УНИФИЦИРОВАННЫЙ ДОСТУП К УНИВЕРСАЛЬНОМУ ХРАНИЛИЩУ БОЛЬШИХ ДАННЫХ

Д.Е. Яблоков

Самарский государственный университет

В статье рассматривается возможность применения обобщенных концепций, основанных на использовании нескольких парадигм программирования, на примере разработки библиотеки программных компонентов для организации унифицированного доступа к универсальному хранилищу больших данных.

Современное состояние индустрии программирования нельзя представить без качественных технологий построения абстракций, базовых принципов, дающих основу соответствующему стилю проектирования и средств, содержащихся в используемых парадигмах, которые определяют стратегию конструирования программного обеспечения. Обдуманное и правильное использование подобного рода механизмов и связанной с ними терминологии дают основу формированию каркаса понятий, определяющего логические рамки для описания объектов или их семейств, работа с которыми становится возможной в терминах используемого понятийного аппарата. Это приводит к необходимости формирования набора абстракций, определяющих семантическое и синтаксическое поведение объектов, а также к появлению семейства обобщенных концепций, содержащих набор требований и дающих представление о свойствах и ограничениях, которыми должна обладать абстракция, являющейся моделью одной или нескольких концепций.

Хорошо известно, что многие задачи, предполагающие работу с данными, как в смысле улучшения рабочих характеристик обработки или сохранения больших объемов сложной информации, так и в смысле повышения качества реализации решений, структурирующих логику предметной области, хорошо формализуются посредством соответствующих паттернов доступа, манипуляции и отображения данных. По сути, такие паттерны могут рассматриваться как широко распространенные и имеющие всем понятный смысл комбинации языковых средств и проектных решений, приводящих к появлению высокоуровневых конструкций, призванных снизить сложность программного обеспечения и синтезировать более качественные продукты. Но, важно подчеркнуть, что понятие паттерна включает не только различные сочетания языковых конструкций и соответствующих стилей проектирования, но и определенные ожидания относительно того, как эти сочетания будут себя вести, т.е. как они будут соответствовать набору абстрактных требований, определяющих интерфейс и семантическое поведение. Обобщенные концепции – это мощный инструмент решения задач связанных с данными, дающий возможность как определения поведения и отношения между дискретными объектами и их семействами, так и применяемый для описания синтаксических и семантических особенностей компонентов предназначенных для обработки и представления большого объема информации.

При разработке очень большой и очень сложной компьютерной программы связанной с большим объемом данных, нужно потратить значительные усилия на уяснение и определение задачи, на осознание ее сложности и разбиение ее на меньшие подзадачи, решение которых легко реализовать. Немаловажным фактором, при этом, является необходимость использования соответствующих парадигм и проверенных временем идиом, основанных на опыте и твердом понимании выбранного языка программирования и объединении воедино наилучших практических решений, рекомендаций и правил, определяющих стандартизированный способ получения на выходе наиболее качественного результата. Одним из примеров реализации такого стандартизированного набора рекомендаций и правил является, разрабатываемая в рамках проекта создания экспертной системы, обобщенная библиотека расширений (GLEX). Ее

теоретическую основу составляет набор базовых концепций и требований к типам данных стандартной библиотеки шаблонов (STL), широко используемой при программировании на языке C++. В дальнейшем планируется широкое применение GLEX при реализации компонентов обработки и представления данных большого объема. GLEX – это не просто библиотека, а скорее набор соглашений, объединяющих компоненты нескольких типов: адаптеры контейнеров, алгоритмы, концепции итераторов и функциональных адаптеров. Идея состоит в том, что адаптеры контейнеров и алгоритмы, работающие с ними, могут ничего не знать друг о друге. Это преимущество достигается за счет обобщенных концепций итераторов.

Обобщенные концепции итераторов (Рисунок 1).

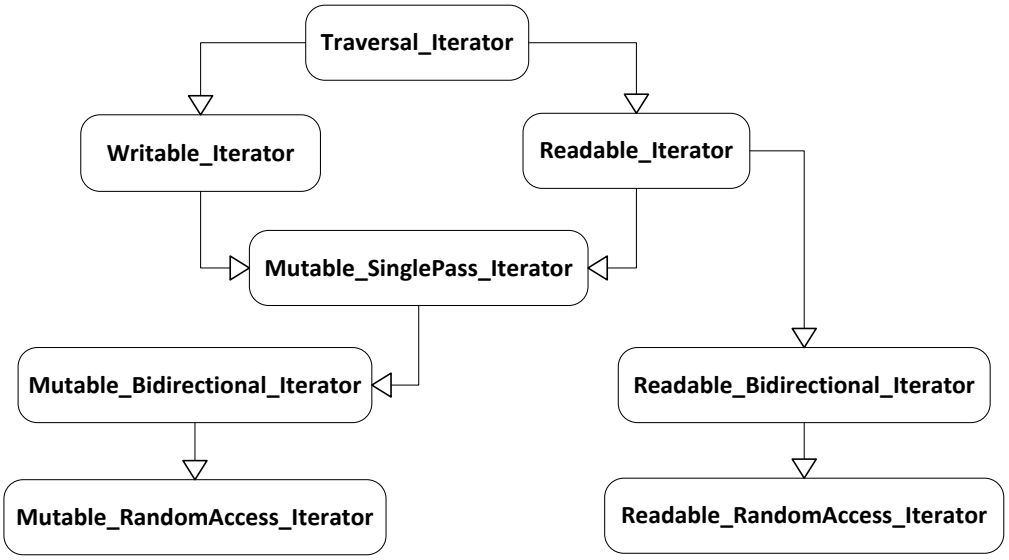

Рисунок 8 – Развитие концепций итераторов

Адаптеры обобщенных концепций итераторов (Рисунок 2 – 4).

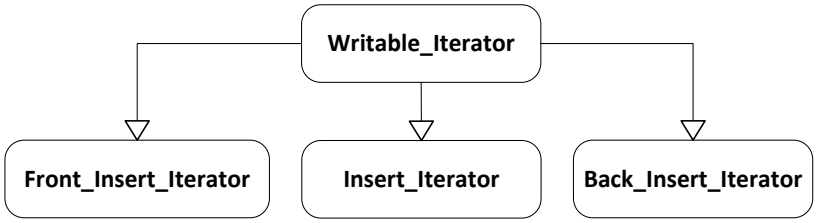

Рисунок 9 – Концепции итераторов вставки

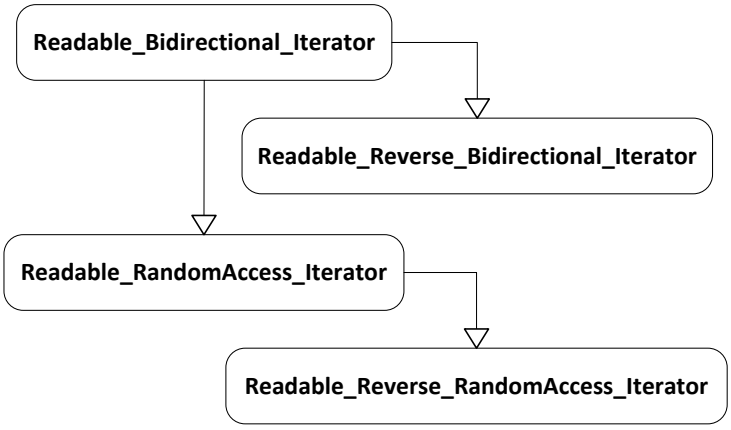

Рисунок 2 – Концепции реверсивных, неизменяющих итераторов

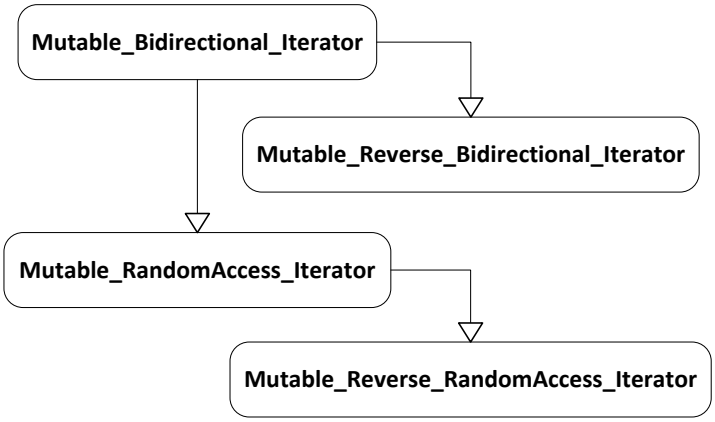

Рисунок 10 – Концепции реверсивных, изменяющих итераторов

Обобщенные концепции адаптеров контейнерных классов (Рисунок 5).

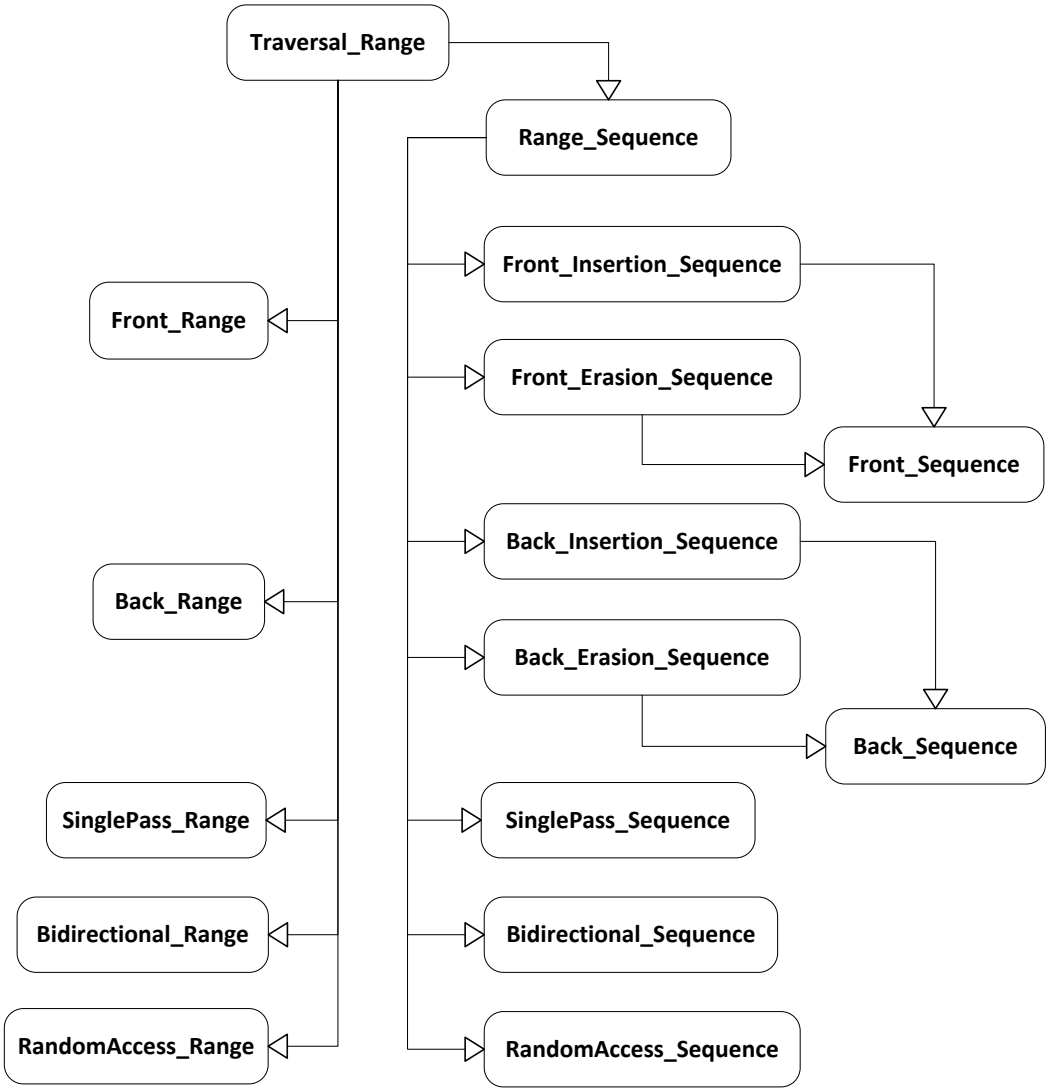

Рисунок 5 – Развитие концепций адаптеров контейнерных классов

Обобщенные концепции адаптеров диапазонов и последовательностей (Рисунок 6 – 7).

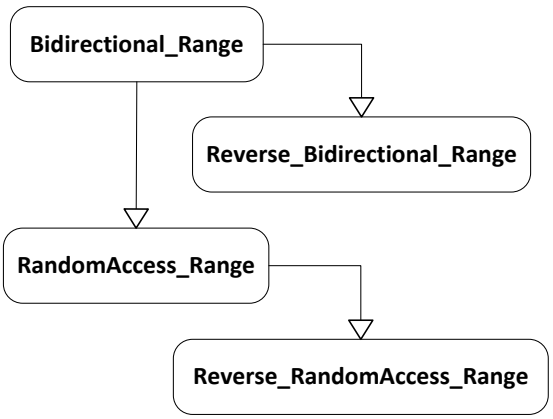

Рисунок 6 – Концепции реверсивных диапазонов

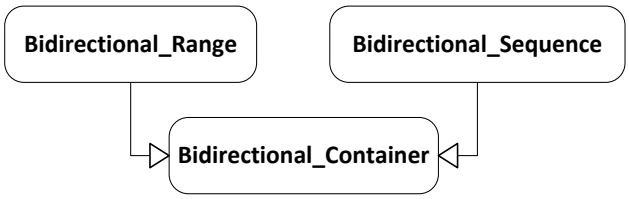

Рисунок 7 – Концепция двунаправленного контейнера

Для удобства организации данных было введено понятие секции, т.е. абстрактной таблицы, содержащей определенный набор обязательных полей и являющейся основным компонентом системы при работе с данными. Логически эту абстракцию можно описать как группу данных, объединенных по общим признакам (полям или атрибутам) либо каким-то иным критериям общности. Каждая из секций имеет свой уникальный идентификатор или дескриптор. Можно выделить три структурных вида типов секций (Рисунок 8):

1. Плоский список (Flat_List) – набор значений атрибутов конкретной сущности.
2. Иерархия строк (Row_Hierarchy) – секция, строки которой ссылаются друг на друга, образуя при этом древовидную иерархическую структуру. Каждая строка может ссылаться на несколько дочерних строк, что дает возможность моделировать иерархии любой сложности и любого уровня вложенности.
3. Подчиненная секция (Sub_Section), т.е. секция, строки которой являются дочерними по отношению к строке или строкам другой родительской секции (Super_Section). Организованные подобным образом данные могут располагаться в таблицах любых структурных видов. Иными словами и подчиненная и родительская секции могут иметь любой структурный вид. Они могут быть как плоским списком, так и иерархией строк.

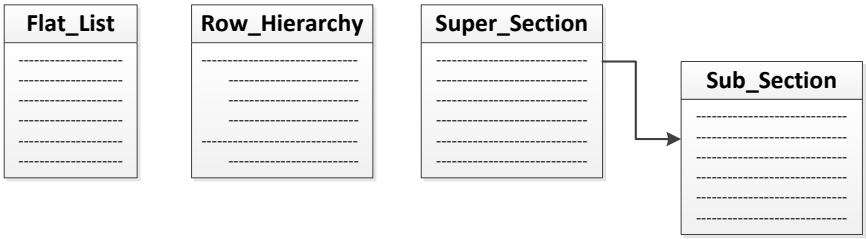

Рисунок 8 – Концепции структурных видов типов секций

На результат процесса программирования всегда влияли ограничения связанные либо с возможностями компьютеров, либо с возможностями человека. Несколько десятилетий назад, главным ограничивающим фактором была низкая производительная

способность компьютера. В настоящее время, физические и аппаратные ограничения отошли на второй план. С все более глубоким проникновением компьютеров во все сферы научной деятельности, программные системы становятся все более простыми для использования специалистами, но сложными по внутренней архитектуре. Считается, что наиболее действенным способом борьбы со сложностью программных продуктов является попытка построения соответствующего уровня абстракции, основанного на анализе общности и изменчивости объектов предметной области. Например, возможность использования иерархии полиморфных типов данных, связанных отношением наследования, а так же наборов абстрактных требований к типам, моделирующих интерфейс и семантическое поведение, даст неоспоримое преимущество при конструировании компонентов и подсистем работы с большими наборами данных (Рисунок 9).

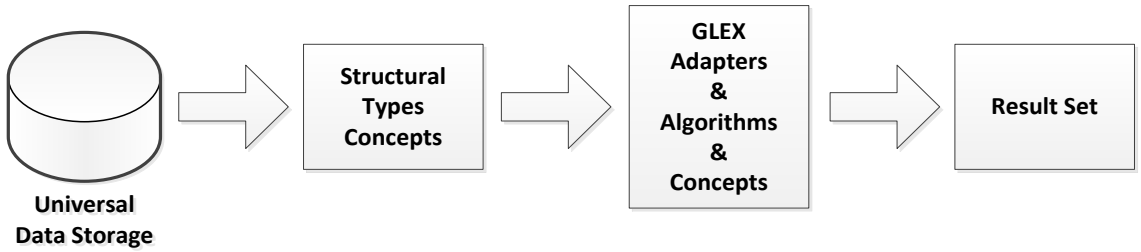

Рисунок 11 – Извлечение больших наборов данных из универсального хранилища

Литература

1. Devon M. Simmonds, The Programming Paradigm Evolution, IEEE Computer, June 2012, IEEE Computer Society.

2. Блинов И.Н., Романчик В.С. Java. Промышленное программирование. – Минск.: УниверсалПресс, 2007. – 704с.

3. Вандевурд Д., Джосаттис Н. М. Шаблоны С++: справочник разработчика. – М.: Вильямс, 2003. – 544 с.

4. Страуструп Б. Язык программирования С++. Специальное издание. – М.: Издательство Бином, 2011 г. – 1136 с.

5. Яблоков Д.Е. Применение парадигмы обобщенного программирования в объектно-ориентированных языках. Информатика, моделирование, автоматизация проектирования: сборник рекомендованных научных трудов / под ред. Н. Н. Войта. – Ульяновск : УлГТУ, 2013. – с. 113-118.

6. Яблоков Д.Е. Парадигмы программирования. XIV МНПК «Научное обозрение физико-технических наук в XXI веке», Москва, Prospero №2(14), 2015.- С. 94-98.

7. Яблоков Д.Е. Использование обобщенных концепций в объектно-ориентированных языках программирования. МНТК «Перспективные информационные технологии» Сборник научных трудов / под ред. С.А. Прохорова. - Самара, СГАУ, 2015.- С. 341-345.