Семантическая реконсиляция прикладных данных на основе моделей [1]

В.А. Семенов, С.Г. Ерошкин, А.А. Караулов, И.В. Энкович

Аннотация. Рассматривается задача семантически корректной и функционально содержательной реконсиляции прикладных данных. Задача имеет ключевое значение для успешного применения современных технологий оптимистической репликации и создания перспективных распределенных приложений, таких как системы коллективной инженерии, мобильные базы данных, семантические сети. Рассматривается модельно-ориентированный подход к семантической реконсиляции данных, описываемых на языках объектно-ориентированного моделирования EXPRESS, UML/OCL, ODL/OQL. На основе статического анализа формальных спецификаций прикладной модели выявляются отношения зависимости и отношения порядка между операциями в конкурентных транзакциях и применяется аппарат логического, полисиллогистического вывода для выработки непротиворечивых (семантически корректных) и полных (обеспечивающих полноту результирующей транзакции) планов реконсиляции. Обсуждаются конкурентные преимущества предложенного модельно-ориентированного подхода, а также особенности его алгоритмической и программной реализации.

1. Введение

Семантическая реконсиляция — фундаментальная научная и прикладная проблема, тесно связанная с применением современных технологий оптимистической репликации в распределенных системах и имеющая индустриально важные приложения, такие как системы коллективной инженерии (concurrent engineering environments), мобильные базы данных, распределенные сервисы, семантические сети [1].

Использование в подобных приложениях оптимистической репликации позволяет отказаться от необходимой централизации управления распределенными информационными ресурсами, свойственной пессимистическим моделям транзакций, и обеспечить возможность

[1] Работа поддержана грантом Президиума РАН в рамках программы фундаментальных исследований “Математические и алгоритмические проблемы информационных систем нового поколения”.

эффективной одновременной работы пользователей и приложений с реплицированными версиями данных. Однако необходимость обнаружения и разрешения конфликтов в конкурентных транзакциях после их применения делают постановку задачи реконсиляции довольно нетривиальной. Требования корректности и полноты реконструируемой итоговой транзакции, а также условия семантической целостности результирующего представления данных существенно усложняют поиск допустимых решений.

В настоящей работе обсуждается предложенный подход к семантической реконсиляции с использованием формальных спецификаций модели данных. Предполагается, что спецификации представлены декларативным образом на объектно-ориентированных языках EXPRESS [2], UML/OCL [3], ODL/OQL [4] (или на подобных им) и охватывают как структуры данных, так и заданные на них семантические ограничения.

На основе формального анализа конкурентных транзакций выявляются возможные отношения между элементами данных и операциями, и применяется логический вывод для их семантически корректного и функционально содержательного слияния. При подобном подходе операции транзакций рассматриваются как объекты системы посылок и заключений, а отношения зависимости и порядка между ними — как правила логического вывода. Выбранная система отношений позволяет, с одной стороны, промоделировать сложные семантические зависимости между операциями транзакций, а с другой стороны — привлечь для решения рассматриваемой задачи хорошо развитый аппарат математической логики, в частности, методы полисиллогистического вывода [5] и интервальной темпоральной логики [6]. С целью обобщения рассматриваются бинарные и множественные отношения, а также учитываются возможные альтернативные способы задания отношений в аналитической и табличной форме.

Обсуждаемый подход относится к так называемому смешанному классу методов и допускает конструктивную реализацию в составе распределенных систем с традиционными клиент-серверной и равнозначной P2P архитектурами на основе поддерживаемых протоколов транзакций. Он также реализуем в автономных приложениях, использующих файловый обмен данными, посредством непосредственного сравнения версий данных и реконструкции журналов изменений.

Подход обладает рядом важных достоинств, связанных с гарантированным обеспечением целостности прикладных данных, которые определяются масштабными, семантически сложными моделями, при относительно невысоких вычислительных затратах. Возможности формализованного и содержательного применения подхода к широким классам приложений оптимистической репликации делают его привлекательным для реализации перспективных систем коллективной инженерии с долгими транзакциями и мобильных информационных систем с неустойчивым соединением и прерываемыми сессиями.

Раздел 2 посвящен обсуждению общих аспектов применения модельно-ориентированного подхода к построению прикладных интегрированных систем и необходимости ослабления фундаментальных принципов ACID (атомарности, целостности, изоляции и долговечности) для приложений оптимистической репликации. В разделе 3 описывается общая схема процесса семантической реконсиляции, а также на примере языка объектно-ориентированного моделирования EXPRESS приводится классификация типовых элементов данных и семантических ограничений. Проведение семантического анализа конкурентных транзакций на основе введенной классификации основных видов отношений зависимости и порядка между операциями обсуждается в разделе 4. Особое внимание уделяется установлению отношений на основе формального анализа семантических ограничений в прикладной модели данных. Процесс декомпозиции и редукции задачи с помощью определяемого графа и матрицы реконсиляции рассматривается в разделе 5. В разделе 6 приводятся конструктивные утверждения относительно корректности постановки задачи и поиска решений на основе эквивалентных преобразований матрицы реконсиляции. В заключении обсуждаются конкурентные преимущества предложенного модельно-ориентированного подхода, а также особенности его алгоритмической и программной реализации.

2. Роль модельно-ориентированного подхода

Модельно-ориентированный подход становится важным доминирующим направлением в инженерии программного обеспечения, критичного по отношению к требованиям мобильности, интероперабельности, развертываемости в разнородных средах. Деятельность международных сообществ по разработке соответствующих информационных стандартов и многофакторных прикладных моделей, прежде всего, в рамках ISO 10303 (STEP — Standards for Representation and Exchange of Product Model Data) [7] и OMG MDA (инициатива Model-Driven Architecture группы индустриальных компаний Object Management Group) [8] способствует развитию этих тенденций.

Вместе с тем, применение модельно-ориентированного подхода при построении прикладных интегрированных комплексов для проведения масштабных междисциплинарных проектов в науке и промышленности, обнаруживает ряд фундаментальных проблем. Прежде всего, это проблема синергетической организации работ с участием большого числа партнеров проекта, вовлеченных в единую творческую и производственную деятельность, но профессионально, организационно и географически отделенных друг от друга. Традиционные решения, использующие в качестве ключевых компонентов интеграции системы управления базами данных и системы документооборота, как правило, имеют довольно ограниченное применение. Они либо не обеспечивают целостность проектной информации,

либо существенно ограничивают возможности участников работать одновременно, что критично сказывается на качестве, стоимости и сроках проведения работ.

Обеспечение эффективного мультидоступа к проектным данным при необходимой централизации управления ими и существенно распределенном, автономном и продолжительном характере проведения индивидуальных пользовательских сессий представляется в данном случае наиболее критичным. Предпринятые попытки ослабления базовых принципов ACID на основе идей последовательной целостности (serial consistency), использования версий данных (multiversion concurrency), применения специальных протоколов транзакций (transaction access patterns), а также разработки специальных механизмов альтруистических блокировок (altruistic locks) себя не оправдывают [9].

Вместе с тем, хорошо известны примеры успешного применения в ряде приложений оптимистической репликации. Это популярные распределенные сервисы UseNet; персональные помощники PDA (Personal Digital Assistants); мобильные базы данных, включая широко известную реализацию Bayou; системы совместной разработки программного обеспечения, в частности, CVS (Concurrent Versions System). Однако в применяемых в них методах игнорируются сложные семантические зависимости между данными и не обеспечивается желаемая целостность итогового представления.

Например, реализуя типовые средства синтаксической реконсиляции текстовых документов, CVS не позволяет контролировать и, тем более, гарантировать семантическую корректность итоговых текстов программ на том или ином языке реализации. Однако, если в системах программной инженерии текстовые данные всегда могут быть скорректированы непосредственно программистами, в научных и промышленных приложениях, оперирующих со сложно структурированными моделями, их коррекция не может быть осуществлена усилиями пользователей.

В связи с этим естественной выглядит попытка разработки и применения модельно-ориентированного подхода к семантической реконсиляции с использованием формальных спецификаций модели данных. Подобные спецификации предоставляют дополнительные возможности для статического и динамического анализа зависимостей между элементами данных и выработки непротиворечивых (семантически корректных) и полных (обеспечивающих полноту результирующей транзакции) планов реконсиляции.

Исследования в этой области в настоящее время привлекают как коммерческие компании, так и многочисленные университетские коллективы.

3. Общий подход к реконсиляции

В соответствии с главным принципом оптимистической репликации, приложения, выполняющиеся в распределенной среде, находятся либо на

этапе изолированного исполнения, либо на этапе реконсиляции. При изолированном выполнении приложение оперирует локальной репликой разделяемых данных, приводя их в некоторые новые состояния. Все выполняемые действия записываются в журналах транзакций, являющихся упорядоченными множествами локально применяемых операций. Все действия детерминированы и обратимы. Применение всех операций, занесенных в журнал, к начальной версии данных должно приводить к той же окончательной версии. И наоборот, применение обратных операций в противоположном порядке должно возвращать данные из конечного состояния в первоначальное состояние.

Неважно, ведется ли журнал в приложении постоянно или реконструируется путем сравнения начальной и текущей версий данных. В этом отношении мы следуем смешанному классу методов (hybrid state-transfer and operation-transfer methods), охватывающему и анализ изменений в состоянии данных и контроль последовательностей выполнения операций в транзакциях. В ряде случаев, например, при определении и проверке предусловий для операций учет порядка их выполнения может быть исключен из анализа без потери содержательности результатов для приложения. Однако в дальнейшем мы будем рассматривать наиболее общий случай.

Будем считать, что транзакции корректны в том смысле, что они могут быть корректно воспроизведены на основе журналов и гарантированно приводят к семантически целостному и функционально значимому представлению данных. Это важное допущение мотивируется тем, что используемые приложения, запущенные на локальной станции и обрабатывающие локальную реплику данных, работают правильно. В этом предположении, если начальное состояние данных было корректно, то корректная транзакция должна порождать итоговое состояние, удовлетворяющее всем семантическим ограничениям целостности и несущее необходимый функциональный смысл, определяемый пользователем.

Одним из главных преимуществ обсуждаемого подхода является возможность сохранять свойства целостности при реконсиляции данных или, более точно, реконсиляции соответствующих журналов транзакций. Требование поддерживать семантическую целостность реплицированных данных рассматривается в нашем исследовании как базовый принцип функционально содержательной реконсиляции.

3.1. Этапы реконсиляции

В предлагаемом подходе общий процесс реконсиляции включает в себя семь этапов, представленных на Рис. 1:

Рис. 1. Основные этапы процесса реконсиляции

— *Сравнение и гармонизация:* На первом этапе две расходящиеся реплики сравниваются между собой для выявления различий и реконструкции журналов. Даже если во время сессии велась постоянная журнализация, и все изменения зафиксированы, проводится дополнительный анализ, необходимый для выявления эквивалентных представлений в сравниваемых репликах и их гармонизации.

— *Семантический анализ:* Все операции транзакций из двух журналов анализируются с учетом семантики самих операций и семантики ограничений для элементов данных; эти ограничения определяются формально специфицированной информационной моделью. Между операциями устанавливаются отношения зависимости и отношения

порядка (предшествования). Заметим, что проводимый на этой стадии семантический анализ охватывает как статические, так и динамические зависимости, которые могут быть реально проверены лишь с учетом текущего состояния элементов данных и фактических параметров операций транзакций. Поэтому мы считаем, что результаты, полученные на данном этапе, могут иметь различный уровень доверительности. Выявленные статические зависимости и отношения — абсолютно достоверны (необходимы и достаточны для корректности результирующей транзакции), динамические зависимости и отношения — предположительно достоверны (достаточны, но вовсе не необходимы для корректности транзакции) и поэтому в ряде случаев могут быть проигнорированы и уточнены на более поздних этапах валидации и коррекции.

— *Декомпозиция и редукция:* На основе анализа отношений определяются классы эквивалентности, импликативные цепочки и свободные группы операций. Для сформированных групп операций переустанавливаются отношения зависимости с другими операциями и группами. Целью данного этапа является уменьшение количества логических элементов (переменных и отношений) при логическом анализе транзакций и предотвращение комбинаторных проблем, обычно возникающих на последующих этапах.

— *Планирование:* Строятся возможные варианты результирующей транзакции, которые удовлетворяли бы всем выявленным отношениям зависимости и порядка и тем самым гарантировали бы ее корректность. Во внимание принимается также требование наиболее полного покрытия результирующей транзакцией множеств операций в исходных транзакциях, поскольку именно этим определяется ее функциональная содержательность. В случаях, когда число приемлемых вариантов слишком велико, может быть построен план реконсиляции, который бы учитывал возникшие конфликты и определял альтернативные способы их разрешения, например, на основе дерева принятия решений.

— *Валидация:* Если в сопутствующем анализе игнорировалась часть ранее установленных (предположительно достоверных) отношений, они должны быть перепроверены с учетом фактического текущего состояния данных и значения параметров операций, поскольку только в этом случае можно гарантировать корректность итоговой транзакции. В случае обнаружения нарушений анализируемые варианты транзакций отвергаются, а ранее построенные планы реконсиляции уточняются путем исключения заведомо неприемлемых решений.

— *Коррекция:* Если содержательные решения не были найдены (например, были найдены лишь тривиальные решения, состоящие в принятии всех операций первой или второй транзакции), то конструктивным подходом может оказаться ослабление предположительно достоверных отношений

между операциями и пересмотр результатов семантического анализа. Если на этапе валидации были выявлены нарушения, но все-таки желательно представить критичные операции в результирующей транзакции; могут быть применены корректирующие методы, вносящие необходимые исправления и переводящие транзакцию и данные в корректное состояние. В обоих случаях процесс возвращается к более ранним этапам реконсиляции;

— *Функциональный отбор:* Наконец, решения, прошедшие этап валидации, ранжируются и отбираются пользователем согласно функциональным требованиям, предъявляемым к результатам реконсиляции. При наличии плана пользователь последовательно уточняет детали итоговой транзакции, включая в ее состав требуемые операции. С целью автоматизации подобных действий могут быть определены и применены политики исключения и консолидации, основанные на априорно заданных функциональных критериях и приоритетах.

В настоящей статье мы сосредоточимся на втором, третьем и четвертом этапах, охватывающих семантический анализ транзакций с использованием формальных спецификаций прикладной информационной модели, а также методы логического, прежде всего, полисиллогистического вывода для поиска решений рассматриваемой постановки задачи реконсиляции. Методы семантической декомпозиции транзакций более подробно обсуждаются в нашей работе [10].

3.2. Действия

Действие — это базовая операция, выполняемая приложением над реплицированными данными. Оно хранится в журнале и становится доступным для анализа сразу после записи или реконструкции журнала. Мы будем рассматривать действие как структуру, состоящую из шести элементов:

— *целевой объект*, идентифицирующий элемент (или элементы) данных, на которое направлено действие; если язык информационного моделирования допускает конструкции запросов, то целевой объект также может быть выражен в терминах запросов;

— *статус*, определяющий метод для динамической проверки, применено ли уже действие или нет; методы статуса не имеют побочных эффектов и возвращают логическое значение;

— *предусловие*, определяющее аналогичный метод для проверки, можно ли выполнить данное действие при текущем состоянии данных;

— *операция* — метод доступа или модификации элемента (элементов) данных, определяемого целевым объектом; предполагается, что операция не оказывает побочного эффекта на данные, не определенные явно как объект действия, но может нарушить их семантическую корректность в контексте наложенных ограничений и вносимых сторонних изменений;

операция возвращает логическое значение, указывающее на успешное или неуспешное выполнение действия;

— *тег*, хранящий всю информацию, относящуюся к параметрам операции; этот элемент необходим для воспроизведения журналов и реконструкции данных;

— *инверсия* — метод инвертирования действия, позволяющий отменить принятые ранее изменения и вернуть данные в первоначальное состояние; реализация метода может быть связана как с изменением соответствующего признака в существующем экземпляре действия, так и с конструированием нового экземпляра действия с инвертированными свойствами.

В рамках рассматриваемой объектно-ориентированной метамодели целевым объектом действия может быть прикладной объект, группа прикладных объектов или целая популяция, определяемая некоторым объектным запросом. Объектом действия может быть также атрибут прикладного объекта или его отдельный элемент, если атрибут сложно структурирован, например, являясь некоторой коллекцией. В большинстве случае это прикладные объекты и их атрибуты.

Стандартный набор действий включает в себя операции создания, удаления, перемещения прикладного объекта, модификации атрибута или его элементов. Семантика операций может быть уточнена в зависимости от специфики рассматриваемого класса приложений. Например, в ряде случаев изменение атрибута целого типа может интерпретироваться как его увеличение или уменьшение на некоторое значение. Изменение атрибута, являющегося множеством элементов некоторого типа, может быть одновременно представлено как включение и/или исключение соответствующих элементов. Подобная детализация имеет смысл только в тех случаях, когда нарушения семантических ограничений проявляются на подобном уровне. Мы рассчитываем, что языки объектно-ориентированного моделирования позволяют конкретизировать репертуар и детальную семантику действий независимо от особенностей конкретного приложения и конкретной модели данных.

3.3. Типы данных и виды ограничений

Объектно-ориентированные языки моделирования, такие как EXPRESS, UML/OCL, ODL/OQL, предоставляют широкий спектр декларативных и императивных конструкций для описания структур данных и накладываемых на них семантических ограничений.

В частности, в языке EXPRESS определяются простые типы данных с общепринятой семантикой: вещественный, целый, числовой, булевский, логический, строковый, двоичный строковый. Сложные типы охватывают четыре вида коллекций: списки, множества, массивы, мультимножества, а также перечисления, выборки и переопределяемые типы, уточняющие

семантику основных типов. Для объектных типов предоставляются развитые механизмы множественного наследования, специализации атрибутов, полиморфного определения производных методов, декларации ограничений, позволяющие пользователям определять сложные модели данных.

На Рис. 2 представлена общая классификация типов, поддерживаемых языком EXPRESS. Полужирным курсивом отмечены исходные типы. Стрелки показывают отношения наследования между ними. Конструкции для определения типов пользователем отмечены обычным шрифтом.

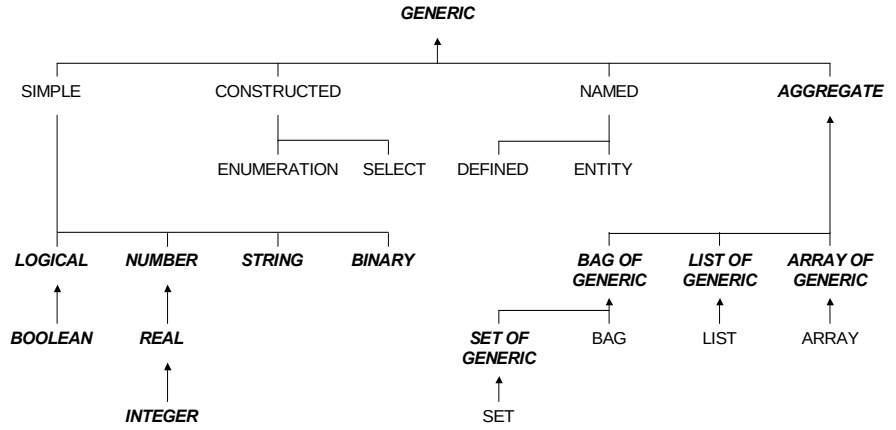

Рис. 2. Классификация исходных типов языка EXPRESS

В зависимости от контекста определения различаются три основных вида ограничений: правила для простых пользовательских типов данных, локальные правила для объектных типов и глобальные правила для наборов объектных типов (Рис. 3).

Кроме неявно предполагаемых условий соответствия присваиваемых значений их типам, данные виды ограничений позволяют задавать:

— ограничение длины символьных и двоичных строк;

— мощность коллекций (множеств, мультимножеств, списков, массивов);

— кардинальность прямых и обратных ассоциаций в объектах;

— уникальность элементов в коллекциях-множествах;

— уникальность значений атрибутов (и групп атрибутов) на объектных популяциях;

— атрибуты с обязательно установленными значениями;

— массивы с установленными элементами;

— область значений для отдельных атрибутов, объектов и целых объектных популяций; область значений задается предикатом, алгебраически выраженным декларативными и императивными конструкциями языка (в том числе управляющими конструкциями, функциями, процедурами и т.д.).

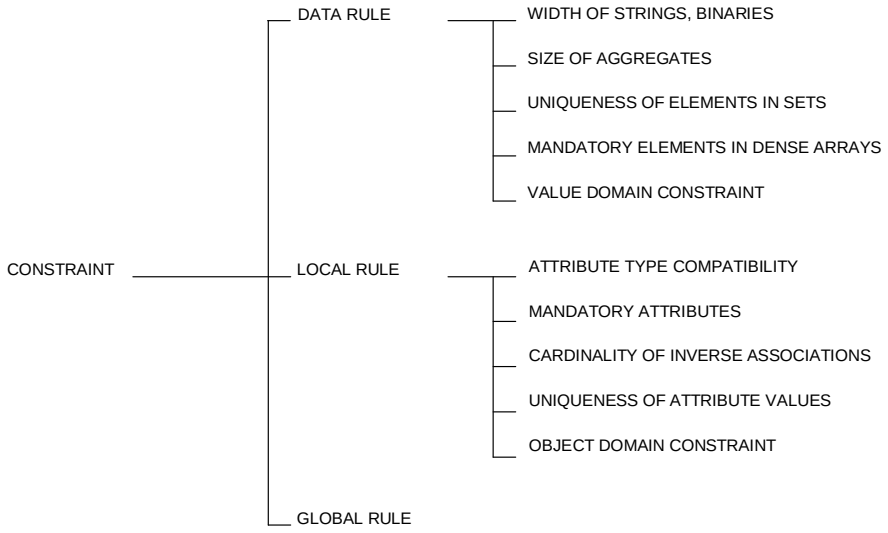

Рис. 3. Классификация видов ограничений EXPRESS

Для получения более подробного описания можно обратиться к упомянутым выше стандартам, регламентирующим языки.

4. Семантический анализ транзакций

Процесс реконсиляции состоит в объединении журналов изменений с тем, чтобы построить новый журнал и, воспроизведя его, реконструировать согласованное представление реплицированных данных. В идеальном случае полученный журнал должен содержать все действия исходных журналов и приводить к представлению данных, которое бы удовлетворяло необходимым семантическим ограничениям и пользовательским требованиям. Тем не менее, простые приемы исключения и консолидации семантически подобных действий в совместном журнале не обеспечивают выполнения необходимых ограничений и нарушают целостность итогового представления данных.

Вместо этого мы предлагаем использовать информационную модель данных для более детального анализа операций транзакций и применения методик группирования и упорядочивания для перманентной поддержки данных в целостном состоянии.

4.1. Понятие корректной транзакции

Пусть X – область значений, которые могут принимать данные рассматриваемой прикладной объектно-ориентированной модели, $C(x) \equiv \{c_i(x) \mid i = 1,..,I\}$ – система ограничений, наложенных моделью на состояние $x \in X$ и принимающих логическое значение *true*, если ограничение удовлетворено, и *false*, если оно нарушено. Если все ограничения удовлетворены в точке $x \in X$, тогда мы говорим, что x находится в корректном состоянии, или $C(x) = true$. Если хотя бы одно из условий не выполняется, т.е. $c_i(x) = false$, то состояние $x \in X$ считается некорректным, что выражается соответствующим образом: $C(x) \equiv c_1(x) \wedge c_2(x) \wedge ... \wedge c_I(x) = false$.

Для некоторых ограничений $c_i(x)$ могут быть определены корректирующие методы $r_{ij} \in R$ таким образом, что если $c_i(x) = false$, то применение метода приводит данные в состояние, удовлетворяющее ранее нарушенному ограничению, т.е. $x' = r_{ij}(x)$, $c_i(x') = true$. Мы считаем, что для любого ограничения $c_i(x)$, $i = 1,\ldots I$ могут быть определены соответствующие корректирующие методы $r_{ij} \in R$. Однако они должны применяться с учетом всех наложенных ограничений $c_{i'}(x)$, $i' \neq i$, $i' = 1,\ldots I$ и возможного нарушения некоторых их них, если они уже были удовлетворены.

Если данные находятся в состоянии $x \in X$, то выполнение действия $t(x_t, p_t)$ с параметрами p_t над целевым объектом $x_t = \mathrm{Pr}_{Dt}\, x$, являющимся некоторой проекцией x на область определения операции D_t, приведет к переводу целевого объекта в состояние $x'_t \in X$, а всех данных – в состояние $x' \in X$. Мы будем обозначать это следующим образом: $x'_t = t(x_t)$ и $x' = t(x)$. Отдельно выполненное действие может нарушать целостность данных, т.е. если $C(x) = true$, то $C(x')$, где $x' = t(x)$, не обязательно принимает значение *true*. Но мы требуем, чтобы любая корректная транзакция вида $T = \{ t_k(x_t, p_t) \mid k = 1,\ldots,K \}$ сохраняла целостность данных. Если транзакция T корректна и начальное состояние данных корректно ($C(x) = true$), то $C(x') = true$, $x' = T(x)$. При этом не требуется, чтобы каждое действие $t_k \in T$ или некоторые группы действий t_{k1}, $t_{k2},\ldots$, $t_{kn} \in T$ перманентно сохраняли целостность данных во время выполнения транзакции.

Для применения каждое действие корректной транзакции должно удовлетворять соответствующему предусловию. Поскольку действия в транзакциях не подчиняются коммутативным свойствам, порядок их применения существенен как для выполнения предусловий, так и для конечных результатов. Поэтому вводимое понятие корректной транзакции предполагает выполнимость всех определенных предусловий для действий и учитывает порядок их следования. В дальнейшем рассматриваются только корректные транзакции за исключением, естественно, некоторых временных представлений журналов.

Теперь определим возможные виды отношений между операциями. Временный журнал транзакции $T = T' \cup T''$ – это упорядоченное множество

действий, полученное путем объединения журналов реконсилируемых транзакций T' и T'', причем если x – начальное состояние данных и x' и x'' – расходящиеся реплики, то $x' = T'(x)$ и $x'' = T''(x)$.

4.2. Отношения зависимости и порядка

Отношения зависимости между операциями D определяются логическими операторами отрицания и импликации следующим образом. Для операций t_1, $t_2 \in T$, если $t_1 \rightarrow t_2$, то журнал должен содержать t_2 при условии, что он содержит t_1. Если $\neg t_1 \rightarrow \neg t_2$, то в журнале должна отсутствовать операция t_2, если в нем отсутствует операция t_1. Эти отношения несимметричны, рефлексивны и транзитивны. Для операций t_1, $t_2 \in T$, если $t_1 \rightarrow \neg t_2$, то в журнале не может присутствовать t_2, если он содержит t_1. Если $\neg t_1 \rightarrow t_2$, то журнал должен содержать t_2, если в него не входит t_1. Эти отношения симметричны, не рефлексивны и не транзитивны. Эти четыре логические отношения с характеристическими функциями, приведенными в Таблице 1, рассматриваются как основные отношения зависимости. Также мы считаем полезным использование симметричных отношений эквивалентности $t_1 \sim t_2 \equiv t_1 \rightarrow t_2 \wedge t_2 \rightarrow t_1$ и взаимоисключения $t_1 \oplus t_2 \equiv t_1 \rightarrow \neg t_2 \wedge \neg t_1 \rightarrow t_2$. Отношение эквивалентности устанавливается между двумя операциями t_1, $t_2 \in T$ и означает, что эти операции могут встречаться в транзакции T только совместно. Отношение взаимоисключения между t_1, $t_2 \in T$ обязывает транзакцию T включать в себя либо t_1, либо t_2 и запрещает содержать обе эти операции или ни одну из них.

t_1	t_2	$t_1 \rightarrow t_2$	$t_1 \rightarrow \neg t_2$	$\neg t_1 \rightarrow t_2$	$t_1 \sim t_2$	$t_1 \oplus t_2$
0	0	1	1	0	1	0
0	1	1	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	1	0

Таблица 1. Характеристические функции бинарных отношений зависимости.

В некоторых случаях приходится рассматривать более сложные множественные отношения между операциями. В общем виде они представляются характеристической функцией $D(t_1, t_2, t_3, \ldots)$ и соответствующей таблицей значений, подобной Таблице 1.

В качестве примера рассмотрим множественное отношение кардинальности $D^{(n:m)}(t_1^+, \ldots, t_{I+}^+, t_1^-, \ldots, t_{I-}^-)$. Данное отношение связано с ограничениями кардинальности ассоциаций и размерности коллекций элементов данных в объектно-ориентированной модели. Оно считается выполненным, если

$$n \le card(T^+) - card(T^-) \le m, \text{ где}$$

$$T^+ = \{ t_i^+, i \in (1, I^+) \mid t_i^+ = true \}, T^- = \{ t_i^-, i \in (1, I^-) \mid t_i^- = true \},$$

а функция *card*() возвращает мощность соответствующих подмножеств операций T^+ и T^-. Характеристические функции отношений кардинальности $D^{(0:0)}(t_1^+, t_2^+, t_1^-, t_2^-)$, $D^{(1:1)}(t_1^+, t_2^+, t_1^-, t_2^-)$ и $D^{(2:2)}(t_1^+, t_2^+, t_1^-, t_2^-)$ приведены в Таблице 2. Функции других возможных отношений могут быть выражены через представленные функции с помощью следующих очевидных тождеств:

$$D^{(n:m)}(t_1^+, \ldots, t_1^-, \ldots) \equiv D^{(-m:-n)}(t_1^-, \ldots, t_1^+, \ldots),$$
$$D^{(n:m)}(t_1^+, \ldots, t_1^-, \ldots) \equiv D^{(n:n)}(t_1^+, \ldots, t_1^-, \ldots) \vee D^{(n+1:n+1)}(t_1^+, \ldots, t_1^-, \ldots) \vee \ldots \vee D^{(m:m)}(t_1^+, \ldots, t_1^-, \ldots).$$

$t_1^+ \in T$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$t_2^+ \in T$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$t_1^- \in T$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$t_2^- \in T$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$D^{(0:0)}(t_1^+, t_2^+, t_1^-, t_2^-)$	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0
$D^{(1:1)}(t_1^+, t_2^+, t_1^-, t_2^-)$	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0
$D^{(2:2)}(t_1^+, t_2^+, t_1^-, t_2^-)$	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Таблица 2. Характеристические функции множественных отношений кардинальности.

Часто требуется удовлетворить некоторое ограничение $c = c(x_1, x_2, \ldots)$, выраженное алгебраическим образом с помощью соответствующей логической функции нескольких переменных. Если конкурентные транзакции содержат операции модификации элементов данных, являющихся фактическими параметрами функции ограничения, то возможным способом сохранить целостность является принятие всех операций модификации первой транзакции или всех операций модификации второй транзакции. В этом случае между операциями устанавливается отношение алгебраической зависимости $D^c(t_1', t_2', \ldots, t_1'', t_2'', \ldots)$, где операции первой транзакции $t_1', t_2' \in T'$ и операции второй транзакции $t_1'', t_2'' \in T'$ модифицируют элементы данных, на которые наложено ограничение $c(x_1, x_2, \ldots)$. Будучи выполненным, данное отношение приводит к выделению классов эквивалентности для операций каждой транзакции и исключению ситуаций одновременного их применения в результирующей транзакции:

$$D^c(t_1', t_2', \ldots, t_1'', t_2'', \ldots) \equiv t_1' \oplus t_1'' \wedge t_1' \sim t_2' \wedge t_2' \sim t_3' \wedge \ldots \wedge t_1'' \sim t_2'' \wedge t_2'' \sim t_3'' \wedge \ldots$$

Отношение порядка или предшествования операций в транзакции P определим следующим образом. Для операций t_1, $t_2 \in T$, если $t_1 \angle t_2$, то

операция t_1 должна появиться до операции t_2 (не обязательно непосредственно перед ней) в любом решении, которое одновременно содержит операции t_1 и t_2. Отношение порядка несимметрично, не рефлексивно, но транзитивно. Так как отношение подразумевает наличие обеих операций в одной транзакции, отношение $t_1 \angle t_2$ может быть установлено только между теми операциями, которые не связаны отношениями зависимости $t_1 \rightarrow \bar{t}_2$ и $t_1 \oplus t_2$ непосредственно или косвенно через другие отношения.

Обсуждаемые отношения являются строгими в том смысле, что если операции не удовлетворяют некоторым установленным отношениям, то транзакция необходимо становится некорректной, что означает невозможность ее выполнения или потерю целостности воспроизводимыми данными. В ряде случаев могут быть рассмотрены более слабые отношения, обозначаемые $\langle D \rangle$ и $\langle P \rangle$, которые совпадают со строгими отношениями D, P за исключением того, что они имеют достаточный, но не необходимый характер для корректности транзакции. Другими словами, иногда они могут быть не выполнены, не нарушая условия корректности транзакции и целостности данных. Поскольку риск нарушения все равно остается, предварительные решения (транзакции и данные), полученные без учета подобных отношений, должны быть дополнительно проверены на более поздних этапах рассмотренного выше процесса реконсиляции.

4.3. Примеры формального анализа транзакций

Обратимся к следующей демонстрационной модели, формально описанной на языке EXPRESS (см. Рис. 4). Модель описывает объектный тип данных A с вещественными атрибутами x, y, z, целочисленным идентификатором id и квалификационным именем q, представленным множеством из трех строк. Модель определяет также объектные типы данных B и C, содержащие необязательную и обязательную ссылки соответственно на экземпляр объекта типа A.

Определение типа A содержит правила $Wr1$, $Wr2$, ограничивающие допустимую область значений атрибутов x, y, z, а также правило уникальности $Ur1$ для значений атрибута id, действие которого распространяется на всю популяцию объектов типа A.

Пусть начальное состояние данных $x = \{a \in A,\ b \in B,\ b.ref = a, \ldots\}$ и сформирован временный журнал совмещенной транзакции $T' \cup T'' = \{t_1' = new(b_1 \in B),\ t_2' = wr(b_1.ref,\ a),\ t_1'' = new(c_1 \in C),\ t_2'' = wr(c_1.ref,\ a),\ \ldots\}$, где символы new, wr, del обозначают операции создания объекта, модификации его атрибутов и удаления соответственно. В дальнейшем используются также символы in, rm, обозначающие операции добавления элементов в атрибуты типа коллекций и их исключения из подобных атрибутов. Тогда между операциями должны быть установлены следующие отношения. Импликация $t_2' \rightarrow t_1'$ вытекает из того, что атрибут $b_1.ref$ может быть установлен, если только объект b_1 существует. Сам же объект должен быть создан перед установкой

его атрибутов, поэтому $t_1' \angle t_2'$. Для операций второй транзакции должно быть установлено отношение эквивалентности $t_2'' \sim t_1''$, так как установка значения атрибута $c_1.ref$ обязательна и операции могут быть включены в итоговую транзакцию только совместно при отношении предшествования $t_1'' \angle t_2''$.

```
ENTITY A;
        x : REAL;
        y : REAL;
        z : REAL;
        q : SET [3:3] OF STRING;
        id : INTEGER;
UNIQUE
        Ur1: id;
WHERE
        Wr1 : (x + y + z < 1);
        Wr2 : (x + y < 1) AND (z < 1);
END_ENTITY;

ENTITY B;
        ref : OPTIONAL A;
END_ENTITY;

ENTITY C;
        ref : A;
END_ENTITY;
```

Рис. 4. Демонстрационная модель данных, формально описанная на языке EXPRESS

Аналогичные отношения $t_1' \rightarrow t_2'$, $t_1' \oplus t_2''$, $t_1' \angle t_2'$, и $t_1'' \angle t_2''$ могут быть установлены для операций в сформированном временном журнале $T' \cup T'' = \{t_1' = del(a),\ t_2' = wr(b_1.ref,\ 0),\ t_1'' = new(b_1 \in B),\ t_2'' = wr(b_1.ref,\ a),\ \ldots\}$. Отношение $t_1' \rightarrow t_2'$ вытекает из того, что удаление объекта a требует обнуления соответствующей ссылки $b_1.ref$, указывающей на него. Отношение $t_1' \oplus t_2''$ появляется, потому что операции удаления объекта a в первой транзакции и установки на него ссылки $b_1.ref$ во второй транзакции являются взаимоисключающими.

Для временного журнала $T' \cup T'' = \{t_1' = rm(a.q,\ldots),\ t_2' = in(a.q,\ldots),\ t_1'' = rm(a.q,\ldots),\ t_2'' = in(a.q,\ldots)\ \ldots\}$, чтобы сохранить атрибут q в корректном состоянии, необходимо потребовать, чтобы количество элементов соответствующего множества было ровно 3. Предполагая, что начальное состояние было корректным и применение каждой транзакции не нарушает целостности, мы устанавливаем отношение кардинальности $D^{(0:0)}(t_2'^{+}, t_2''^{+}, t_1'^{-}, t_1''^{-})$. Это означает, что количество добавляемых элементов равно количеству исключаемых элементов, и мощность множества остается неизменной.

Рассмотрим отношения между операциями, порождаемые правилами ограничения области значений. Для корректности итоговой транзакции с временным журналом $T' \cup T'' = \{t_1' = wr(a.x),\ t_2' = wr(a.y),\ t_1'' = wr(a.z),\ \ldots\}$ требуется выполнение отношения алгебраической зависимости $\langle D^{Wr1}(t_1', t_2', t_1'')\rangle$. В этом случае в итоговую транзакцию включаются все участвующие в ограничении операции первой или второй транзакции, и ограничение будет гарантированно удовлетворено. Отметим, что данное отношение носит предположительный характер, являясь достаточным, но не необходимым условием корректности транзакции. Поэтому при необходимости оно может быть снято на этапе семантического анализа транзакции, но тогда связанное с ней ограничение обязано быть проверено на последующем этапе валидации. Аналогичный вывод справедлив для отношения $\langle D^{Wr2}(t_1', t_2', t_1'')\rangle$. Однако более детальный анализ показывает, что данное отношение следует исключить из рассмотрения, так как модифицируемые в различных транзакциях атрибуты появляются в различных термах конъюктивной нормальной формы предиката ограничения области значений.

Наконец, рассмотрим отношения, устанавливаемые для ограничения уникальности. Предположим, что в каждой транзакции создается объект типа *A*, и инициализируется его обязательный атрибут *id*. Журнал для представления транзакции выглядит следующим образом: $T' \cup T'' = \{t_1' = new(a_1 \in A),\ t_2' = wr(a_1.id,\ id1),\ t_1'' = new(a_2 \in A),\ t_2'' = wr(a_2.id,\ id2),\ldots\}$. Чтобы удовлетворить ограничение уникальности значений атрибута *id*, в дополнение к отношениям импликации $t_2' \rightarrow t_1'$, $t_2'' \rightarrow t_1''$ и предшествования $t_1' \angle t_2'$, $t_1'' \angle t_2''$ следует установить отношение исключения $t_1' \oplus t_2''$. Однако такое отношение может оказаться слишком сильным, так как присваиваемые значения идентификаторов могут различаться. Поэтому здесь следует применить нестрогую форму отношения. Заметим, что даже если в дальнейшем на этапе валидации будет обнаружено, что идентификаторы случайно совпали, существует возможность переназначить их с использованием корректирующих методов.

Конечно, приведенные примеры не исчерпывают все содержательные случаи семантического анализа транзакций. Тем не менее, они описывают типовые ситуации, возникающие на данном этапе реконсиляции, и иллюстрируют возможности применения формального, основанного на спецификациях прикладной модели данных, подхода. Отметим только, что для определения алгебраических зависимостей между данными и установления соответствующих отношений между операциями транзакций могут быть использованы методы инкрементального анализа и верификации данных [11, 12].

5. Логический анализ

Реализация намеченного подхода тесно связана с проведением логического анализа отношений зависимости и порядка, выявленных между операциями в

конкурентных транзакциях на этапе семантического анализа. С целью формализации этапа решения предлагается использовать представление логической системы отношений в виде графа и/или матрицы реконсиляции и применить для эффективного поиска решений аппарат математической логики, в частности, полисиллогистический вывод и методы интервальной темпоральной логики. При этом декомпозиция и редукция логической системы рассматриваются и как самостоятельные этапы общего процесса реконсиляции, предваряющие непосредственный поиск решения, и как алгоритмические элементы единого метода, распространяемого на все переменные и отношения формализованной логической системы. Главной особенностью рассматриваемого метода является поиск непротиворечивых, семантически корректных решений, удовлетворяющих всем логическим отношениям, при обеспечении их полноты и репрезентативности. Последнее свойство принципиально для обсуждаемой задачи, поскольку для функциональной содержательности решения важно включить в итоговую транзакцию как можно больше операций исходных транзакций.

В случаях, когда в логической системе присутствуют только бинарные отношения, удобно иллюстрировать применение метода с помощью введенного графа реконсиляции, имеющего прозрачную графическую нотацию. Для представления множественных отношений в логической системе естественным выглядит обобщение понятия на гиперграф, однако мы предпочитаем использовать эквивалентное представление в виде матрицы реконсиляции. В настоящей статье рассматривается метод анализа с использованием графа реконсиляции.

5.1. Граф реконсиляции

Граф (гиперграф) реконсиляции *RG* определим как набор пяти множеств ($V = V_1 \cup V_2 \cup V_C$, $E = E_D \cup E_P$), где

— V_1, V_2 – множества вершин, соответствующих операциям исходных транзакций T' и T''. V_C – множество операций, применяемых в корректирующей транзакции. При проведении анализа каждой вершине на графе присваивается логический статус. Статус указывает, включается ли соответствующая операция в итоговый журнал или нет.

— E_D, E_P – множества ребер (гипер-ребер), которые соответствуют отношениям зависимости и порядка, установленным между операциями. Каждое ребро помечается семантическим ограничением, порождающим соответствующее отношение. Бинарные отношения графически представляются обычными линиями, а множественные отношения — прямоугольными элементами, опирающимися на узловые вершины.

Граф реконсиляции обладает рядом свойств, существенных для его конструктивного анализа. Так как предполагается, что каждая транзакция, участвующая в процессе реконсиляции, является корректной, ребра, соответствующие обратным импликациям $t_1 \rightarrow \neg t_2$ и $\neg t_1 \rightarrow t_2$, могут быть

установлены только между вершинами, принадлежащим разным транзакциям. В то же время ребра, соответствующие прямым импликациям $t_1 \to t_2$ и $\neg t_1 \to \neg t_2$, могут появиться как внутри отдельных транзакций, так и в разных транзакциях. Аналогично, между парой вершин одной транзакции не может существовать противоположных отношений порядка, что противоречило бы возможности их одновременного применения.

Следующие свойства связаны с обобщающими идеями сильных цепей зависимости и предшествования.

Пусть *RG(V,E)* — граф реконсиляции. Путь $v_1(t_1), v_2(t_2), ..., v_n(t_n) \in V$ вдоль ребер зависимости $e_1(D_1), e_2(D_2), \ldots, e_{n-1}(D_{n-1}) \in E_D$, начинающийся в вершине $v_1(t_1)$ и заканчивающийся в вершине $v_n(t_n)$, назовем цепью зависимости (или v_n зависит от v_1), если только существует соответствующая последовательность отношений зависимости $t_1 D_1 t_2$, $t_2 D_2 t_3$, …, $t_{n-1} D_{n-1} t_n$, эквивалентная простой импликации $t_1 D t_n$. Прямой зависимостью назовем цепь зависимости, эквивалентную импликации $t_1 \to t_n$ или $\neg t_1 \to \neg t_n$. Обратной зависимостью будем называть цепь, эквивалентную импликации $t_1 \to \neg t_n$ или $\neg t_1 \to t_n$.

В графе реконсиляции не существует циклов обратной зависимости. Если бы такой цикл был найден, это означало бы, что любой статус операции, приписанный ее вершине, не соответствует логическим отношениям. Это противоречит существованию тривиальных решений логической системы, в качестве которых всегда могут быть выбраны операции первой или второй транзакции.

Путь $v_1(t_1), v_2(t_2), ..., v_n(t_n) \in V$ на графе реконсиляции *RG(V,E)* вдоль ребер предшествования $e_1(P_1), e_2(P_2), \ldots, e_n(P_{n-1}) \in E_P$, начинающийся в вершине $v_1(t_1)$ и заканчивающийся в вершине $v_n(t_n)$, назовем цепью предшествования (или v_1 предшествует v_n), если только отношения предшествования $P_1, P_2, \ldots, P_{n-1}$ устанавливают порядок $t_1 \angle t_2, t_2 \angle t_3, \ldots, t_{n-1} \angle t_n$.

В графе реконсиляции не существует циклов предшествования, содержащих только вершины какой-либо одной транзакции. В противном случае это также противоречило бы существованию тривиальных решений задачи.

5.2. Логический вывод на графе реконсиляции

Обсудим метод, использующий графическую нотацию для графа реконсиляции на Рис. 5. Вершины графа изображены окружностями и помечены символами операций. Каждая вершина располагается в одной из областей, соответствующих исходным транзакциям и, возможно, применяемой корректирующей транзакции. На диаграмме области исходных транзакций располагаются на противоположных сторонах и отделяются от центральной области корректирующей транзакции двумя вертикальными линиями. В случаях, если коррекция отсутствует, на диаграмме размещаются лишь области исходных транзакций, отделенные друг от друга одной вертикальной линией.

Бинарные отношения зависимости между операциями изображаются дугами, помеченными упорядоченными парами черных и/или белых кружков, бинарные отношения предшествования — стрелками. Строгие отношения зависимости представляются сплошными дугами, а нестрогие отношения — пунктирными дугами. Строгие отношения предшествования заканчиваются темными стрелками, а нестрогие отношения предшествования заканчиваются светлыми стрелками. При использовании некоторых визуальных элементов метода полисиллогистического вывода [5] графическая нотация обеспечивает прозрачную и непротиворечивую интерпретацию обсуждаемой задачи. Для представления множественных отношений зависимости и порядка могут быть добавлены необходимые визуальные элементы, соответствующие ребрам соответствующего гиперграфа реконсиляции. На Рис. 5 приведен пример графа реконсиляции для рассмотренных выше транзакций.

Рис. 5. Графическая нотация элементов графа реконсиляции.

Опишем предлагаемый метод логического вывода, используя представленную графическую нотацию. Для выделения цепей зависимости и определения того, какие вершины могут зависеть от заданной вершины, удобно использовать следующее визуальное правило. Заметим, что цепями зависимости в графе реконсиляции являются только те цепи, ребра которых инцидентны общим вершинам с чередующимися цветами кружков. Это объясняется тем, что пара цветов, присваиваемая отношению импликации, соответствует запрещенной комбинации статусов операций, при этом черный цвет обозначает статус

включения, а белый — исключения. Таким образом, чередование цветов в инцидентных ребрах порождает цепочки логического вывода и соответствующие зависимости.

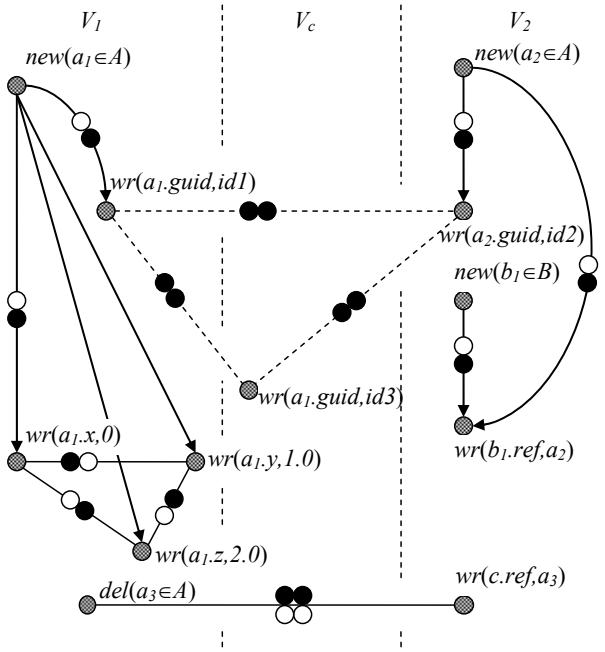

Рис. 6. Пример графа реконсиляции.

Общая схема метода вывода представляется следующими этапами.

На первом этапе все операции исходных транзакций объединяются в один временный журнал.

На втором этапе проводится поиск циклов прямой зависимости. Вершины найденных циклов формируют классы эквивалентности для операций, или, другими словами, множества операций, которые включаются в итоговую транзакцию только совместно. Таким образом, осуществляется декомпозиция транзакций на семантически связанные группы операций. В конечном итоге это приводит к снижению числа независимых переменных в решаемой логической системе, поскольку статус любой из вершин цикла однозначно определяет состояния всех остальных вершин.

На третьем этапе выявляются подобные цепи — логически эквивалентные цепи зависимости, начинающиеся и заканчивающиеся в одних и тех же вершинах графа. В результате проводимой редукции логической системы из

анализа исключаются избыточные отношения и зависимые переменные, связанные с внутренними вершинами подобных цепей.

На заключительном этапе проводится поиск вершин, порождающих конфликты. Это вершины, инцидентные ребрам обратной зависимости, а также вершины, образующие циклы предшествования. Согласно применяемым политикам реконсиляции и решениям пользователя, операции, соответствующие конфликтным вершинам, последовательно удаляются из журнала. Вместе с удаляемой операцией из журнала исключаются все зависимые операции с вершинами, располагающимися вдоль цепей прямой зависимости от исходной вершины. Для оставшихся вершин пересматривается их возможность порождать конфликты. Например, если удалить некоторую вершину из цикла предшествования, то оставшиеся вершины становятся неконфликтными относительно данной группы отношений. После разрешения конфликтов, оставшиеся в журнале операции упорядочиваются согласно отношениям предшествования. Полученная в результате транзакция удовлетворяет всем наложенным семантическим ограничениям и предусловиям, необходимым для корректного воспроизведения журнала.

Представленный метод был изложен в предположении строгих необходимых отношений зависимости и порядка между операциями транзакций. Он естественным образом обобщается на нестрогий случай в результате инициирования процесса семантической реконсиляции с наиболее полной системой ограничений и ее последовательного ослабления. Однако полученные результаты нуждаются в дополнительной валидации, а при выявленных нарушениях — и в возможной дополнительной коррекции. Поскольку коррекция вносит новые изменения, которые могут конфликтовать с операциями исходных транзакций, процесс должен быть повторен, начиная с этапа семантического анализа.

Организация более сложного итерационного процесса сама по себе не гарантирует нахождение решения. Однако решения, полученные таким образом, являются более значимыми, поскольку обеспечивают полноту представления результирующей транзакции. В случае неудачи метод допускает возвращение к предыдущим стадиям решения, учитывающим большее количество ограничений. Заметим, что тривиальные решения задачи всегда существуют, и применение метода в результате возврата к ее строгой исходной постановке гарантирует их нахождение.

6. Заключение

Представленный модельно-ориентированный подход обладает рядом важных конкурентных преимуществ, главным из которых является строгое обеспечение целостности результирующего реконсилированного представления данных и, как следствие, возможность эффективного применения в приложениях, использующих технологии оптимистической репликации и оперирующих масштабными, семантически сложными моделями данных.

Значительная универсальность, достигнутая за счет применения общего математического аппарата, не препятствует содержательному применению подхода к разнообразным классам приложений, включая системы коллективной инженерии с долгими транзакциями и мобильные информационные системы с прерываемыми сессиями. Благодаря использованию формальных методов статического анализа спецификаций прикладных моделей, удается преодолеть проблемы комбинаторного характера, свойственные многим другим методам. Важно отметить, что логический вывод является одним из наиболее затратных элементов предлагаемого подхода, поскольку число операций в конкурентных транзакциях и число установленных отношений между ними может быть значительно. Тем не менее, применение разреженных схем представления логических отношений и соответствующих матричных методов преодолевает эту проблему. Данный аспект является принципиально важным, поскольку позволяет существенно снизить вычислительную сложность и сделать возможным практическое применение подхода для важных научных и индустриальных приложений.

Литература

[1] Y. Saito, M. Shapiro. Optimistic Replication // In ACM Computing Surveys, Vol. 37, No. 1, March 2005, pp. 42–81.

[2] ISO 10303-11: 1994, Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.

[3] Unified Modeling Language (UML), Version 2.0, http://www.uml.org/#UML2.0.

[4] The Object Data Standard: ODMG 3.0. eds. R.G.G. Cattel, D.K. Barry. Morgan Kauffmann, 2000.

[5] А.Д. Закревский. Логика распознавания. — Мн.: Наука и техника, 1988.— 118 с.

[6] J.F. Allen, G. Ferguson. Actions and Events In Interval Temporal Logic. // In Technical report 521, University of Rochester, Computer Science Department, July 1994, pp.1–59.

[7] ISO 10303: 1994, Industrial automation systems and integration — Product data representation and exchange.

[8] OMG. Model Driven Architecture: How systems will be built, http://www.omg.org/mda.

[9] K. Ramamritham, P. Chrysanthis. Executive Briefing: Advances in Concurrency Control and Transaction Processing. IEEE Computer Society Press, 1997.

[10] Semenov V.A., Karaulov A.A., Semantic-Based Decomposition of Long-Lived Transactions in Advanced Collaborative Environments. // Proceedings of 6 European Conference on product and process modeling, ECPPM 2006, Valencia, September 11–15, 2006, ISBN 10: 0-415-41622-1, pp.223–232.

[11] Семенов В.А., Морозов С.В., Тарлапан О.А. Инкрементальная верификация объектно-ориентированных данных на основе спецификации ограничений. // Труды ИСП РАН, 2004, стр. 23-55.

[12] Semenov V.A., Bazhan A.A., Morozov S.V., Tarlapan O.A. Efficient Verification of Product Model Data: an Approach and an Analysis. // Proceedings of 22nd Conference on Information Technology in Construction, CIB-W78, Dresden, July 19-21, 2005, ISBN:3-86005-478-3, pp. 261-268.