DOI 10.37882/2223–2966.2020.12.10

ИНТЕГРАЦИЯ ИНФОРМАЦИОННЫХ РЕСУРСОВ СИТУАЦИОННЫХ ЦЕНТРОВ

Дриленко Максим Владимирович
Аспирант, Кубанский государственный технологический университет
mvdrilenko@gmail.com
Симанков Владимир Сергеевич
Д.т.н., профессор, Кубанский государственный технологический университет
vs@simankov.ru

Аннотация. Постановка задачи: существуют отдельные подходы к формированию единого информационного пространства между различными информационными ресурсами, зачастую хранилища данных представлены в виде сформированных вручную уникальных алгоритмов преобразования таблиц из различных источников, что вызывает потребность в наращивании организационных ресурсов при появлении новых значительных объемов неструктурированной информации, что приводит к экономическим и операционным потерям, что невозможно в условиях функционирования интеллектуального ситуационного центра. Целью работы является формирование методики объединения информационных ресурсов из различных источников для решения задач интеллектуального ситуационного центра, что необходимо и важно при обработке больших объемов неструктурированной информации. Используемые методы: методы системного анализа, методы объектно-ориентированного анализа и проектирования.

Результат: предложена методика формирования физических моделей данных из разнородной неструктурированной и слабоструктурированной информации. Эта методика совместима с четырьмя типами СУБД NoSQL: колонками, документами, графиками и ключевым значением. Модели данных (концептуальные, логические и физические), используемые в разработанном процессе, соответствуют мета-моделям: от концептуального к логическому, а затем от логического к физическому.

Практическая значимость: представленное решение предлагается реализовать в виде аппаратно-программного комплекса, использующего представленную методику для интеграции информационных потоков различных ситуационных центров. Реализация позволит обеспечивать адаптивное динамическое преобразование поступающих данных и их дальнейшее использование в рамках ситуационного центра.

Ключевые слова: интеграция данных, интеллектуальные ситуационные центры, обработка данных, анализ данных, объединение ситуационных центров, объединение информационных потоков, обработка данных.

INTEGRATION OF INFORMATION RESOURCES IN SITUATION CENTERS

M. Drilenko
V. Simankov

Summary. Statement of the problem: there are separate approaches to the formation of a single information space between various information resources, often data warehouses are presented in the form of manually generated unique algorithms for converting tables from various sources, which causes the need to increase organizational resources when new significant volumes of unstructured information appear, which leads to economic and operational losses, which is impossible in the conditions of the functioning of an intelligent situational center. The aim of the work is to form a methodology for combining information resources from various sources to solve the problems of an intelligent situational center, which is necessary and important when processing large volumes of unstructured information. Methods used: methods of system analysis, methods of object-oriented analysis and design. Result: a technique for the formation of physical data models from heterogeneous unstructured and semi-structured information is proposed. This technique is compatible with four types of NoSQL DBMSs: Columns, Documents, Graphs, and Key Value. The data models (conceptual, logical and physical) used in the developed process correspond to meta-models: from conceptual to logical, and then from logical to physical. Practical relevance: the presented solution is proposed to be implemented in the form of a hardware and software complex using the presented methodology for integrating information flows of various situational centers. The implementation will provide an adaptive dynamic transformation of incoming data and their further use within the situational center.

Keywords: data integration, intelligent situational centers, data processing, data analysis, situation center consolidation, information flow consolidation, data processing.

Введение

В настоящее время существуют отдельные подходы к формированию единого информационного пространства для доступа из различных информационных ресурсов, которые представлены в виде сформированных вручную уникальных алгоритмов преобразования таблиц из различных источников. Такой подход вызывает потребность в наращивании организационных ресурсов при появлении новых значительных объемов неструктурированной информации, что приводит к экономическим и операционным потерям, что невозможно в условиях функционирования интеллектуального ситуационного центра.

Таким образом, существует необходимость совершенствования методических положений объединения информационных ресурсов из различных источников для решения различных прикладных задач.

Исследование зарубежной литературы [3,4,5,6] по данной тематике показывает, что научные изыскания направлены на автоматическое приведения данных к единой структуре. Однако, разнообразие имеющихся типов данных не позволяет сформировать единый подход для обработки получаемой информации (Рисунок 1).

Целью данной работы является формирование методики объединения информационных ресурсов из различных источников для решения задач интеллектуального ситуационного центра, что необходимо и важно при обработке больших объемов неструктурированной информации.

Методология

Для достижения поставленной цели в работе требуется осуществить исследование неструктурированных информационных потоков и их структуры, выделить особенности и модели таких данных, изучить возможности преобразования в различные формы представления.

Модель данных — это схема описания структуры данных для конечного потребителя (приложения, базы данных). Модель содержит типы и структуры, совокупность операций, накладываемые на типы ограничения [1].

Структурированные данные имеют определенные ограничения для каждого атрибута, которые усложняют модификацию модели в соответствии с новыми требованиями. Структура таких данных определена с помощью схем данных, автоматическое преобразование затруднительно [2].

Слабоструктурированные данные имеют неполную структуру, имеют исключения, значения скалярных полей зачастую представлены в виде текстовой информации. Дополнительно возникает проблема определения принадлежности данных, требуется дополнительная верификация идентифицированного документа. Неструктурированные данные представлены полностью отсутствующей структурой и ограничениями применимых операций с ними. Автоматическое изменение структуры таких данных не может быть выполнено.

Представление накопленной информации в преломлении к каждому типу данных (структурированных, полуструктурированных, неструктурированных) показано на рисунке 2 [7].

В слабоструктурированных данных атрибуты могут быть сформированы в виде текста, следовательно необходим надежный механизм проверки сопоставления данных конкретному атрибуту. Схема может не в полной мере отвечать обрабатываемой информации [4,8]. Работать с документом, не имея представлений о его структуре затруднительно, возникает задача определения схемы обрабатываемых массивов информации, их распознавания в процессе использования модели для получения новой информации. Дополнительно атрибуты могут не существовать, или не удовлетворять условиям корректности данных, заданным для этих атрибутов. Таким образом, в формируемой модели должны использоваться инструменты обработки исключений, который позволят установить структуру запроса к таким данным, используя заданные критерии.

Для перехода к единому информационному пространству необходимо использовать общую модель данных универсального хранилища [9,14], которая формируется последовательно и состоит из концептуальной, логической и физической модели данных. Переход между моделями осуществляется последовательно.

Концептуальная модель универсального хранилища данных рассматривается как описание основных объектов и связей между ними [10]. Концептуальная модель отражает предметную область, в рамках планируемого универсального хранилища данных [11,12].

Логическая модель расширяет концептуальную путем определения для сущностей их атрибутов, описаний и ограничений, уточняет состав сущностей и взаимосвязи между ними.

Физическая модель данных описывает реализацию объектов логической модели на уровне объектов конкретной базы данных, на ней строится взаимодействие подсистем виртуального уровня и уровня приложений (Рисунок 3).

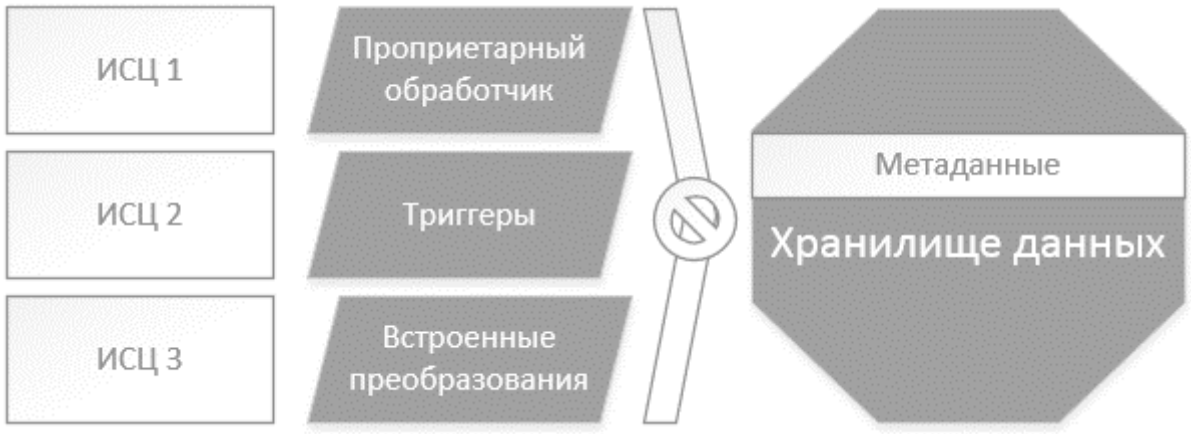

Рис. 1. Представление данных

Рис. 2. Схема данных

Для работы с неструктурированной или слабоструктурированной информацией требуется сформировать информационное пространство [13,14] для определения конечного вида представления данных, который имеет необходимый функционал и является удобным для использования данных.

Поскольку для формирования ассоциаций между объектами и характеристиками необходимо работать с разными источниками данных, в которых один и тот же самый объект может быть представлен под разными названиями, то для сравнения схем источников данных целесообразно использовать пространство данных с каталогом данных и словарем данных для сравнения названий объектов.

Каждый участник пространства данных поддерживает модель данных и соответствующий язык запросов, соответствующий формируемой модели. Запрос к такому программному средству поддерживается в файловых системах относительно директорий: сопоставление имен, поиск в диапазоне дат, сортировка за размером файла и др. На следующем уровне пространства данных модель данных должна поддерживать мультимножение слов с целью осуществления эффективного поиска необходимой информации за ключевыми словами. Ниже уровня модели мультимножение слов в иерархии может располагаться модель слабоструктурированных данных, основанная на обозначенных графах. Поскольку источники данных разнотипные, то необходимо определить платформу и архитектуру хранилища данных.

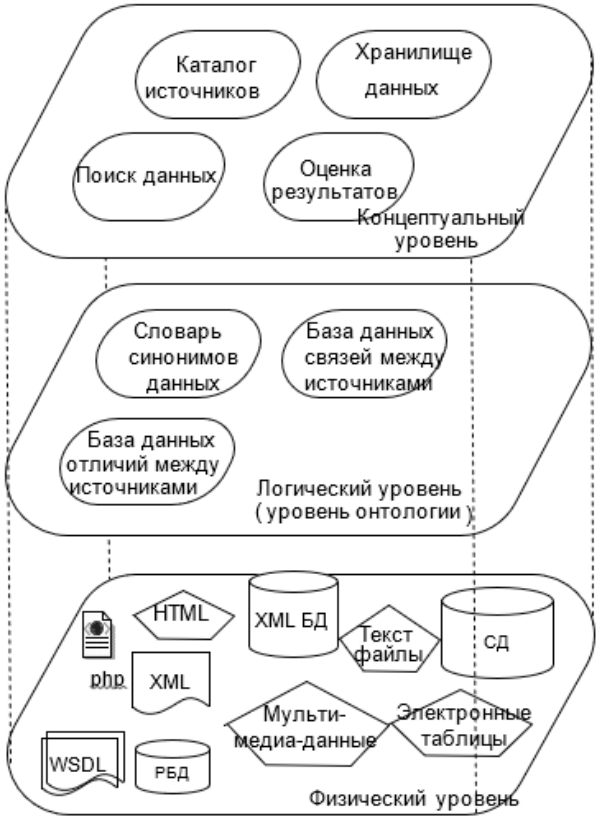

Рис. 3. Уровень реализации физической модели пространства данных

Платформа поддержания хранилища данных — это набор программного обеспечения хранения и поиска данных в информационном пространстве [15].

Архитектура пространства данных спроектирована уровнями (Рисунок 3). Уровень приложений предназначен для реализации операций над данными в пространстве данных. Уровень онтологий используется для установления связи между источниками.

Последний уровень содержит источники данных и обеспечивает доступ к данным и выполнению операций уровня применений непосредственно в источнике (например, операция выборки на уровне реализации выполняется как запрос в конкретной базе данных).

Для обеспечения последовательного перехода от объекта данных (неструктурированных данных) к физической модели необходимо создать алгоритм перехода от объекта к конкретному представлению данных, что можно реализовать в виде модуля, который отвечает за преобразование в физическую модель NoSQL. В соответствии с обоснованием введен логический промежуточный уровень между концептуальным и физическим уровнями. Этот уровень направлен на техническое описание структуры данных без указания характеристик, характерных для каждой СУБД. Другими словами, модуль Object-to-NoSQL работает в два последовательных шага: концептуальный > логический, а затем логический > физический. Переход от одной модели к другой осуществляется с использованием преобразований типа M2M, формализованных в QVT.

Элемент модуля Object-to-NoSQL представляет собой диаграмму классов. Пользователь предоставляет входную группу данных, конкретизируя концептуальную мета-модель PIM. Данная мета-модель показывает основные элементы, составляющие модель данных, а также их структурные характеристики.

Окончательный результат, возвращаемый модулем Object-to-NoSQL, представляет собой физическую модель NoSQL (колонки, документы, графики или значение ключа), которая включает в себя:

- Модель данных, содержащая необходимые элементы для реализации базы данных NoSQL;
- Набор руководящих принципов, определяющих условия использования атрибутов и реализации отношений в соответствии с приемами, присущими выбранной СУБД NoSQL.

Очевидно, что для заданного вывода (физической модели NoSQL) необходимо сохранить его параметры,

т.е. его мета-модель и правила преобразования, которые позволяют его генерировать. Для иллюстрации работы выбрано производство физических моделей с очень четкими характеристиками, используя СУБД Cassandra, SSDB, Neo4j и Redis. Если пользователь хочет использовать другую СУБД, модуль необходимо дополнить новыми параметрами, специфичными для этой системы.

Модуль Object-to-NoSQL состоит из двух преобразований: Object-to-GenericModel и GenericModel-to-PhysicalModel. На первом этапе входная DCL трансформируется в общую NoSQL модель, соответствующую логической PIM-модели. На втором этапе в качестве входной информации принимается общая модель и генерируются необходимые элементы для реализации БД, а также набор руководящих принципов поддержки, специфичных для выбранной СУБД NoSQL. Эти два преобразования выполняются набором правил M2M, формализованных в QVT, таким образом два преобразования бесшовно связаны между собой.

Стоит обратить внимание, что модуль преобразования DCL способен преобразовывать DCL в физическую модель [13] для одной из платформ реализации NoSQL: столбцов, документов, графиков и ключевых значений. Как уже упоминалось выше, в этой работе наш рассмотрены СУБД NoSQL каждого типа: Cassandra для колонок, SSDB для документов, Neo4j для графиков и Redis для ключей/значений. В данной работе рассмотрено, что решение совместимо и с другими СУБД NoSQL, такими как HBase (ориентированная на столбцы) и CouchDB (ориентированная на документы).

Рассмотрим этапы реализации модуля преобразования Object-to-NoSQL. Эта реализация требует предварительного определения набора мета-моделей и правил преобразования M2M-типа.

Сначала необходимо создать мета-модели ECORE. Это мета-модели концептуального MIP, логического MIP для Cassandra, SSDB, Neo4j и Redis. Эти мета-модели описывают, соответственно, структуру UML MCI, общую NoSQL модель и физические модели Cassandra, SSDB, Neo4j и Redis.

Используем язык QVT для реализации правил преобразования, обеспечивающих два прохода: концептуальный к логическому и логический к физическому. Предложим следующие шаги:

После формализации концепций, присутствующих в исходной модели (UML Class Diagram) и в целевой модели (Generic NoSQL model) преобразования UML-to-GenericModel, здесь представлен автоматический переход от концептуального PIM к логическому PIM. Этот переход выполняется цепочкой преобразований:

Шаг 1: Каждая диаграмма класса DCL преобразуется в базу данных, где BD.N = DCL.N.

Шаг 2: Каждый класс $c \in C$ преобразуется в таблицу $t \in T$, где $t.N = c.N$;

Каждый атрибут класса $a^c \in c.A^c$ преобразуется в табличный атрибут a^t, где $a^t.N = a^c.N$, $a^t.Ty = a^c.C$, а затем добавляется в список атрибутов его преобразованного контейнера t, который $a^t \in t.A^t$;

Идентификатор объекта с трансформируется в идентификатор строки t, где $Id^t.N = Id^c.N$ и $Id^t.Ty = Rid$, затем добавляется в список t атрибутов типа $Id^t \in t.A^t$.

Шаг 3: Каждая связь $l \in L$ степени 2, связывающая два класса c_1 и c_2, трансформируется в связь $r \in R$, связывающую таблицы t_1 и t_2, соответствующие классам c_1 и c_2, где $r.N = l.N$, $r.Cp^r = \{(t_1, cr^{c_1}),(t_2, cr^{c_2})\}$.

Шаг 4: Каждое соединение $l \in L$ степени n (при $n > 2$) приводит к (1) появлению новой таблицы t^l с собственным идентификационным атрибутом ld^{t^l}, где $t^l.N = l.N$, $t^l.A = \{Id^{tl}\}$ и (2) набор из n двоичных связей $\{r_1, \ldots, r_n\}$, $\forall i \in [1.n]$ r_i связи t^l в другую таблицу t^l, соответствующую родственному классу c_i, где $r_i.N = (tl.N)\_(ti.N)$ и r_i. $Cp^r = \{(t^l, null), (t_i, null)\}$.

Шаг 5: Каждый класс c_{asso} ассоциаций между n классами $\{c_1, \ldots, c_n\}$ ($c_n \geq 2$) трансформируется как звено степени строго выше 2 в (1) новую таблицу t^{asso}, где $t^{asso}.N = l.N$, $t^{asso}.A = c_{asso}.A^{asso}$ и (2) набор n двоичных отношений $\{r_1, \ldots, r_n\}$, $\forall i \in [1\ldots n]$ ri связывает tasso с другой таблицей ti, соответствующей родственному классу ci, где $ri.N = (t^{asso}.N)\_(ti.N)$ и r_i. $Cpr = \{(t^{asso}, null), (t_i, null)\}$.

Окончательный результат состоит из модели данных, содержащей элементы, необходимые для реализации БД, и набора руководящих принципов, специфичных для СУБД SSDB.

Преобразование объекта к общей модели (1) является первым шагом в процессе Object-to-NoSQL. Она транслирует диаграмму входного класса UML в общую модель NoSQL (2); эта модель соответствует логической PIM-модели. Преобразование общей модели в физическую (3) является вторым этапом, который генерирует физические модели NoSQL (PSM) (4) и набор ограничений (5) из общей модели.

Результаты

В результате исследования существующих методов объединения различных источников информации вы-

явлена проблема отсутствия принципиальных подходов к интеграции данных в единое информационное пространство. На основе рассмотрения действующих моделей преобразования информации построен подход к интеграции информации в виде модели «объект-характеристика», которая дает возможность обрабатывать данные разных форматов.

Решена задача определения модели ассоциации объектов и характеристик основных представлений данных. Построена новая информационная структура интеллектуального ситуационного центра.

Разработаны инструменты многоуровневого преобразования информации, которые состоят из цепочки преобразований с использованием общей модели, расположенной на промежуточном уровне между DCL (концептуальным уровнем) и моделью реализации информации в базы данных (физическим уровнем).

Предложена методика формирования физических моделей информации из разнородной неструктурированной и слабоструктурированной информации. Эта методика совместима с четырьмя типами СУБД NoSQL: колонками, документами, графиками и ключевым значением.

Модели данных (концептуальные, логические и физические), используемые в разработанном процессе, соответствуют мета-моделям, которые предложены для выполнения целей рассмотренных этапов: от концептуального к логическому, а затем от логического к физическому.

ЛИТЕРАТУРА

1. Симанков В.С., Дриленко М. В. Методические основы выбора платформ представления информации в интеллектуальном ситуационном центре // Современная наука: актуальные проблемы теории и практики.— 2020.— № 8., г. Москва;
2. Симанков В.С., Дриленко М. В. Методические основы преобразования информационных потоков от концептуальной к физической модели данных в интеллектуальном ситуационном центре // Перспективы науки.— 2020.— № 7, г. Тамбов;
3. Len Silverston — The Data Model Resource Book, Vol. 1: A Library of Universal Data Models for All Enterprises.— Принстон, США: Wiley Publishing, 2019;
4. David C. Hay — Enterprise Model Patterns: Describing the World (UML Version).— Энн-Арбор, США: Technics Publications, LLC, 2019;
5. Michael Blaha — Patterns of Data Modeling (Emerging Directions in Database Systems and Applications).— Вашингтон, США: CRC Press, 2019;
6. Martin Fowler — Analysis Patterns: Reusable Object Models.— Энн-Арбор, США: CRC Press, 2019.
7. Левин Н. А. Алгебра многомерных матриц как универсальное средство моделирования данных и ее реализация в современных СУБД [Текст] / Левин Н. А., Мунерман В. И., Сергеев В. П. // Системы и средства информатики.— Москва: Наука, 2014.— Вып. 14.— С. 86–99.
8. Magoulas Roger Big data: Technologies and techniques for large scale data [Electronic Resours] / Roger Magoulas, and Lorica Ben.— Access mode: http://assets.en.oreilly.com/1/event/54/mdw_online_bigdata_radar_p df.pdf.
9. Kossmann D. Personal Data Spaces [Electronic Resours] / D. Kossmann, J. P. Dittrich.— Access mode: http://www.inf.ethz.ch/news/focus/res_focus/ feb_2006/index_DE.
10. Hooman J. Equivalent semantic models for a distributed Data Space architecture [Текст] / J. Hooman, J. van de Pol // Formal Methods for Components and Objects.— Berlin; Heidelberg: Springer, 2003.— P. 182201.
11. The Open Archives Initiative Protocol for Metadata Harvesting Protocol Version 2.0 of 2002–06–14. [Electronic Resours].— Access mode: http://www.openarchives.org/OAI/2.0 /openarchivesprotocol.htm.
12. C.U. Kumarasinghe, K.L.D.U. Liyanage, W.A.T. Madushanka and R.A.C.L. Mendis. (2015, September). Performance Comparison of NoSQL Databases in Pseudo Distributed Mode: Cassandra, MONGODB & Redis [Online].
13. B.F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, “Benchmarking cloud serving systems with ycsb”. In Proceedings of the 1st ACM symposium on Cloud computing (New York, NY, USA, 2010), SoCC ’10, ACM, pp.143–154.
14. Chinonso, Okereke, Osemwegie Omoruyi, Kennedy Okokpujie, and Samuel John. “Development of an Encrypting System for an Image Viewer based on Hill Cipher Algorithm”, Covenant Journal of Engineering Technology 1, no. 2 (2017).
15. Инмон Б. Производительность систем хранилищ данных [Текст] / Инмон Б. // Performance In The Data Warehouse Environment.— 2016.— № 4.— С. 41–48.

© Дриленко Максим Владимирович (mvdrilenko@gmail.com), Симанков Владимир Сергеевич (vs@simankov.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»