нее объектов с родственным поведением или состоянием. Возможные типы объектов: кольцо или цикл (Circle, Cycle), молекула (Molecule).

- Вершина (Vertex), ребро (Edge) – объекты любой природы, которые, как правило, имеют в своем описании какую-либо характеристику, позволяющую идентифицировать их среди множества подобных объектов. Кроме того, они могут быть снабжены и некоторыми дополнительными атрибутами, касающимися, например, положения или статуса вершины, веса или ориентированности ребра, а также сведениями о свойствах той абстракции, которая в данный момент может рассматриваться как вершина или ребро. Типы объектов с поведением подобным вершине или ребру: атом (Atom) и межатомная связь (Bond) [8].

3.2. Отношения

Каждому экземпляру отношения («relationship») ставится в соответствие экземпляр объекта, представляющего само отношение («object»), а также экземпляры объектов-потомков («descendant»), задавая, таким образом, смысловую принадлежность отношения и его дочерних элементов к конкретным типам объектов (рисунок 3). Например, тип связи («object_type») между атомами может быть обозначен как «Bond», а базовым типом («meta_type») для данного типа связи будет являться мета-тип «Edge» (ребро). Соответственно, объекты-потомки такой межатомной связи будут иметь тип «Atom» с базовым типом «Vertex» (вершина). Это дает гарантию, что при проведении анализа или декомпозиции этой межатомной связи ее можно будет рассматривать в терминах примитивных графовых абстракций. Кроме того, описание каждого отношения между объектами снабжено соответствующим типом отношения («relationship_type»), определяющим необходимый уровень абстракции, для выделения существенных характеристик, отличающих конкретный экземпляр отношения от отношений, принадлежащих другим классам. Например, тип отношения может содержать информацию, касающуюся взаимоотношений объектов по принципу «часть-целое», т.е. специфицировать форму отношения между объектами по таким уровням как осведомленность, агрегирование или композиция.

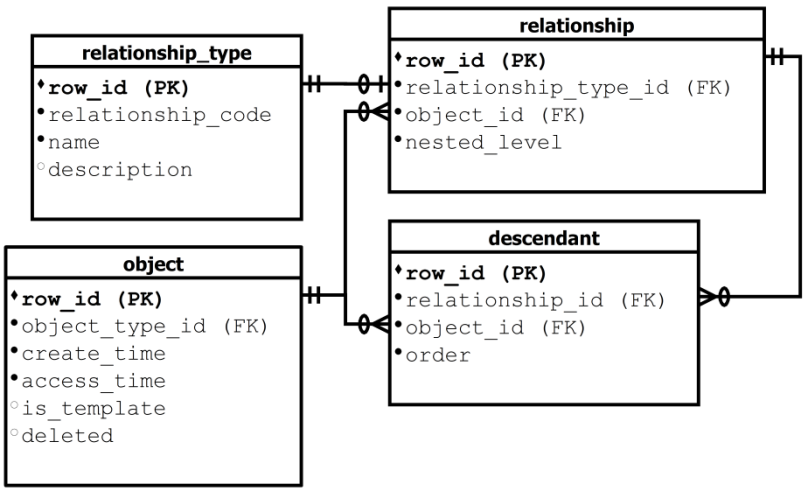

Рисунок 3. Универсальная модель данных. Отношения.

Основные типы отношений:

- Осведомленность (Acquaintance) – это отношение осведомленности одного объекта о другом, означающее некоторую семантическую связь между ними, выраженную в каком-то виде, который будет уточнен в будущем. Такое отношение может быть использовано на ранних стадиях работы с данными, чтобы просто показать, что взаимосвязь между объектами существует [2, 3].
- Агрегирование (Aggregation) – это отношение типа «часть-целое», при котором агрегирующий объект содержит в своем описании агрегируемый объект, который может рассматриваться как его часть. Это отношение является слабой формой отношения

включения, в котором сроки жизни целого и его части не зависят друг от друга. Агрегирование – это ослабленное отношение композиции [2, 3].

- Композиция (Composition) – это отношение типа «часть-целое», при котором объекты объединяются для получения более сложного поведения. Объект-владелец содержит в своем описании подобъект, который может рассматриваться как его часть. Это отношение является сильной формой отношения включения, когда время жизни объекта, являющегося частью, зависит от времени жизни его владельца [2, 3].

3.3. Атрибуты

Каждый атрибут («attribute») содержит в своем описании информацию о коде (поле «attribute_code»), задающим семантику его хранения. Это могут быть как примитивные атрибуты, соответствующие определенным типам данных, так и составные, состоящие из примитивных или таких же составных атрибутов, либо атрибуты-ссылки на атрибуты других объектов (рисунок 4). Ассоциативная сущность «attribute_data_type» кроме ссылки на соответствующий тип данных (поле «data_type_id»), определяющего, совместно с типом атрибута, сигнатуру примитивного атрибута, содержит также поле «value_exid», значения которого используются для конструирования псевдонимов таблиц, отвечающих за хранение значений примитивных атрибутов.

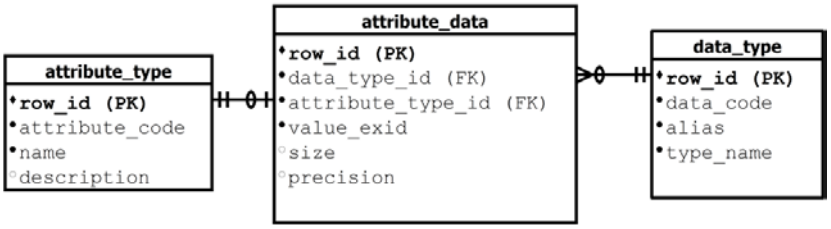

Рисунок 4. Универсальная модель данных. Атрибуты.

Типы атрибутов:

- Типизированный атрибут (Typed Attribute) – тип для примитивных атрибутов, связанных с соответствующим типом данных.
- Составной атрибут (Composite Attribute) – тип для иерархически подчиненного описания атрибутов, корнем которого является родительский атрибут, обращение к которому производится в терминах обычных несоставных атрибутов. Доступ к значениям дочерних атрибутов осуществляется с помощью специализированных функций подсистемы хранения.
- Ссылка на атрибут (Attribute Link) – тип для атрибута, который поддерживает механизм ссылок на атрибуты объектов и позволяет определять относительное значение для другого атрибута объекта или атрибута отношения между объектами.

4. Результаты

Финальная выборка данных (рисунок 5) для топологического типа [6], с полным списком атомов, межатомных связей и их свойствами, содержащимися в элементарной ячейке, представлена ниже.

	atom	level	el	x	y	z	cn	rsd	rs	ts	bond	sa	r	sseg	bv	mult
1	Root Atom	1	C	0.125	0.125	0.125	4	1.106	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	Branch Atom 1	2	C	NULL	NULL	NULL	NULL	NULL	-y,-x,-z	-1,-1,-1	HB1	22.0	1.545	4.1	V	16
3	Branch Atom 2	2	C	NULL	NULL	NULL	NULL	NULL	1/4+y,1/4+x,-z	0,0,-1	HB2	22.0	1.545	4.1	V	16
4	Branch Atom 3	2	C	NULL	NULL	NULL	NULL	NULL	1/4+y,-x,1/4+z	0,-1,0	HB3	22.0	1.545	4.1	V	16
5	Branch Atom 4	2	C	NULL	NULL	NULL	NULL	NULL	-y,14+x,1/4+z	-1,0,0	HB4	22.0	1.545	4.1	V	16

Рисунок 5. Результат выборки данных для топологического типа.

Поле «level» определяет уровень вложенности между атомами по принципу «часть целое». В этом смысле при представлении данных в виде древовидной структуры, следуя логике хранения списка смежности, строка содержащая большее значение для поля «level» является

подчиненной относительно строки с меньшим значением того же поля. Поэтому в результирующей выборке для верхней строки, содержащей информацию о свойствах корневого элемента, значение уровня вложенности равно 1, а для строк, относящихся к дочерним элементам и их связям с корнем списка, уровень вложенности равен 2.

При создании систем хранения на основе универсальной модели данных наиболее подходящей основой является дедуктивный метод. Он обеспечивает декомпозицию сложных понятий на более простые компоненты с математически и семантически обоснованным поведением. Использование представленных в статье примитивов служит важной предпосылкой разработки эффективного и надежного способа описания данных, необходимого при проведении исследований.

5. Выводы

Введенный в понятийный аппарат системы хранения набор объектных типов («object_type») позволяет формализовать стратегию выборки данных и однозначно идентифицировать экземпляр любого объекта и ассоциированные с ним атрибуты и соответствующие им значения. Мета-типы («meta_type») представляют собой абстракции, позволяющие сконцентрироваться на базовых особенностях производных типов объектов так, что во время работы с данными можно игнорировать излишние характеристики. Это снижает уровень сложности и позволяет отбросить нерелевантные детали не свойственные типам объектов при их рассмотрении в терминах мета-типов.

Механизм типизации отношений («relationship_type») определяет стратегию манипулирования данными, имеющими более сложную структуру, организованную по принципу «часть-целое». Это позволяет упростить разработку прикладных программ и формализовать поведение иерархически взаимосвязанных объектов, объединенных отношениями осведомленности, агрегирования или композиции.

Сложная и избыточная на первый взгляд типизация атрибутов определяет точную характеристику структуры и поведения свойств объектов. Совокупность иерархически подчиненных типов атрибутов («attribute_type») и отображений («data_type»), сопоставляющих типы данных предметной области и типы данных СУБД, задают набор правил для доступа к значениям атрибутов. Возможность использовать составные атрибуты позволяет создавать семантические зависимости между подчиненными свойствами объектов и четко определять и регламентировать их допустимые сочетания.

6. Заключение

Работа с кристаллохимическими данными при использовании описанного в статье уровня абстракции подразумевает легкость в понимании структуры хранения универсальной модели [5] и позволяет достаточно просто моделировать предметную область, что предполагает точные формальные определения, которые интуитивно понятны. Важно отметить, что при работе с универсальной моделью становится возможным ввод информации, структура которой не определена заранее, а изменение структурных связей типа «сущность-атрибут», «сущность-сущность» или «отношение-атрибут» может производиться в режиме runtime. В определенном смысле использование такого подхода в совокупности с соответствующими правилами, определяющими способ работы с базой данных, имеет неоспоримое преимущество так как при правильно выбранной стратегии проектирования обеспечивает присутствие только полной, непротиворечивой и адекватно отражающей предметную область информации.

7. Благодарности

Работа по созданию структуры универсальной модели данных выполнена при поддержке правительства Российской Федерации (грант 14.B25.31.0005).

8. Литература

[1] Яблоков, Д. Паттерны проектирования моделей баз данных как систем хранения экспериментальной информации при решении исследовательских задач / Д. Яблоков //

Материалы Международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ-2017). – Самара, 2017. – С. 1798-1806.
[2] Booch, G. Object-Oriented Analysis and Design with Applications. Third Edition / G. Booch. – Addison-Weatley, 2007. – 534 p.
[3] Гамма, Э. Приемы объектно–ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб., 2001. – 368 с.
[4] Fowler, M. Patterns of Enterprise Application Architecture / M. Fowler. – Addison-Weatley, 2003. – 736 p.
[5] Silverstone, L. The data Model Resource Book. Vol. 3: Universal Patterns for Data Modeling / L. Silverstone // Len Silverstone. – Wiley Computer Publishing, 2009. – 648 p.
[6] Blatov, V.A. Periodic-Graph Approaches in Crystal Structure Prediction / V.A. Blatov, D.M. Proserpio // Modern Methods of Cristal Structure Prediction. – Wiley-VCH, 2011. – P. 1-28.
[7] Hahn, T. International Tables for Crystallography Vol. A Space–group symmetry / T. Hahn. – Springer, 2005. – 911 p.
[8] Sedgewick, R. Algorithms, Fourth edition / R. Sedgewick, K. Wayne. – Addison-Weatley, 2011. – 976 p.

Using universal data model in materials science for storing crystal-chemical information

D.E. Yablokov[1]

[1]Samara National Research University, Moskovskoye shosse, 34, Samara, Russia, 443086

Abstract. Specialists in a theoretical crystal chemistry need to obtain and process relevant and complete information about the objects of the different nature and their investigated or predicted properties. The main problem of unstructured data is the mismatch between data source and applications, which with it interacts. Most often, there are two different data representations of the database level and the data access layer requiring the special translation. The abstraction layer considered in the article allows avoiding such mismatch. It allows to store in the database supporting the universal data model, information of any type and complexity. The elementary primitives provided in article for the description of objects and relationships create a conceptual meta-model. They build the basic framework of concepts common to the database and applications that work with it. For example, the conceptual framework in materials science is connected to definitions of graph theory that allows one to describe crystal-chemical data using graph abstractions. With interrelated concepts from chemistry and discrete mathematics, we can describe the basic objects, such as atom or bond as well as more difficult objects. This provides the ability to use different methodological approaches in the processing and storage of data during research process.

Keywords: Universal data model, Crystal-chemical information, Materials science, Object-oriented approach, Relational database, Elementary primitive. Concepts refinement.