ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

А.А. Кресов, В.В. Уваров

МОДЕЛЬ И АЛГОРИТМ СОГЛАСОВАНИЯ АТРИБУТИВНОЙ ИНФОРМАЦИИ ПРИ ИНТЕГРАЦИИ РАЗНОРОДНЫХ ИНФОРМАЦИОННЫХ РЕСУРСОВ

Рассматривается актуальная проблема согласования на примере интеграции двух систем; приводится обобщенная модель многоуровневого согласования атрибутивной информации; выполняется переход от общей модели к частной, включающей блок интеллектуального принятия решений; приводится блок-схема алгоритма согласования.

Системы, модели многоуровнего согласования, атрибутивная информация.

Согласование данных является важной составляющей процесса интеграции разнородных информационных ресурсов. Существует множество форм представления информации, для каждой из которых необходимы принципиально разные методы и средства согласования. Остановимся на атрибутивной информации (которую можно представить в виде объектов с некоторым набором атрибутов), как наиболее удобной форме для анализа и хранения в реляционных и объектно-реляционных базах данных. Для того чтобы ответить на вопрос, что такое согласование атрибутивной информации и для чего оно необходимо, рассмотрим пример.

Пусть существует некоторая информационная система D, задачей которой является учет информации по договорам физических лиц, и система L для учета лицензий, выданных частным предпринимателям (физическим лицам). На определенном этапе возникла необходимость интеграции информации из D и L в хранилище A (рис. 1).

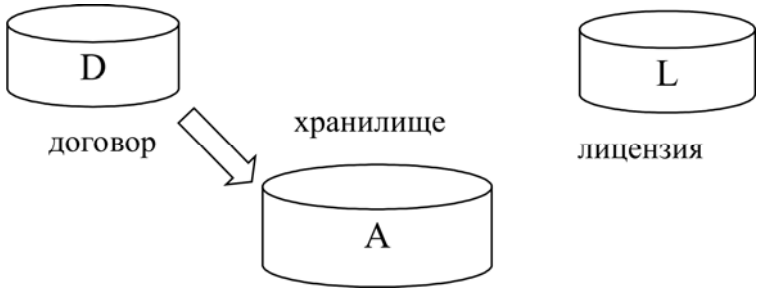

Рис. 1. Интеграция двух систем в хранилище

В системах D и L существует один и тот же тип объекта «Физическое лицо» (предприниматель), и вероятно, что на одного и того же предпринимателя может быть оформлен и договор, и лицензия. Следовательно, если данные будут перенесены из D и L в хранилище L в виде договоров, лицензий и физических лиц с сохранением связей между объектами, то в хранилище произойдет дублирование объектов типа «Физическое лицо», что может привести к ошибкам при анализе и построении отчетов, коллизиям при поиске и другим

нежелательным последствиям. Правильнее было бы сначала перенести данные из одной системы в хранилище, а при переносе данных из второй системы выполнять проверку на существование в хранилище идентичного объекта типа «Физическое лицо» и, если объект существует, создать на него ссылку, не создавая повторно сам объект. В этом случае необходимо будет ответить на вопрос вида «Иванов А.А. из системы L и Иванов И.И. из системы D — это одно и то же лицо?».

Таким образом, задачей согласования будет являться *установление идентичности* двух или более объектов, полученных из разных систем, по их атрибутивному составу.

Вышеизложенный пример показывает, что атрибутов {фамилия, имя, отчество} явно недостаточно для установления идентичности объектов «Физическое лицо». Если к этим атрибутам добавится серия и номер паспорта физического лица, то по ним можно будет однозначно идентифицировать объекты, поскольку пара атрибутов {серия паспорта, номер паспорта} уникальна для каждого физического лица. В результаты мы пришли к понятию «множество идентифицирующих атрибутов», т.е. минимальный набор атрибутов, который позволяет установить идентичность объектов разных систем.

В некоторых случаях определение множества идентифицирующих атрибутов еще не решает проблему согласования. Вернемся к нашему примеру. Допустим, в системе D для некоторых договоров серия и номер паспорта юридического лица не указаны, а указан его ИНН. При согласовании нам потребуется уже более сложная процедура поиска вида «если указан ИНН, то искать идентичный объект по ИНН, иначе искать по серии и номеру паспорта». Таким образом, множество идентифицирующих атрибутов в совокупности с правилами их использования образуют правила согласования атрибутивной информации.

Процесс согласования в нотации IDEF0 выглядит следующим образом (рис. 2).

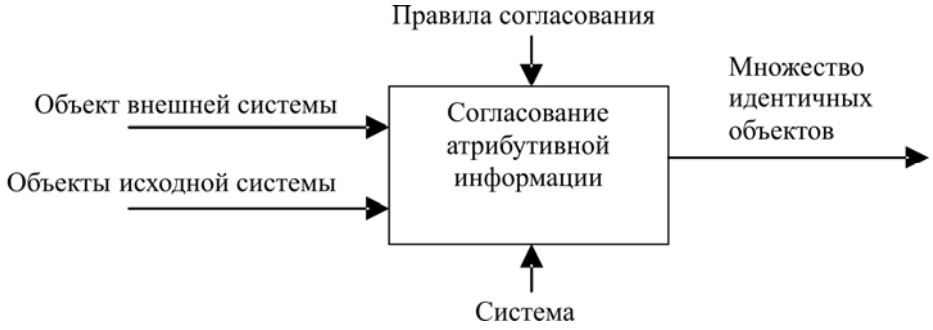

Рис. 2. Диаграмма A0 процедуры согласования

Результатом согласования является множество идентичных объектов исходной системы. В идеале это множество должно содержать единственный объект, однако не исключена ситуация, когда в результате будет найдено несколько объектов, идентичных объекту внешней системы. Причины этому могут быть разные, такие как: наличие дублирующих объектов в исходной системе, отсутствие некоторых идентифицирующих атрибутов, неверно заданные правила согласования и др. В этом случае может быть проведен следующий этап (уровень) согласования среди полученного множества объектов по более сложным правилам, требующий больше вычислительных ресурсов. Затем

следующий этап, и так до тех пор, пока множество не будет содержать один или ноль объектов. Обобщенная модель многоуровневого согласования в нотации IDEF0 представлена на рис. 3.

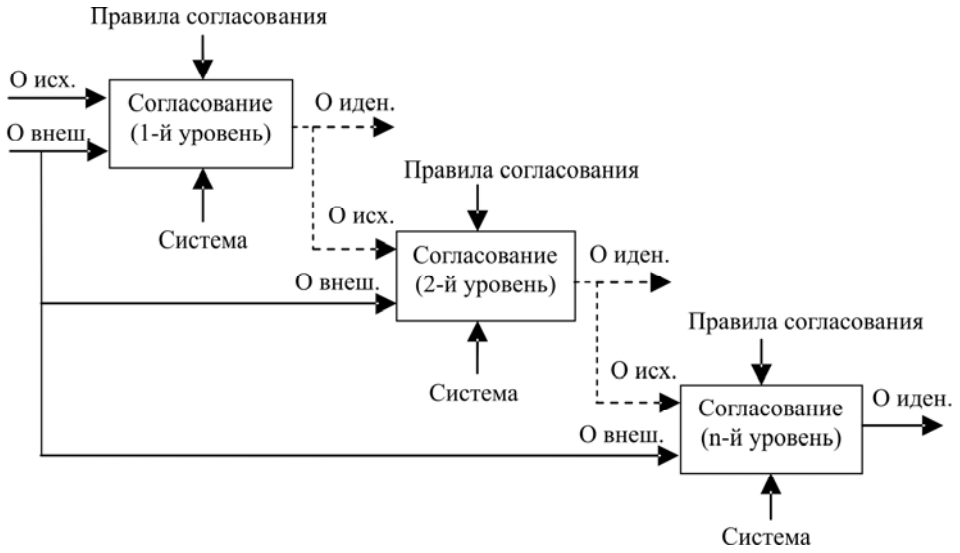

Рис. 3. Обобщенная модель многоуровневого согласования

На каждом уровне количество элементов в выходном множестве объектов уменьшается, и затем для множества выполняется следующий уровень согласования. Пунктирная линия означает, что на каждом уровне возможно завершение процедуры согласования, если в результате согласования было получено пустое множество или единственный объект.

На первый взгляд очевидным способом решения был бы переход сразу к последнему, наиболее сложному уровню согласования, минуя предыдущие, что избавило бы от разработки программного кода для остальных уровней. Однако если последний этап требует выполнения сложных аналитических функций, которые будут применяться к большому количеству объектов, то процесс согласования затянется на неопределенное время. Оптимальнее было бы сначала произвести предварительную выборку объектов на основе фильтра, а затем применять более сложные функции.

В большинстве ETL-систем интеграции ограничиваются одним уровнем согласования, реализуемым средствами языка запросов SQL. Если в результате SQL-запроса для объекта внешней системы получено более одного идентичного объекта, это считается ошибкой и объект выносится в так называемый «файл ошибок». Что делать с такими объектами, решает уже сам оператор/аналитик. Такой подход имеет как преимущества, так и недостатки. С одной стороны, он прост в реализации, с другой — процесс согласования не интеллектуализирован и может потребовать от оператора большого объема ручной работы по согласованию отдельных объектов.

Практически все современные учетные системы и хранилища данных используют СУБД в качестве системы хранения. Перейдем от общей многоуровневой модели согласования к частной, оптимальной именно для таких систем. На первом уровне согласования производится выборка данных с использованием возможностей СУБД. Запрос будет иметь вид:

выбрать все О, для которых А1 = *x*1 и А2 = *x*2 и … и А*n* = x*n*,

где А1, А2, …, А*n* — множество идентифицирующих атрибутов объектов О; *x*1, *x*2, …, *xn* — значения соответствующих атрибутов в объекте внешней системы.

Разработка генератора подобных запросов является тривиальной задачей и не требует дополнительного рассмотрения. Входными параметрами генератора запросов будут являться список идентифицирующих атрибутов и значения этих атрибутов в объекте внешней системы. Если в результате выборки получено более одного объекта, выполняется переход ко второму уровню.

На втором уровне к полученному множеству объектов применяется *интеллектуальный выбор*, реализующий более сложные правила согласования. Если и в этом случае было получено более одного объекта (что маловероятно), система переходит к третьему уровню, где выбрать из множества предстоит оператору. Чтобы не замедлять общий процесс загрузки данных в хранилище, система переходит к обработке следующего объекта, не дожидаясь решения оператора. Загрузка ожидающих объектов производится дополнительно по завершении основного процесса.

Модель согласования для частного случая представлена на рис. 4.

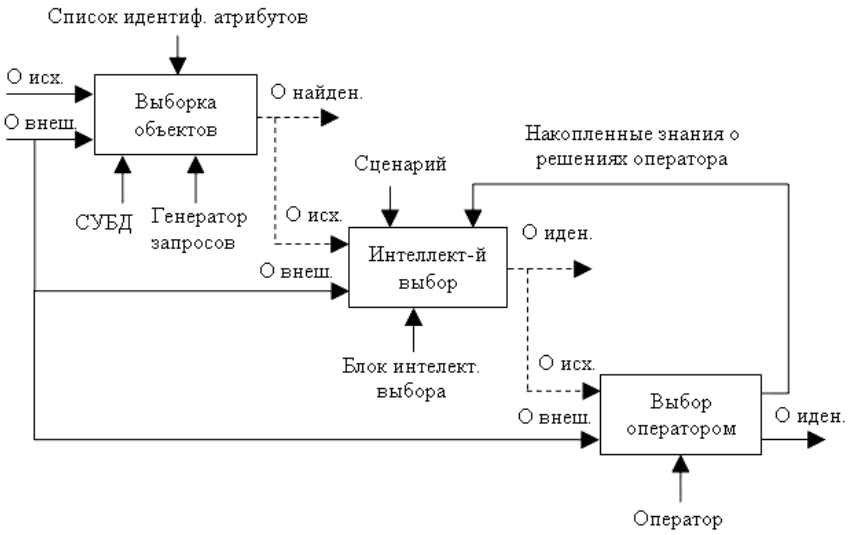

Рис. 4. Модель согласования для частного случая

Блок интеллектуального выбора является обучаемым элементом системы. Результаты выбора оператором сохраняются и тем самым расширяют базу знаний данного блока. В дальнейшем при возникновении подобной ситуации блок будет в состоянии принять решение без участия оператора. В общем виде каждую запись базы знаний можно представить как пару «ситуация — выбранный объект». Описание ситуации может содержать такие элементы, как название внешней системы, тип объекта, значения некоторых атрибутов объекта и др. Кроме базы знаний о решениях оператора в блок могут закладываться сценарии (скрипты) согласования для данного типа объектов. Добавление сценария выполняется через интерфейс системы и непосредственно перед выполнением процедуры согласования подгружается, компилируется и включается в работу. Реализовать такую возможность позволяет

один из скриптовых движков, которые входят в поставку вместе со стандартными библиотеками для языков высокого уровня, таких как Java, C#. Блок-схема алгоритма представлена на рис. 5.

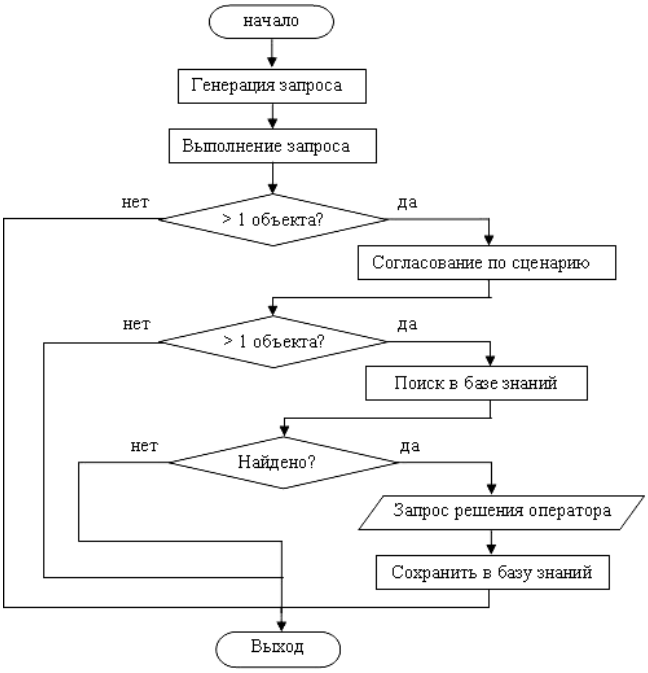

Рис. 5. Блок-схема алгоритма согласования в общем виде

Перспективой развития данного подхода является внедрение блока, выполняющего поиск и выявление закономерностей в базе знаний и решений оператора по завершении основного процесса. Использование закономерностей повысит число случаев успешного применения базы знаний и соответственно сократит работу оператора.

Таким образом, представленная модель описывает основные идеи и принципы, которые могут послужить базисом при разработке ETL-систем хранилищ данных. Основным преимуществом данных систем будет являться обучаемость, возможность расширения алгоритмической базы без внесения изменений в ядро системы. При геометрическом росте объемов информации за последние годы такой подход позволит сократить объем ручного труда оператора или по крайней мере удержать его на приемлемом уровне.

A.A. Kresov, V.V. Uvarov

MODEL AND ALGORITHM OF ADJUSTMENT AS APPLIED TO ATTRIBUTIVE INFORMATION UNDER INTEGRATION OF HETEROGENOUS INFORMATION RESOURCES

The article presents urgency of a problem of adjustment illustrated by integration of two systems, giving a generalized model of multiple adjustment of attributive information; passing from a general to a particular model including a unit of intellectual decision making; together with supplying a bloc scheme of adjustment algorithm.

Systems, models of multiple level adjustment, attributive information.