О схеме данных для представления классификационных систем

On Data Schemes for Classification Systems Representation

Про схему даних для представлення класифікаційних систем

Калачихин В. Ю.
Российская государственная библиотека, Москва, Россия

Vladimir Yu. Kalachikhin
Russian State Library, Moscow, Russia

Калачихін В. Ю.
Російська державна бібліотека, Москва, Росія

Описывается схема структурированного представления классификационных систем и возможности ее использования при поиске.

The scheme of classification systems structural representation and opportunities for its use for searching are described.

Описується схема структурованого представлення класифікаційних систем і можливості її використання під час пошуку.

При проектировании автоматизированных библиотечно-информационных систем возникает нужда в представлении различных тезаурусов, справочников, библиографических систем и других подобных конструкций — суть по-разному структурированных перечней объектов различной природы. Следует ожидать, что подход к этой задаче с более-менее общих позиций позволит её достаточно эффективно решить и, быть может, получить новые возможности.

Так, будем считать, что придётся иметь дело со всем разнообразием природных и социальных объектов, явлений и сущностей, являющихся достоянием человеческой культуры. Будем полагать также, что в силу как общего несовершенства знания, так и ограниченности реальных потребностей каждая рассматриваемая сущность имеет некоторое неопределённое, и, в общем, наперёд неизвестное количество приписываемых ей свойств, наименований, пояснений и других подобных атрибутов. Нужно иметь возможность расширять список рассматриваемых атрибутов, и приписывать каждой сущности произвольный, и каждой — свой, перечень этих атрибутов.

Интересующие нас сущности мы хотим объединять в различные, по возможности — произвольные, структуры. Структуры также могут быть связаны между собой.

Рассматривая задачу реализации всего перечисленного в виде реляционной базы данных, можно видеть, что она распадается на две части — задачу реализации сущностей, и задачу представления структур сущностей.

Сущности

Структура данных для объектов, представляющих сущности реального мира, может быть следующая:

- Сами объекты представляются уникальным идентификатором (ID), перечень которых хранится в соответствующей таблице. ID не имеет никакого содержательного смысла, а имеет лишь одно свойство — он никогда не изменяется, только появляется и исчезает. ID олицетворяет объект в самом абстрактном виде, без каких-либо его свойств.
- Произвольное количество различных свойств (атрибутов) объектов хранится в других таблицах, по одной на каждый обрабатываемый тип свойства. Практически в библиотечно-библиографических системах достаточно только одной таблицы, содержащей атрибуты текстового типа. Атрибутам может приписываться различный содержательный смысл (пользовательская семантика) из расширяемого списка семантик.

Такая структура данных известна как «инвертированная структура» (инвертированная организация) базы данных.[5] Она позволяет иметь переменное число атрибутов для каждой уникальной сущности, позволяет добавлять и удалять атрибуты, в некоторых пределах изменять структуру данных без необходимости обязательного изменения клиентских приложений. Вместе с тем оказывается, что смысл хранимых данных не выражается в структуре базы данных.

Действительно, каждый объект реального мира представлен в нашей базе данных абстрактным идентификатором, при каждом из которых имеется некоторое количество не менее абстрактных атрибутов. Сведе-

ний о том, какие из них названия, какие — описания, и в каком случае какие из этих атрибутов для какой цели следует применять — в структуре базы данных не содержится.

Обычных выход из этого положения — «объектно-ориентированный» подход к построению базы данных, когда имеется процедура, выполняющая обработку атрибутов объекта, и жёстко связанная с этим объектом (технически, например, хранится в той же базе данных, вместе с прочими атрибутами). Эта процедура оперирует значениями атрибутов и выражает их семантику в пользовательском отношении. В целом же семантика данных остаётся вне структуры базы данных, и, в общем, вне сопутствующего общего программного обеспечения.

Такой подход, например, демонстрирует активно развивающееся xml-представление форматов marc. Вся семантика полей в marcxml вынесена в значения атрибутов, а сама структура записи едина для всех вариантов и абстрактна. Это полная противоположность традиционным вариантам marc, где семантика выражается структурой записи. Поскольку стремление выражать семантику в структуре и породило многообразие marc форматов, возможно, использование marcxml позволит вновь говорить о marc'e как о едином общепонятном стандарте представления библиографической информации.

На практике, для целей представления классификационных систем достаточно иметь один, универсальный обработчик атрибутов объектов, который имеет свой специальный набор данных, и знает из них, когда, для чего и каким образом употребить тот или иной из имеющихся текстовых атрибутов.

Структуры

Имеющиеся объекты нужно связать отношениями различных типов. Перечень возможных типов отношений содержится, например в ГОСТ 7.25 «Тезаурус информационно-поисковый одноязычный».[3,4] Если мы обратимся к форматам MARC, то там мы обнаружим и другой, и более обширный список возможных отношений. Есть и другие варианты. [6] Тем не менее, в этом многообразии выделяются отношения, связывающие объекты в структуры типа графов, и, так сказать, «индивидуально-парные» отношения, не образующие структур. Например, отношения «выше», «ниже» связывают объекты в древовидную иерархическую структуру, когда как отношение «смотри» связывает только пару объектов, и никакой структуры не образует. Таким образом, мы должны уметь связывать имеющиеся у нас объекты в графы и уметь устанавливать индивидуальные отношения между парой объектов.

Для построения графов можно воспользоваться простой конструкцией, известной как «точечная пара» — составить из идентификаторов пару предок — потомок. С помощью точечной пары можно выразить произвольный граф, в том числе и древовидные структуры классификационных систем. Тот факт, что такой способ построения не препятствует использованию одного объекта в одной структуре произвольное число раз, порождая в ней циклы, может считаться как недостатком, так и важным достоинством способа. Мы предпочитаем считает его достоинством, решая проблемы циклов при обходе структуры алгоритмическими средствами.

Для целей представления классификационных систем точечная пара, как минимум, обладает достаточной выразительной мощью. С помощью такой структуры можно адекватно представить любую из известных классификационных систем, будь то УДК, ББК, или предметные рубрики Библиотеки конгресса. Хранение точечных пар в реляционной базе данных позволяет использовать механизм СУБД для манипулирования структурой, и тем самым достаточно эффективно с машинной точки зрения решать некоторые свойственные ей вычислительные проблемы.

Для представления индивидуальных связей между парой объектов достаточно иметь возможность хранить при каждом атрибуте объекта ссылку на другой объект. Смысл получившегося отношения с точки зрения пользователя будет выражаться в тексте этого атрибута.

Уровень абстрагирования

Вынесение семантики из структуры базы данных продиктовано объективной необходимостью иметь средства для хранения наперёд неизвестных объектов, связанных наперёд неизвестными отношениями. Попытка заранее определить хотя бы список этих отношений рамками упомянутых ранее ГОСТов и выразить их явно в базе данных приводит к построению довольно громоздкой и труднообозримой информационной структуры, не обладающей, тем не менее, полной выразительностью. [1] Абстрактная, отделённая от пользовательской семантики конструкция, основанная на хранении объектов в базе данных инвертированного типа, а связей между ними в виде точечных пар компактна и не усложняется по мере роста сложности хранящейся в ней информации.

Можно расширить описанный подход к построению базы данных, и ввести ещё один уровень абстракции. Добавив понятие типа объекта мы сможем манипулировать классами сущностей и, например, хранить в той же базе данных и документы. Для электронной библиотеки это означает возможность более тесной интеграции объектов хранения и поисково-справочного аппарата.

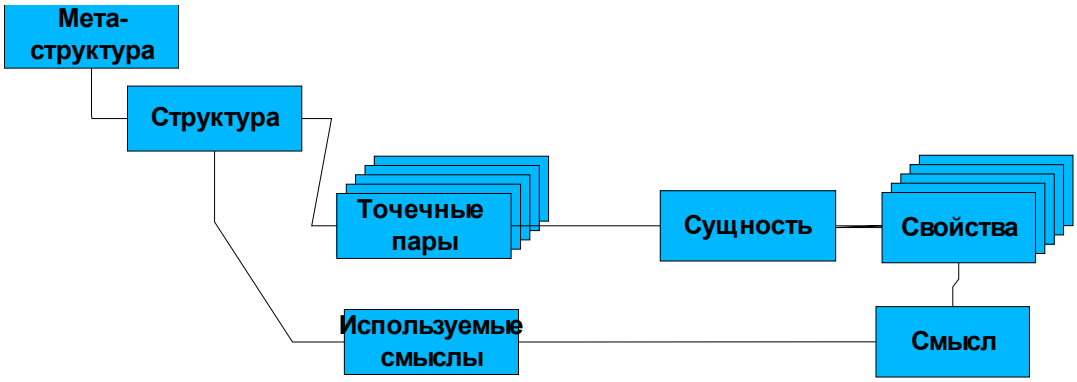
На самом деле, при полном удалении из базы данных пользовательской семантики и оперировании объектами простых (например, машинных) типов мы можем хранить в такой системе вообще всё что угодно, главное, иметь подходящие обработчики данных. Но нам кажется, библиографическая база данных не должна полностью уподобляться файловой системе...

Открытость

Предполагая необходимость предоставления информации сторонним потребителям, желательно уметь делать это через стандартную коммуникационную среду (таковой является интернет), для неопределённого количества клиентов, снабжённых средствами доступа точно неизвестных возможностей. Предлагаемая информационная структура в целом соответствует идеям SemanticWeb, по крайней мере, в части спецификации свойств отдельно от объектов и возможности наделения объектов одного типа разным набором свойств без необходимости оповещать об этом всех пользователей. Это обстоятельство позволяет предполагать, что задача предоставления информации сторонним потребителям может быть достаточно эффективно решена на основе подходов SemanticWeb. Основная проблема здесь — выбор достаточно общепринятой схемы данных и составление корректного RDF описания.

Реализация

Практическая реализация сказанного в базе данных «Программно-информационного комплекса ведения и использования классификационных систем (ПИК ВИКС)» выглядит следующим образом (уникальные ключи выделены жирным шрифтом):



Объекты

Отношение «*Сущности*»

состоит из одного атрибута —
ID_сущности — автонумеруемое целое.

Отношение «*Свойства сущностей*»

атрибуты:
ID_сущности
ID_свойства — автонумеруемое целое
Свойство — текст
ID_смысла_свойства
ID_ похожей_сущности
ID_похожей_структуры

Для хранения собственно свойства используется текстовый атрибут «Свойство», в котором может хранится текст произвольной длины. Многие СУБД позволяют хранить в таком поле произвольный двоичный объект.

Для реализации парных отношений между сущностями служат атрибуты ID_ похожей_сущности и ID_похожей_структуры. Эти атрибуты в соответствии со смыслом свойства, к которому они относятся, позволяют реализовать ссылки типа «смотри также», «смотри в» и тому подобное.

При необходимости можно добавить к отношению «***Свойства сущностей***» атрибут «язык».

Пользовательская семантика свойства указывается в атрибуте «ID_смысла_свойства». Для перечня пользовательских семантик существует

Отношение «*Смыслы*»

ID_смысла_свойства — автонумеруемое целое
Смысл_свойства — текст

Текст выражает собственно смысл в человеческом понимании. Например, в случае ББК атрибут «Смысл_свойства» может содержать, например, текст «Индекс ББК таблиц для массовых библиотек». Если у него атрибут «**ID_смысла_свойства**» будет равен, допустим, 10, то в кортеже из отношения «***Свойства сущностей***» для некоторой сущности атрибут «Свойство» будет иметь значение «6/8», а атрибут «ID_смысла_свойства» — значение «10». Т.е., атрибут будет иметь смысл «Индекс ББК таблиц для массовых библиотек», равный «6/8» (индекс деления «Общественные и гуманитарные науки»).

Структуры

Перечень имеющихся структур, таких, как классификационные системы или тезаурусы, хранится в

Отношении «Перечень *структур*»

ID_структуры — автонумеруемое целое
Наименование_структуры — строка
ID_коренной_сущности

Сами структуры — в

Отношении «*Структуры*»

ID_структуры
ID_сущности-родителя
ID_сущности-потомка

Как видно, в представлении структуры отсутствует указание, какие именно атрибуты сущности следует использовать в качестве названия или описания для показа пользователю. Эта информация хранится в отдельном

Отношении «*Используемые свойства*»

ID_структуры
ID_ смысла_свойства
Используется_в — фиксированный перечень

По существу, это служебная информация программы — обработчика данных объекта. Для каждой структуры указывается, какие атрибуты сущностей в ней используются, и в каком качестве. Атрибут используется_в указывает обработчику данных объекта, что делать с этими данными. Он используется для отображения в интерфейсе значений атрибутов сущностей с указанным смыслом на соответствующем месте.

Связь между «основными» и «подчинёнными» структурами, необходимая для выражения таких объектов, как специальные деления в ББК, описывается

Отношением «*Метаструктура*»

ID_структуры-родителя
ID_структуры-потомка

Реальная база данных информационной системы ПИК ВИКС включают ещё и некоторую служебную информацию, служащую для управления пользовательским интерфейсом, самими пользователями, администрирования и т.п. В данном случае организация этих данных не представляет интереса.

Легко видеть, что описанная структура данных находится в третьей нормальной форме, обладает достаточными для заявленных целей выразительными свойствами и позволяет эффективно решать обычные задачи, связанные с хранением, использованием и разработкой классификационных систем. Кроме этого, появляется простая возможность создания общих справочников, например, единых нормативных файлов для всех имеющихся классификационных систем. При рациональном построении системы алфавитный перечень атрибутов, имеющих смысл наименований сущностей, является естественным алфавитно-предметным указателем, единым для всех имеющихся в системе справочников, тезаурусов и классификаци-

онных систем. Это открывает путь к созданию читательской поисковой системы, позволяющей свободно сочетать предметный и систематический поиск при составлении поискового образа документа.

Литература

1. Бездушный А.А., Бездушный А.Н., Нестеренко А.К., Серебряков В.А., Сысоев Т.М. RDFS как основа среды разработки цифровых библиотек и Web-порталов // Электронные библиотеки. — 2003. — Вып. 3 http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2003/part3/BBNSS
2. Березовский А.М. Об одной модели формирования поискового образа документа и поискового образа запроса: проблемы и решения. // Восьмая международная конференция «LIBCOM-2004» Доклады и тезисы докладов. — М. : ГПНТБ России, 2004
3. ГОСТ 7.24-90 Тезаурус информационно-поисковый многоязычный. Состав, структура и основные требования к построению
4. ГОСТ 7.25-2001 Тезаурус информационно-поисковый одноязычный. Правила разработки, структура, состав и форма представления
5. Мейер М. Теория реляционных баз данных. — М.: Мир, 1987.
6. М.Х. Нгуен, А.С. Аджиев Описание и использование тезаурусов в информационных системах, подходы и реализация // Электронные библиотеки. — 2004. — Т. 7. — Вып. 1 (http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part1/NA)
7. Сукиасян Э .Р. Пришло ли время «закрывать» систематический каталог ? // Науч. и техн. б-ки. — 2002. — № 12. — С. 29