
#Гиперграф:Интегратор

Организация единого информационно-функционального
пространства

023.ГГГ.004-22.01.00.019

Листов 40

Информация, содержащаяся в настоящем документе, может быть изменена без предварительного уведомления. Фонд развития технологий «Инженеры устройства мира» не дает никаких гарантий в отношении настоящего документа, включая (но не ограничиваясь этим) подразумеваемые гарантии о товарном качестве и пригодности для конкретной цели.

Фонд развития технологий «Инженеры устройства мира» не несет ответственности за ошибки, содержащиеся в настоящем документе, так же как и за случайный ущерб или за ущерб, нанесенный в связи с поставкой, эксплуатацией и использованием настоящего документа.

АННОТАЦИЯ

#Гиперграф:Интегратор – технология интеграции разнородных прикладных программ, в т.ч. унаследованных для создания единого информационного и функционального пространства над пространством гетерогенных систем. #Гиперграф:Интегратор обеспечивает технологию описания проекций разнородных систем в единой интеграционной модели и работу с данными других систем в режиме on-line.

#Гиперграф:Интегратор – обеспечивает миграцию от унаследованных программ в новые технологии и архитектуры путём использования технологии адаптивных проекций TableX.

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	8
2. ЦЕЛИ ИНТЕГРАЦИИ.....	10
3. АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ И РЕШЕНИЙ ПРОБЛЕМ ИНТЕГРАЦИИ	11
3.1. Направление 1: интеграция межсистемных сообщений	11
3.2. Направление 2: интеграция через дополнительные интегрирующие системы	12
3.3. Направление 3: интеграция в архитектуре SOA	12
4. ГИПЕРГРАФ-ПАРАДИГМА РЕШЕНИЯ ПРОБЛЕМЫ ИНТЕГРАЦИИ.....	14
5. ОБЗОР МЕТОДОВ ИНТЕГРАЦИИ	17
5.1. Интеграция на уровне пользовательских интерфейсов	17
5.2. Интеграция на уровне данных	17
5.3. Интеграция на уровне приложений	18
5.4. Метод Web-сервисов.....	18
6. Основные принципы #Гиперграф:Интегратор	19
7. СОСТАВ #Гиперграф:Интегратор	20
7.1. Технология адаптивных проекций TableX	21
7.2. Создание единого информационно-функционального пространства	23
7.2.1. <i>Аттестация интегрируемого ПО</i>	23
7.2.2. <i>Создание единого информационного хранилища данных</i>	23
7.2.3. <i>Синхронизация внешних хранилищ данных</i>	25
7.3. Анализ и контроль данных единого информационного хранилища	25
7.3.1. <i>Автодокументирование единого информационного хранилища</i>	25
7.3.2. <i>Анализ модели единого информационного хранилища</i>	26
7.3.3. <i>Анализ качества данных. Расчет рисков</i>	26
7.4. Доступ к ресурсам и авторизация действий в Интеграторе.....	29
7.4.1. <i>Разграничение доступа к ресурсам модели</i>	29
7.4.2. <i>Проектирование рабочих столов</i>	29
7.4.3. <i>Регистрация пользователей в системе и назначение прав доступа</i>	29
7.4.4. <i>Работа пользователя в среде Интегратора</i>	30
7.5. Авторизация действий.....	31
7.6. Навигация по информационному пространству	31
7.7. Оперативная работа с данными единого хранилища	31
7.8. Развитие модели организации на базе единого хранилища данных.....	33

7.9. Обмен данными в едином информационном пространстве	33
7.10. Управления мастер-данными	39
7.10.1. Базовое интегрированное пространство	39
7.10.2. Библиотеки моделей интеграции	39
8. ОПЫТ ПРИМЕНЕНИЯ.....	40

ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ И ТЕРМИНЫ

#Гиперграф:Платформа	Единая информационная среда описания и управления е-моделями произвольной предметной области. Программный продукт обеспечивает создание единой среды эволюционного сетевого и распределенного коллективного визуального проектирования е-модели знаний о той или иной предметной области. Эволюционная е-модель описывается гиперграфом большой размерности с тремя типами специфицированных вершин, семью видами бинарных и тремя видами множественных связей. На её основе реализована автоматическая генерация мультиплатформенного программного кода прикладных программ конечного пользователя в динамически выбираемой требуемой конфигурации.
#Гиперграф:Корпорация	#Гиперграф:Корпорация – программный продукт, основанный на принципах адаптивной динамики изменения программного обеспечения конечного пользователя, позволяющий проектировать адекватные эффективные системы управления для государственных органов, регионов, бюджетных предприятий и организаций (здравоохранение, образование, городское хозяйство и др.), структур крупного и среднего бизнеса. Представляет собой эффективно организованное Единое информационное и функциональное пространство, спроектированное с эволюционным использованием библиотек прототипов, содержащее заполненную базу нормативно-справочной информации и обеспечивающее одновременную работу Единого программного кода с любым количеством различных информационных моделей предприятий (холдингов). #Гиперграф:Корпорация обеспечивает динамичные, адекватные, адаптивные изменения в модели управления и может быть средой построения прогнозов с ситуационным многофакторным анализом.
Цифрогенез	Новое поколение архитектуры прикладного программного обеспечения для создания сложных слабодетерминированных динамических информационных систем управления. Цифрогенез реализует формирование единого адаптивного информационно-функционального пространства автоматически формируемого на основе эволюционно и распределённо изменяемой модели, которая представляет собой гиперграф классов большой размерности. Структура Цифрогенеза включает проектируемую модель и исполняемую модель с автоматическим переходом между ними для обеспечения их адекватности и обратимости. Цифрогенез обеспечивает сопоставимость темпов

	изменения требований к информационной системе с их реализацией в прикладном программном обеспечении. Цифрогенез реализует сбалансированное применение эволюционных и креационных принципов формирования единого информационно-функционального пространства. Цифрогенез обеспечивает длительное и непрерывное функционирование программных систем. Цифрогенез сохраняет состояния моделей в хронологии их изменения.
#Гиперграф:Модель	#Гиперграф:Модель – единая информационная среда эволюционного моделирования и конвергентного коллективного проектирования сложных динамических систем управления для различных проектных групп в сфере разработки, настройки и модернизации прикладных программных продуктов адаптивных систем управления. Основные характеристики #Гиперграф:Модель: Визуальная среда описания модели систем управления реализована на основе единого гиперграфа классов. Гиперграф классов является полугруппой, а также, коммутативным идемпотентным моноидом. Пространство, создаваемое на основе гиперграфа классов является ультраметрическим. Реализованы принципы объектно-ориентированного проектирования (наследование, инкапсуляция, полиморфизм). Обеспечена распределенная коллективная работа с удобными средствами навигации. Обеспечена конвергенция модели при распределенной разработке многообразных систем различными группами разработчиков. Обеспечено автоматическое использование принятых стандартов проектирования. Реализовано непрерывное накопление в модели предметных знаний проектировщиков с обеспечением доступа к историческому, актуальному и прогнозному состояниям. Обеспечено многократное использование спроектированных элементов модели в различных программных решениях. #Гиперграф:Модель обеспечивает открытые технологии проектирования модели. Модель, полученная в #Гиперграф:Модель обеспечивает сетевую рефлекссию, многомерное балансирование. #Гиперграф:Модель обеспечивает процессы самоорганизации в модели. Реализованы функции управления моделями, такие как сравнение e- моделей, мержирование и другие операции с гиперграфом. #Гиперграф:Модель инвариантно к целям, к организационной структуре управления. Обеспечена независимость от поставщика среды.
ПО	Программное обеспечение
РС	Рабочий стол
РД	Руководящий документ

1. ВВЕДЕНИЕ

Информационное пространство современных структур (организаций, предприятий, союзов, блоков, ведомств, стран, родов войск, видов вооружений, негосударственных организаций, органов государственной власти и т.д.) является многокомпонентным. Оно состоит из множества информационных управляющих систем, разработанных в разное время различными разработчиками на разных платформах, с использованием разнообразных подходов и методик. Зачастую возможности обмена данными между отдельными компонентами затруднены или отсутствуют. При этом каждая компонента информационного пространства функционирует и не может быть удалена. Но, с другой стороны, её дальнейшее развитие невозможно в силу различных причин, например, отсутствия связи с разработчиком, морального устаревания инструмента разработки и т.п.



Рисунок 1 – Проблемы интеграции информационных систем

Попытки интеграции разрозненных информационных систем приводят к проблемам, основная причина которых как раз и заключается в том, что информационные системы создавались и создаются для локальных целей, задач и функций различными группами разработчиков по различным технологиям в разное время и являются по существу набором дорогостоящих прикладных программ – «чёрных ящиков».

Отсутствие эффективной интеграции систем управления способствует формированию не транспарентного, не адекватного, не актуального, искаженного информационного пространства. В сотнях информационных систем, как в осколках кривого зеркала не видно целостной картины. Руководители, ответственные за принятие решений, заведомо обречены, хотя и не по собственной воле, на совершение преступных действий, т.к. они, по существу, являются функцией ложного образа, созданного неявной информационной сетью используемых управляющих систем.

Необходимость интеграции функционирующих приложений в единое информационно-функциональное пространство очевидна, проблемы интеграции и семантического взаимодействия – интероперабельности (СИ-Semantic Interoperability) существующих систем остры и актуальны.

В традиционных формулировках проблем интеграции информационных систем акцент делается на видимую и меньшую (хотя и очень трудную) часть задачи: на обмен данными

между информационными системами и полную автоматическую интерпретацию принимающей системой смысла передаваемой информации.

Фрагментарность такого формулирования и такой постановки проблемы интеграции влияет и на поиск подходов, и на отсутствие эффективности у решений, анализ которых приведён в главе 3 «Анализ существующих подходов и решений проблем интеграции».

Говоря об интеграции систем управления уместно разделить все программы, функционирующие в составе информационных систем управления, на два класса:

- инфраструктурное программное обеспечение;
- функциональное программное обеспечение.

К инфраструктурному программному обеспечению относятся программные продукты, для которых характерна абстрактность понятия обрабатываемой информации. Классические примеры – текстовый процессор, система управления базами данных (СУБД), почтовая система и т.д.

К функциональному программному обеспечению относятся программные продукты, обрабатывающие информацию в соответствии с её семантикой, формой, поведением, знанием о реальном предмете управления.

Для инфраструктурных программ с интеграцией всё обстоит в основном благополучно. Действительно, интеграция текстового процессора и почтовой системы настолько логична, бесшовна, глубока и адекватна, что миллионы людей пользуются ими, не задумываясь.

Для функциональных программ складывается другая ситуация в связи со следующими их особенностями:

- знание о предмете управления распределено между структурой данных и алгоритмами её обработки,
- структуры данных содержат неявные предположения о способах их обработки,
- способы обработки данных практически всегда жестко связаны с их хранилищем,
- обработка данных практически всегда основана на функциональной деятельности объекта управления,
- функционирование программы специфицируется пользовательским интерфейсом.

Функциональные информационные системы являются семантическими системами.

Основные типичные проблемы в достижении семантической интеграции информационных систем:

- многократное избыточное описание предметов и процессов предметной области,
- концептуальная несовместимость, нецелостность, противоречивость и т.п. описания общих частей предметной области и самих данных в рамках разных систем,
- необходимость учета знания об обработке данных, распределенного между их структурой и процедурами их обработки, при формировании единого информационного хранилища,

- наличие в системах собственных хранилищ данных исключает возможность простой потоковой обработки,
- распределенная разработка сложных систем, так разработка Системы 1 одной крупной части предметной области неизбежно входит в противоречие с одновременной, но отдельной разработкой Системы 2 другой крупной части предметной области, что в дальнейшем усиливается разностью в кодировании программ,
- обеспечение взаимодействия систем между собой становится еще одним «видом деятельности»,
- ограничения скорости модификации систем в ответ на высокую динамику изменений объектов и процессов управления,
- проблемы интероперабельности разноплатформенных данных и интерфейсов унаследованных и вновь разрабатываемых программных комплексов,
- отсутствие комплексной безопасности фрагментарных систем и межсистемного информационного пространства,
- низкая надежность сложных программных комплексов,
- высокие издержки на развитие, модернизацию, сопровождение и эксплуатацию,
- и другие.

Проблема семантической интеграции становится схожей с проблемами комбинаторных задач. В обеспечении взаимодействий каждой конкретной пары программных систем может быть достигнут (и часто достигается) успех. Однако этот подход никак не масштабируется на комплексную систему, и обеспечение динамической семантической интеграции по сложности начинает превышать суммарную сложность разработки отдельных программных компонент.

В данном документе описывается технология #Гиперграф:Интегратор – инновационное решение, обеспечивающее динамическую семантическую интеграцию информационных систем.

#Гиперграф:Интегратор – технология интеграции разнородных прикладных программ, в т.ч. унаследованных, для создания единого информационного и функционального пространства над пространством гетерогенных систем. #Гиперграф:Интегратор обеспечивает технологию описания проекций разнородных систем в единой интеграционной модели и работу с данными других систем в режиме on-line.

#Гиперграф:Интегратор – обеспечивает миграцию от унаследованных программ в новые технологии и архитектуры путём использования технологии адаптивных проекций **TableX**.

2. ЦЕЛИ ИНТЕГРАЦИИ

Решение проблемы интеграции приложений должно обеспечить:

- Работу всех служб организации в едином информационном пространстве и реальном масштабе времени.
- Оперативное получение сводных, консолидированных и аналитических отчетов.
- Уменьшение избыточности информации в разных системах, повышение достоверности на момент выполнения запросов.

- Значительное сокращение затрат на поддержку системы за счет исключения ручного переноса данных из системы в систему, стандартизации обмена между всеми системами и работы в режиме реального времени.
- Возможность развивать автоматизацию последовательно, следуя принципам эволюционного развития систем и благодаря сокращению числа взаимосвязей.
- Возможность сократить затраты на обновление компонент системы и интеграцию новой компоненты.
- Выбор оптимальной платформы реализации решения независимо от поставщиков аппаратных и программных систем, от "пристрастий" групп разработчиков.

Тем самым, решение проблемы интеграции приложений должно способствовать получению реальной (адекватной) картины деятельности организации и позволить качественно планировать направления этой деятельности и финансовые результаты.

Интеграция приложений должна быть направлена на повышение эффективности бизнеса, на сокращение временных и финансовых затрат.

3. АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ И РЕШЕНИЙ ПРОБЛЕМ ИНТЕГРАЦИИ

Анализ используемых подходов в решении проблем семантической интеграции (далее по тексту – семантическая интероперабельность) показал, что можно выделить три основных направления разработок за последние 20-30 лет:

- интеграция межсистемных сообщений;
- интеграция через дополнительные интегрирующие системы;
- интеграция в архитектуре SOA.

3.1. Направление 1: интеграция межсистемных сообщений

Интероперабельность межсистемных сообщений – работа над форматами, структурой, синтаксисом, семантикой и т.п. сообщений, циркулирующих между функциональными информационными системами, представленными «черными ящиками», т.е. основной акцент делается на семантическую прозрачность директив, приказов, приказаний, указаний, планов, отчетов, донесений, докладов и т.п. (Рисунок 2).

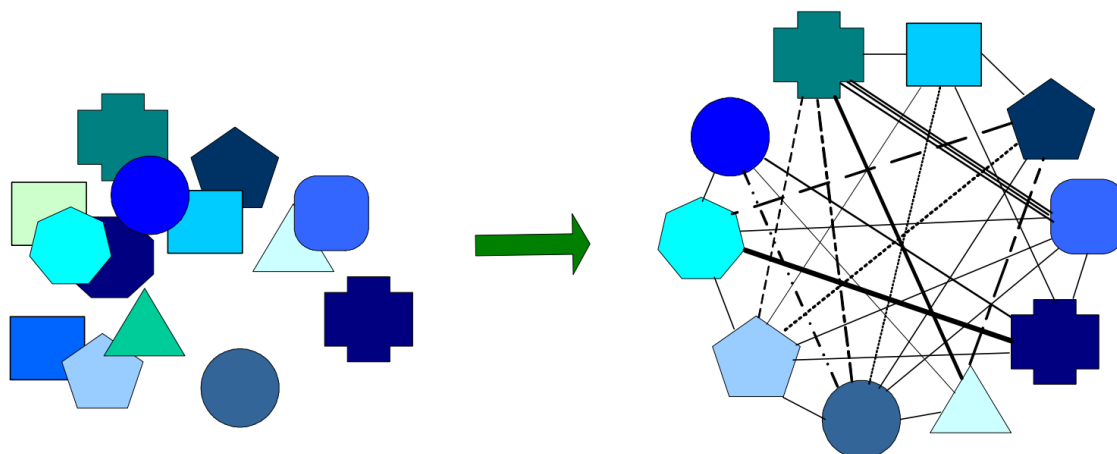


Рисунок 2 – Интероперабельность межсистемных сообщений

3.2. Направление 2: интеграция через дополнительные интегрирующие системы

Интероперабельность через дополнительные интегрирующие системы – работа над дополнительным набором новой совокупности программных систем, которые могут включать адаптеры, хабы, среды передачи сообщений, единое хранилище общих данных, бизнес-моделлеры, онтологии баз данных/знаний и т.п. (Рисунок 3).

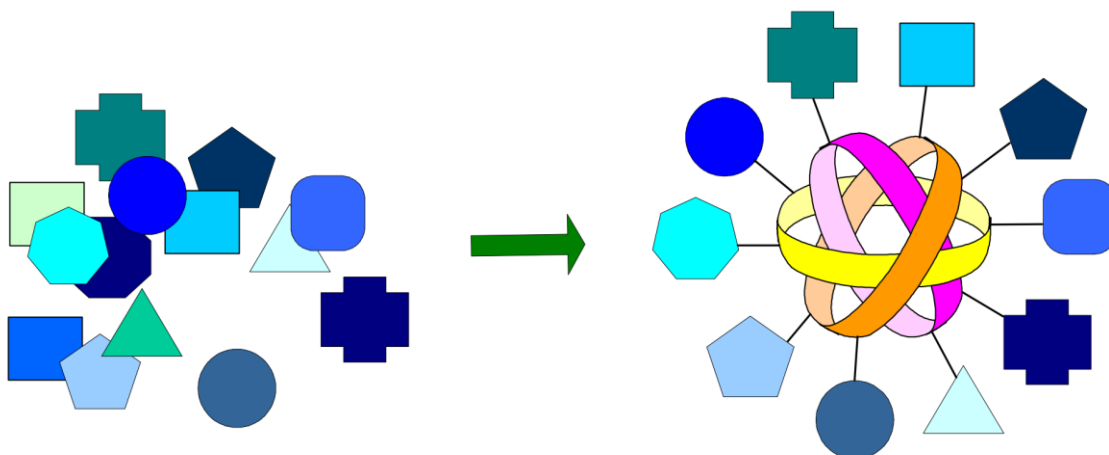


Рисунок 3 – Интероперабельность через дополнительные интегрирующие системы.

3.3. Направление 3: интеграция в архитектуре SOA

Интероперабельность в архитектуре SOA – работа над новыми архитектурами функционального программного обеспечения, которые должны обеспечивать создание единого информационного пространства и помочь в решении проблем противоречивости, избыточности, несопоставимости и т.п. данных различных функциональных систем. Например, предлагается использовать SOA-сервис-ориентированную архитектуру. Однако: при использовании новых архитектур создается НОВОЕ программное обеспечение с

ТРАДИЦИОННЫМ жизненным циклом информационных систем, и остается нерешенной проблема миграции от унаследованных информационных систем, а также проблема семантической интероперабельности систем в новой архитектуре разных производителей (Рисунок 4).

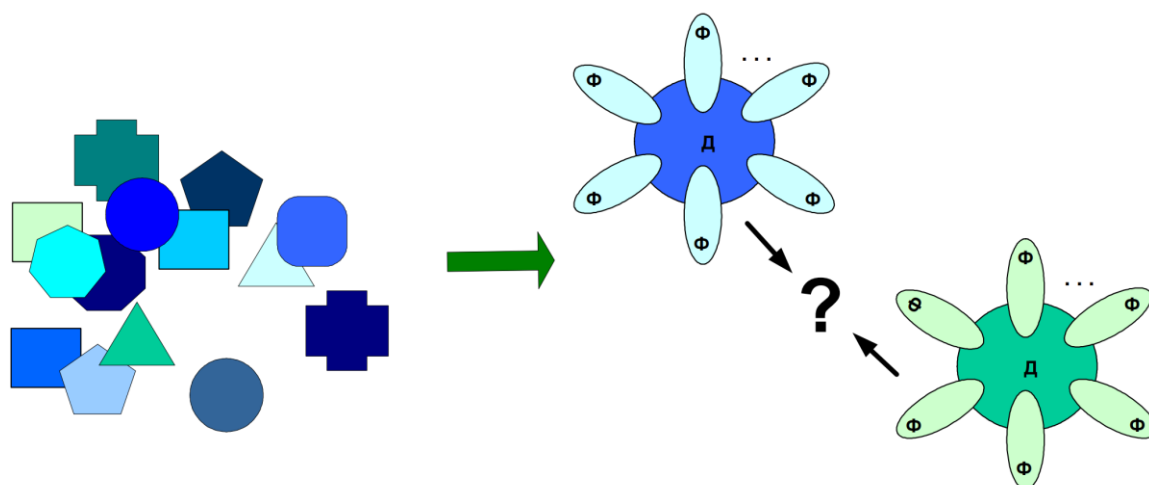


Рисунок 4 – Интероперабельность в новой архитектуре SOA

По данным направления IT-лидерами: IBM, Microsoft, ORACLE, SAP и другими сделаны многомиллиардные вложения, выпущены и рекламируются программные продукты.

Но цель, эффективная семантическая интеграция информационных систем, не достигнута и вышеперечисленными методами достигнутой быть не может по причинам, описанным ниже.

Все представленные решения не «живут» в динамике изменения требований к объединяемым информационным системам, не поддерживают жизненный цикл систем, требуют колоссальных затрат ресурсов на ещё один вид деятельности – поддержку интероперабельности, не учитывают фактор времени, не эффективны при количестве систем больше 3, не надежны, не решают комплексно вопросы безопасности и многое другое.

Кроме того, трудно представить, что проблему семантической интероперабельности N систем можно решить добавив к ним ещё M систем, обеспечивающих их объединение.

Применение онтологий добавит к сформулированным и понятым проблемам семантической интероперабельности систем проблемы адаптивной гармонизации онтологий.

Пропаганда перехода на принципы открытых стандартов функциональных информационных систем на уровне миллиардов строк текстов программ – так же бесперспективна. Открытый стандарт в онтологиях концептуально не может обеспечивать on-line адекватность функциональных информационных систем и их семантических описаний.

Дополнение и без того длительного затратного жизненного цикла информационных систем новым шагом «онтологической поддержки» увеличивает растущую пропасть между

информационными системами управления и динамично изменяемыми объектами и процессами управления.

Вывод один: «лечение» не адекватно «диагнозу», поэтому, применение «зеленки» не помогает «больному раком».

Однако требование формирования онтологической целостности мы считаем ключевым для систем управления.

4. ГИПЕРГРАФ-ПАРАДИГМА РЕШЕНИЯ ПРОБЛЕМЫ ИНТЕГРАЦИИ

Предлагается изменить саму парадигму решения проблемы семантической интеграции и перейти от обеспечения онтологической транспарентности обмена сообщениями к главному: созданию целостной модели информационно-функционального управленческого пространства для коллективного решения задач и достижения целей – Системы Глобальной Рациональности.

То есть, для решения семантической интеграции, необходимо осуществить переход от ИНТЕР-операбельности информационных систем – к ТРАНС-операбельности (в том числе, транснациональной) объектов управления.

Проблема эффективной интероперабельности трех и более функциональных информационных систем управления не разрешима.

К семантическому информационно-функциональному управленческому пространству были сформулированы следующие требования:

- Непрерывная трансформация «на лету», адаптивность;
- Адекватность;
- Эволюционность;
- Обеспечение необычных, не прямых, ассиметричных новаторских управленческих воздействий;
- Обеспечение ведения иррегулярных действий и бизнес-процессов;
- Единое экспертное пространство;
- Извлечение и применение знаний в режиме реального времени;
- Организация единого информационно-функционального пространства;
- Обеспечение многоцелевого сбалансированного использования объединенных ресурсов;
- Единая модель сетецентрического управления;
- и ряд других.

Фонд развития технологий «Инженеры устройства мира» в целях решения задачи достижения эффективной семантической интероперабельности информационных систем предложила новый подход и создала инновационную технологию Цифрогенез (Рисунок 5), при этом, решаются две задачи:

0. Создать «идеальную» технологию Будущего;
1. Создать технологию Эволюционной Миграции.

В решении данных задач разработаны и промышленно используются следующие технологии:

- Создание «идеальной» технологии Будущего – система адаптивного бесшовного непрерывного семантически целостного графо-центричного управления глобальными структурами.

#Гиперграф:Язык – визуальный язык описания единого «генезиса» информационных систем; #Гиперграф:Моделлер – создание единой адаптивной семантической Модели Знаний – «ДНК» информационных систем;

#Гиперграф:РоботПО – технология автоматического программирования адаптивных систем управления на основе Модели Знаний – гиперграфа классов описанного в единой виртуальной среде Эволюционного Моделирования;

#Гиперграф:Модель – программный продукт, обеспечивающий динамическое произвольное формирование конфигурации информационной системы конечного использования, адаптированной к должностным инструкциям, полномочиям, компетенциям, ролевой модели пользователя, и реализующий регламентированный доступ к данным и функциям.

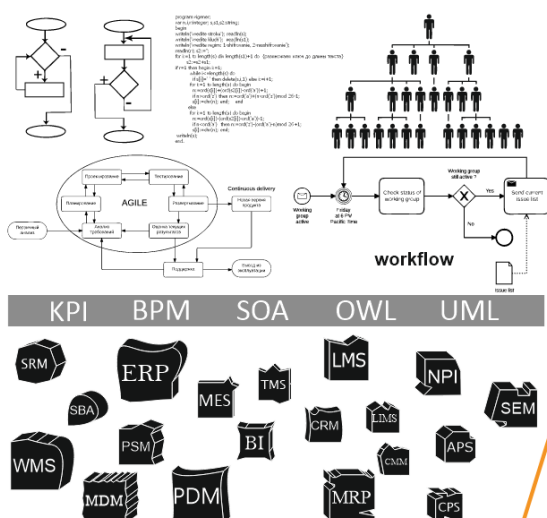
#Гиперграф:Знания – формирование целостных Баз Знаний в единых правилах Модели Знаний.

- Создание технологии Эволюционной Миграции – система перехода в цифрогенез с постепенным «безболезненным» замещением Настоящего, в т.ч. Унаследованного на Будущее.

#Гиперграф:Интегратор – эволюционное проецирование унаследованных информационных систем в «ДНК» Модели Знаний на новом языке G3L и постепенное их замещение, при этом замещение происходит тем быстрее, чем больше объединяется унаследованных систем и динамичней идет изменение требований к ним.

#Гиперграф:Платформа – единая адаптивная динамическая система управления общими условно-постоянными данными, в т.ч. нормативно-справочной информацией, являющейся частью базы знаний.

ИЕРАРХИЧЕСКИЕ СИСТЕМЫ УПРАВЛЕНИЯ,
АРХИТЕКТУРА, ПРОГРАММА, ЖИЗНЕННЫЙ ЦИКЛ,
МОДУЛЬ, СЕРВИС, ПОРЯДОК ДЕЙСТВИЙ, ИНСТРУКЦИЯ,
АЛГОРИТМ, МЕХАНИЦИЗМ, РЕДУКЦИОНИЗМ,..



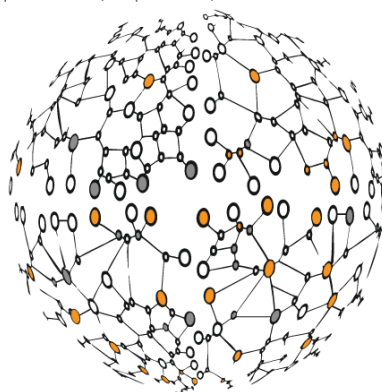
ХАОТИЧЕСКИ РАСТУЩАЯ СЛОЖНОСТЬ

ПРЕОДОЛЕНИЕ БАРЬЕРА СЛОЖНОСТИ

НОВЫЙ ТЕХНОЛОГИЧЕСКИЙ УКЛАД
ЭНЕРГЕТИЧЕСКИЙ ПЕРЕХОД

ЦИФРОГЕНЕЗ

ГРАФО-ЦЕНТРИЧНАЯ ПЛАТФОРМА
ГОСУДАРСТВЕННОГО, СОЦИАЛЬНОГО, КОРПОРАТИВНОГО УПРАВЛЕНИЯ



НОВОЕ ЗНАНИЕ
КОНТИНУИТИВНО

ОРГАНИЗОВАННАЯ СЛОЖНОСТЬ

GIPERGRAF.RU

Рисунок 5 – Эволюционная миграция в *цифрогенез*.

В парадигме цифрогенеза создана карта взаимоувязанности основных понятий: информация, данные, знания, модель знаний, базы знаний, управление знаниями, семантика, онтология, когнитивность, гносеология, и ряд других.

Цифрогенез успешно реализует конвергентную распределенную параллельную разработку функциональных информационных систем с самоорганизацией и гармонизацией единой адаптивной Модели Знаний. Принципы открытых стандартов в цифрогенезе применены на уровне Модели Знаний – гиперграфа классов Хохловой (Рисунок 8).

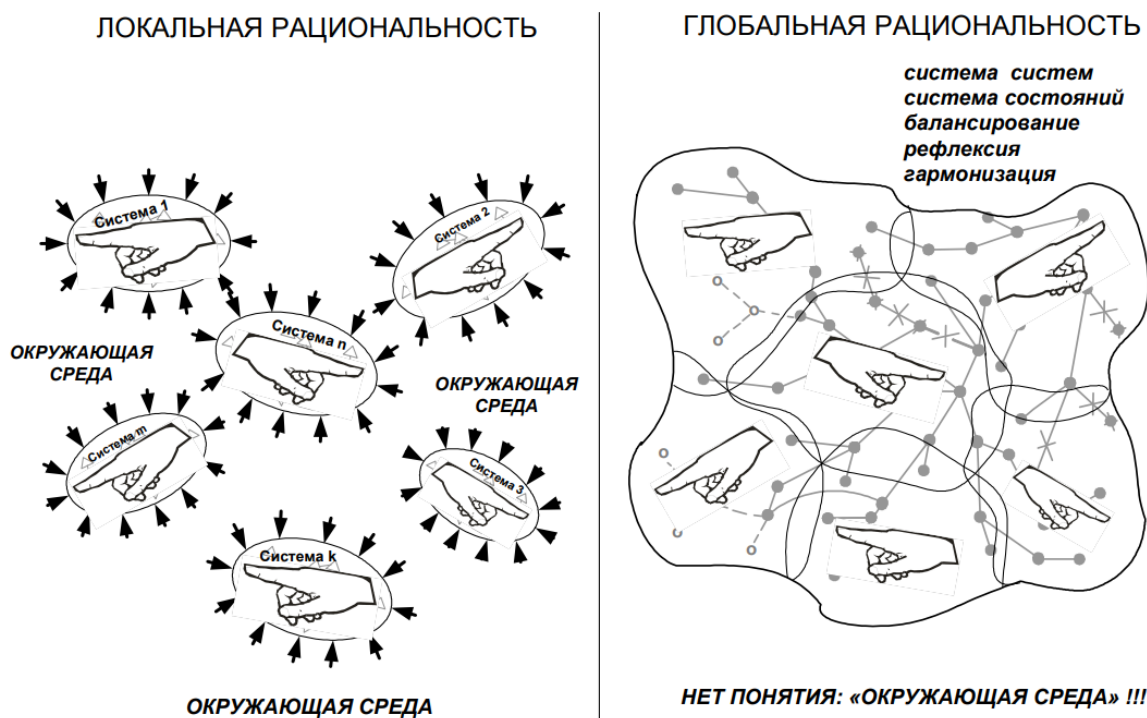


Рисунок 6 – Переход от ИНТЕР-операбельности к ТРАНС-операбельности.

Цифрогенез обеспечивает реализацию новой парадигмы Системы Принятия Сбалансированных Решений:

Целеполагание. Цифрогенез формирует единое пространство описания динамического множества различных, в том числе противоречивых, целей по обеспечению управления и противодействия вызовам

Моделирование. Эволюционное коллективное формирование адекватной адаптивной информационной модели объектов и процессов управления. Цифрогенез позволяет постепенно наращивать генезис модели, реально используемой уже на ранней стадии её появления.

Мониторинг. Целостное глобальное отражение транзакционных изменений системы состояний объектов и процессов управления, в целях обеспечения достаточной транспарентности, гарантирующей, в том числе предотвращение ошибочного восприятия действий сторон.

Баланс. Многомерное динамическое балансирование состояний множества объектов управления с выявлением веса изменяемого состояния неопределенности (дисбаланса).

Подготовка решений. Цифрогенез позволяет целостно рассматривать последствия принимаемых решений и оценить их совокупную «стоимость». Обеспечение в единой сетевой виртуальной модели целостной картины on-line рефлексии на производимые воздействия на элементы системы с анализом процессов затухания и резонанса в цепочках взаимосвязей.

Анализ. Прогноз. Гармонизация. Цифрогенез позволяет создать единое информационно-функциональное аналитическое пространство исторической, актуальной и прогнозной ситуации для гармонизации средств и методов непрерывного управления в том числе и в кризисных ситуациях.

5. ОБЗОР МЕТОДОВ ИНТЕГРАЦИИ

Для решения проблемы интеграции информационных систем, как правило, используются следующие подходы:

- метод screen scraping – интеграция на уровне пользовательских интерфейсов;
- метод хранилищ данных – интеграция на уровне данных;
- метод EAI – интеграции на уровне приложений;
- метод Web-сервисов;
- технология адаптивных проекций TableX.

5.1. Интеграция на уровне пользовательских интерфейсов

Основная идея метода screen scraping – взаимодействие приложений через пользовательский интерфейс. Метод достаточно экономичен, но в нём отсутствуют средства масштабирования, и он неэффективен в распределенных сложных средах.

Технология адаптивных проекций TableX (см. 7.1стр. 20), на которой базируется #Гиперграф:Интегратор , использует универсальность и экономичность метода screen scraping, расширяя и дополняя возможности данного метода.

5.2. Интеграция на уровне данных

Метод хранилищ данных подразумевает поддержку данных в специальных хранилищах, не зависящих от бизнес-логики, их «породившей». Доступ к хранилищам могут получать различные приложения.

Данный метод является основным способом оптимизации взаимосвязей между различными системами, благодаря успеху продуктов, созданных на основе реляционных баз данных и сопутствующих стандартов (таких, как SQL и ODBC).

Основные проблемы метода:

- использование общей модели данных для всего набора приложений значительно ограничивает гибкость;

- интеграция на уровне данных сама по себе не предлагает средств связи с традиционными приложениями. Для связи между слоями данных требуются специальные коннекторы, что влечет за собой дополнительные расходы, организацию внесения изменений и зависимость от поставщика.

Данный метод полностью реализуется в #Гиперграф:Интегратор Интеграторе средствами технологии адаптивных проекций TableX (см. 7.1).

5.3. Интеграция на уровне приложений

Метод EAI (Enterprise Application Integration) подразумевает использование исполняемого кода приложений, а не внутренних данных, т.е. одно приложение может использовать функциональность, присутствующую в другом, независимо от базовой платформы.

Для реализации подхода используются традиционные библиотеки функций и API, созданные для объединения программ на одной платформе, а также межплатформенные инструменты, такие как RPC и ORB.

Суть метода сводится к тому, что EAI интегрирует традиционные и пакетные приложения, не внося каких-либо модификаций, что гарантирует отсутствие ошибок в критически важных приложениях.

Недостатком этого метода является сложность осуществления. EAI-проекты, как правило, оказываются дорогостоящими и длительными, требуют значительных экспертных знаний.

Подключение внешних приложений в программных продуктах G3 производится средствами API. В G3 ERP реализована интеграция с командной строкой операционной системы Windows, что позволяет работать с практически любыми внешними приложениями. Опираясь на данный метод, G3 ERP работает с геоинформационными-системами (ГИС) такими, как MapInfo и MapObjects. Построение графиков и диаграмм в G3 ERP производится также внешними приложениями, например, MS GRAPH из состава MS Excel (при необходимости можно подключить любое другое аналогичное средство).

5.4. Метод Web-сервисов

Методы EAI и Web-сервисов, фактически, являются разными реализациями одного подхода.

Метод Web-сервисов основан на использовании объектно-ориентированных технологий для заключения данных и программных элементов в некую «оболочку», чтобы к ним могли получать доступ различные web-приложения.

Отличие использования в проектах web-сервисов от EAI заключается в том, что Web-сервисы предоставляют стандартизированные способы работы с интеграцией, а EAI-технологии всегда зависели от одного-двух поставщиков или разрабатывались для конкретных продуктов.

На основе метода Web-сервисов в последние годы возникло несколько стандартов, в основе которых лежит возможность работать и воспринимать сообщения в формате XML.

Работа с текстовыми данными и данными в формате XML в #Гиперграф:Интегратор Интеграторе производится средствами технологии адаптивных проекций TableX (см. 7.1).

Кроме того, в G3 ERP работа с удаленного клиентского места может выполняться через Web-интерфейс (см. Руководство пользователя глава 7 «Работа пользователя через Web-интерфейс»).

6. Основные принципы #Гиперграф:Интегратор

Интегратор аккумулировал лучшие из методик, приведенных выше, и предлагает комплексный подход: интеграция данных, пользовательских интерфейсов, приложений с использованием Web-сервисов.

Интегратор – уникальное средство объединения разнородных прикладных программных комплексов, используемых в организации (предприятии, холдинге), для создания единого информационного и функционального пространства над пространством гетерогенных систем.

На рисунке (см. Рисунок 7) приведен пример решения задачи интеграции в организации, использующей унаследованные программные комплексы, которые:

- реализованы на различных платформах;
- работают с данными различной организации и различных форматов;
- выполнены и поставлены различными разработчиками;
- функционируют под различными операционными системами;
- и т.п.

Интегратор базируется на принципах:

- открытых технологий;
- объектно-ориентированного проектирования модели единого информационного хранилища;
- коллективности проектирования модели единого информационного хранилища и работы по аналитической обработке данных;
- адекватности модели единого информационного хранилища;
- эволюционного развития модели управления знаниями.

Интегратор не только способен повысить эффективность управления ресурсами организации (предприятия, холдинга) сегодняшнего дня, но стать основой дальнейшего эволюционного развития всей её информационно-функциональной деятельности.

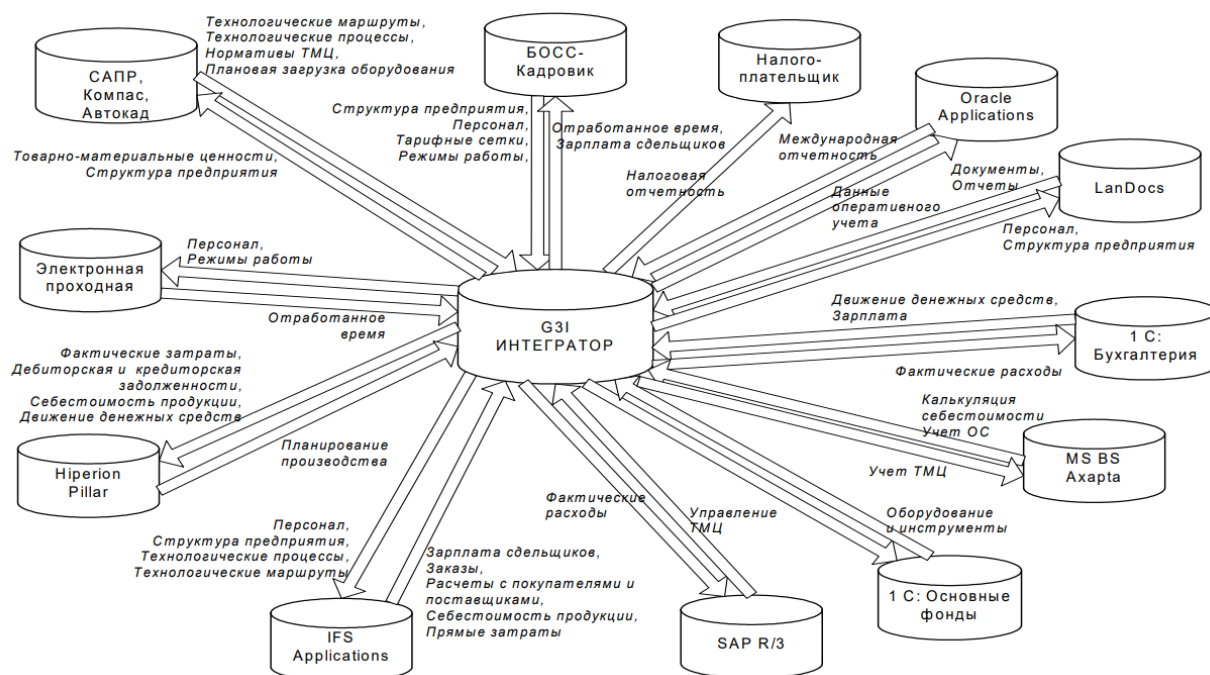


Рисунок 7 – Пример использования Интегратора для решения задачи создания единого информационного и функционального пространства организации

7. СОСТАВ #Гиперграф:Интегратор

Интегратор включает:

- Создание единого информационного и функционального пространства
 - – Аттестация автоматизированных рабочих мест
 - – Создание единого информационного хранилища данных
 - – Синхронизация внешних хранилищ данных
- Анализ и контроль данных единого информационного хранилища
 - – Автодокументирование единого информационного хранилища данных
 - – Анализ модели единого информационного хранилища
 - – Анализ качества данных. Расчет рисков
- Доступ к ресурсам и авторизация действий
 - – Разграничение доступа к ресурсам модели
 - – Авторизация действий
 - – Навигация по информационному пространству
- Оперативная работа с данными единого хранилища
 - – Проектирование интерфейсов для работы с единым хранилищем данных
 - – Оперативная работа с данными единого хранилища, их анализ
- Развитие модели организации на базе единого хранилища данных

– Проектирование новых задач системы аналитической обработки данных на базе единого хранилища

- Обмен данными между компонентами единого информационного пространства
 - – Импорт и экспорт данных
 - – Репликация данных
 - – Работа с буферными базами данных

- – Разработка и исполнение регламентов обмена данными
- Дополнительные сервисы
 - – Базовое интегрирующее пространство
 - – Библиотеки моделей интеграции

Проектирование в среде Интегратора выполняется с использованием встроенных Case-средств.

7.1. Технология адаптивных проекций TableX

В Интеграторе #Гиперграф:Интегратор реализован уникальный метод – технология адаптивных проекций TableX, позволяющий работать с данными внешних систем в режиме «он лайн» без передачи данных (экспорт, импорт, репликация). Данные внешних систем могут лежать в различных форматах (Oracle, MS SQL, СУБД формата dbf , MS Access и др.)

#Гиперграф:Интегратор позволяет строить произвольные аналитические запросы к данным внешних систем, эффективно визуализировать информацию, сопоставлять, использовать в аналитических расчетах, строить прогнозные модели и др.

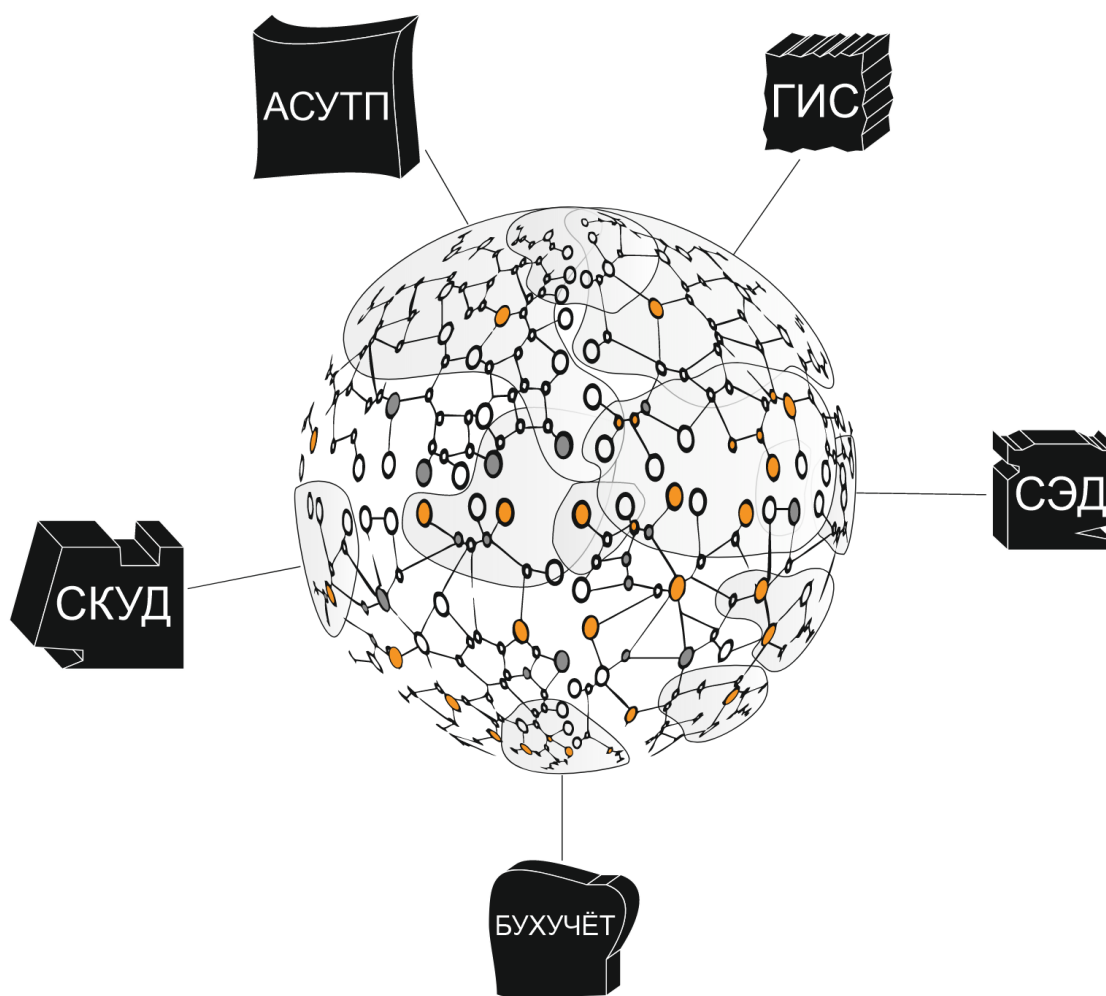
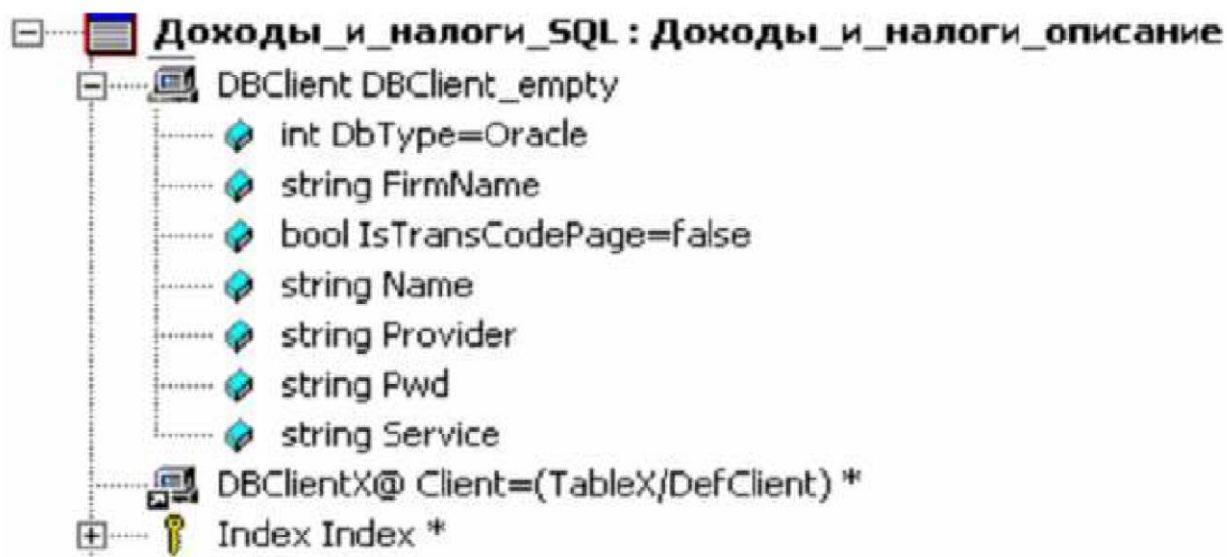


Рисунок 8 – Реализация интеграции методом проекции в #Гиперграф:Интегратор

Базовый класс TableX применяется для воссоздания в Интеграторе метаописания

структуры данных унаследованных программных комплексов. Параметры подключения к внешней базе данных в технологии адаптивных проекций описываются через Базовый класс TableX:



Описание структуры внешних таблиц в базовом классе TableX:



При высокой динамике изменений предметной области применение технологии адаптивных проекций приводит к тому, что отказ от морально устаревших унаследованных систем, программное обеспечение которых не поддерживается производителями, и переход к новой СИСТЕМЕ происходит наиболее быстро и безболезненно.

Единая модель, полученная при использовании технологии адаптивных проекций является гибкой, прозрачной, масштабируемой, эволюционирующей, интегрирующей общие знания.

7.2. Создание единого информационно-функционального пространства

7.2.1. Аттестация интегрируемого ПО

До начала выполнения работ по интеграции разрозненных программных комплексов необходимо провести аттестацию каждого подготовленного рабочего места, с целью определения:

- – перечня решаемых задач (функциональных обязанностей);
- – набора программного обеспечения;
- – состава вычислительной техники;
- – параметров сетевого подключения;
- – ит.п.

Для решения этой задачи предназначен специальный рабочий стол Интегратора «Аттестация автоматизированных рабочих мест» (см. Рисунок 9).

Рисунок 9 – Пример регистрации программного обеспечения, используемого на рабочем месте

На основании аттестации каждого автоматизированного рабочего места и определения функционального взаимодействия этих мест можно составить реальную схему необходимой и возможной интеграции программных продуктов в организации в целом.

7.2.2. Создание единого информационного хранилища данных

Интегратор создает единое хранилище данных – единую модель, объединяющую унаследованные программные комплексы.

Средствами Интегратора в едином информационном и функциональном пространстве – модели организации создаются проекции информационных сред унаследованных программных комплексов. На рисунке (см. Рисунок 10) показан принцип проектирования с использованием технологии адаптивных проекций TableX.

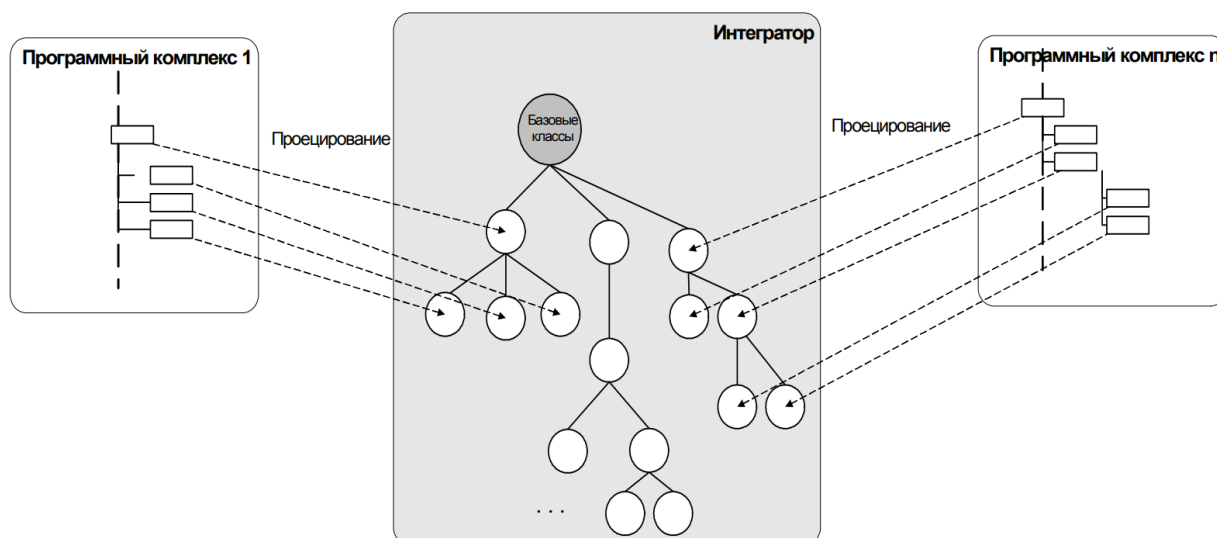


Рисунок 10 – Принцип проектирования в Интеграторе проекций таблиц унаследованных программных комплексов

В классе TableX заложена возможность создания адаптивных проекций и описания параметров подключения к внешней базе данных.

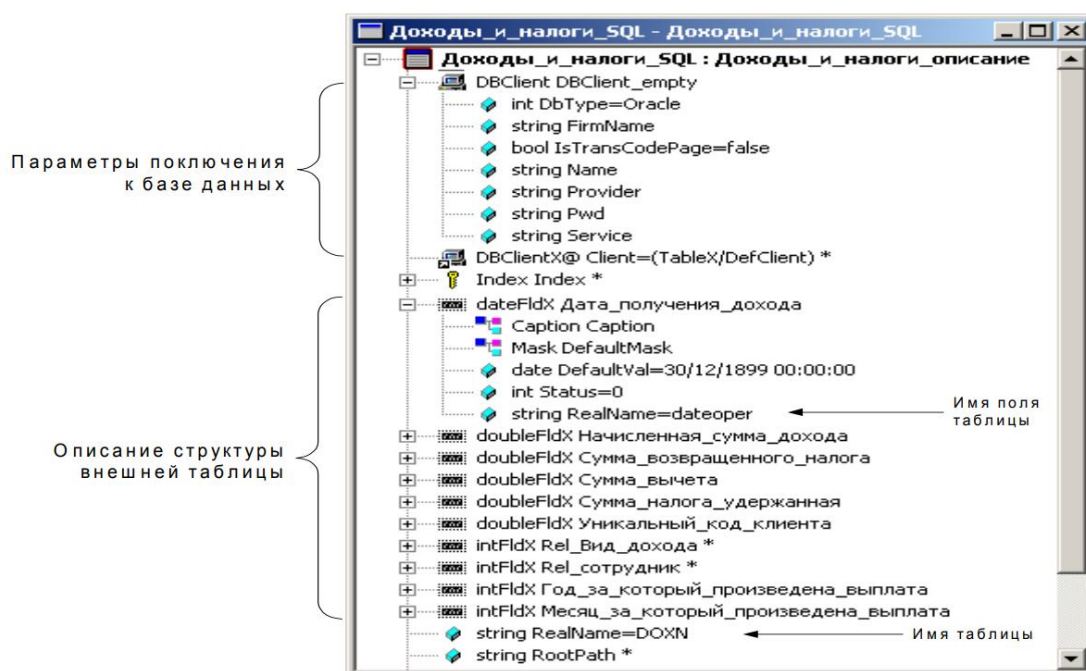


Рисунок 11 – Пример описания структуры внешней таблицы средствами TableX

Для обмена информацией между базами данных отдельных компонент единого информационного пространства в Интеграторе используются различные механизмы, представленные на рисунке ниже.

7.2.3. Синхронизация внешних хранилищ данных

Интегратор может использоваться как средство синхронизации хранилищ унаследованных программных комплексов на уровне идентичности данных. На рисунке (см. Рисунок 13) представлена схема синхронизации хранилищ унаследованных программных комплексов в едином информационном пространстве.

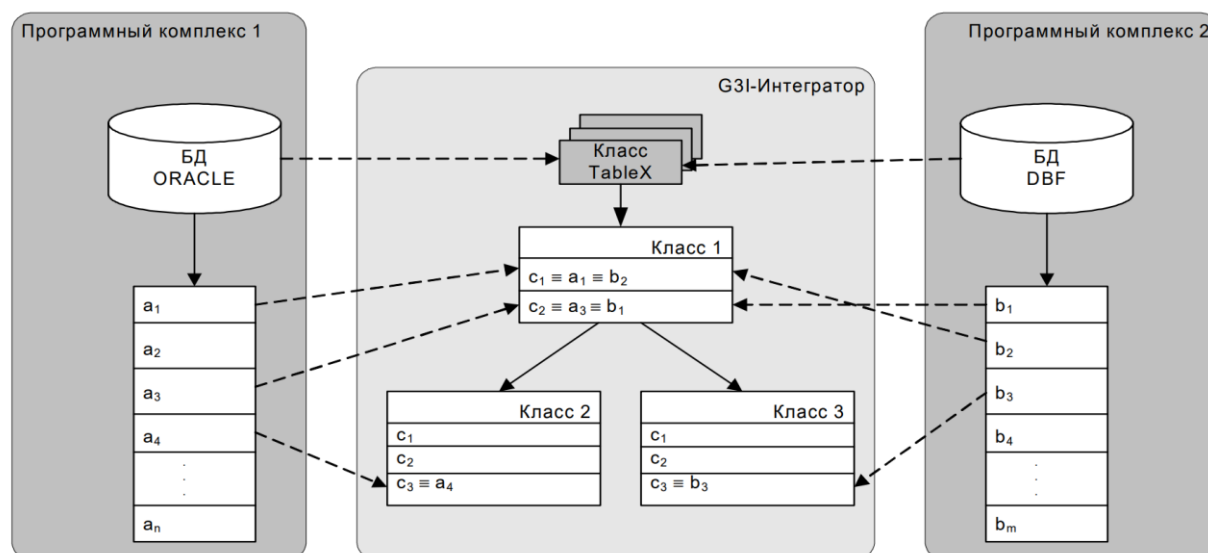


Рисунок 13 – Синхронизация хранилищ унаследованных программных комплексов в едином информационном пространстве

7.3. Анализ и контроль данных единого информационного хранилища

7.3.1. Автодокументирование единого информационного хранилища

Механизм автодокументирования Интегратора позволяет просматривать и анализировать структуры хранилищ данных унаследованных программных комплексов, отразив их (спроецировав) в едином информационном пространстве.

Автодокументирование позволяет получить:

- – общий перечень полей всех таблиц единого информационного хранилища;
- – состав каждой таблицы хранилища данных, визуализируя отношение Таблица – Поле;
- – реляционные связи таблиц в каждой базе данных, визуализируя отношение Таблица – Таблица;
- – повторяемость идентичных данных, хранимых в таблицах различных баз данных, визуализируя отношение БД – БД.

Проанализировав результаты автодокументирования можно:

- – сократить избыточность данных, полей, связей, таблиц;
- – повысить достоверность данных;
- – обеспечить сопоставимость данных;

В системе G3-ERP для реализации функционала автодокументирования создан специальный рабочий стол. Работа пользователя со средствами автодокументирования подробно описана в документе «Автодокументирование исполнимой модели» (077.Ц.001-07 36 064 036).

7.3.2. Анализ модели единого информационного хранилища

В Интеграторе есть механизмы просмотра, анализа и корректировки компонент модели единого информационного хранилища – набор специализированных редакторов:

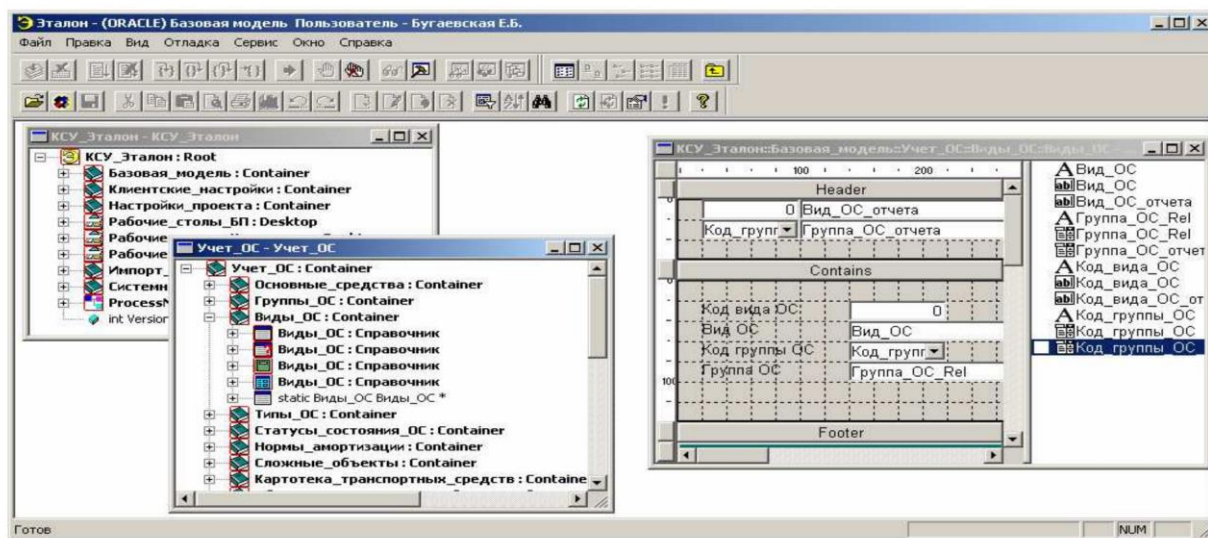


Рисунок 14 – Пример интерфейсов специализированных редакторов Интегратора

Объем и характер информации, отображаемой в окнах редакторов Интегратора, настраивается администратором или аналитиком, непосредственно проектирующим задачу.

7.3.3. Анализ качества данных. Расчет рисков

В Интеграторе, используя специальные базовые классы, можно спроектировать схемы проверки данных, хранящихся в таблицах интегрированной базы, на соблюдение условий:

- – полноты (заполненность указанных полей по условию И или ИЛИ);
- – уникальности значений по полям или комбинациям полей;
- – ссылочной целостности записей;
- – соответствие условиям.

Для упрощения проектирования схемы проверки корректности введенных данных разработан специальный редактор схем контроля данных. С его помощью можно спроектировать схему проверки и выполнить проверку данных следующих типов:

- Контроль на соответствие заданным условиям. Входные данные не должны быть противоречивыми, т.е. должны находиться в соответствующем диапазоне значений, удовлетворять условиям контроля сумм.
- Проверка на целостность данных. Контроль целостности подразумевает, что передаваемые данные для отдельных реквизитов. Для этого каждому реквизиту устанавливается в соответствие поле в справочнике, затем ищется запись в справочнике и в случае положительного результата поиска устанавливается связь на этот справочник, если установлено поле связи. В случае отрицательного результата поиска запись заносится в отчет как не удовлетворяющая критериям проверки.
- Контроль полноты данных. Для каждого поля (если необходимо) устанавливается проверка на наличие данных, т.е. данное поле обязательно должно содержать

информацию. Для каждой таблицы устанавливается количество необходимых записей, которые должны использоваться в дальнейшем (например, при выполнении определенного расчета).

– Контроль уникальности данных. Для каждого поля или группы полей (если необходимо) таблиц устанавливается признак проверки на уникальность. Уникальность по полю таблицы означает, что данные, хранящиеся в этом поле, не могут повторяться.

Редактор позволяет достаточно просто задать необходимые параметры проверки: создать классы и объекты проверки, определить для проверяемой таблицы контролируемые поля и условия, которым должны соответствовать данные. Таблицы, поля которых подлежат проверке, определяет системный аналитик при проектировании задачи и, в дальнейшем, при настройке контроля входных данных.

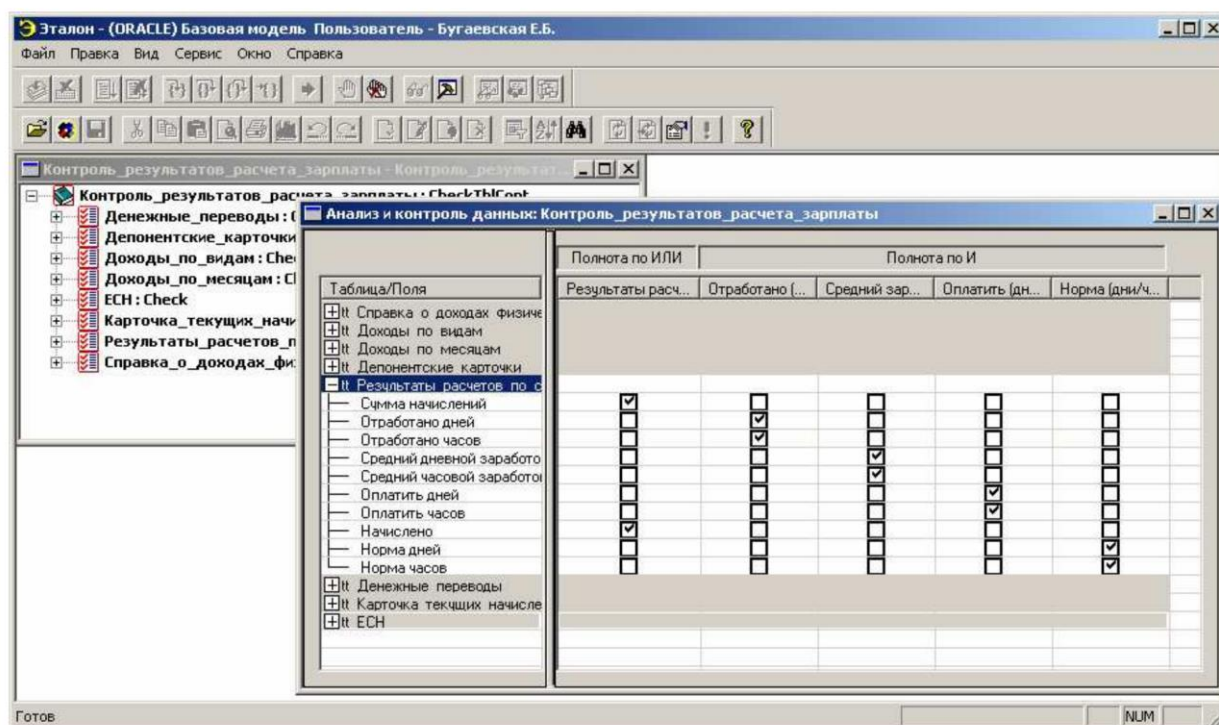


Рисунок 15 – Пример задания параметров проверки качества результатов расчета

При выполнении проверки пользователю предоставляется возможность (см. Рисунок 16):

- задать параметры проверки;
- получить протокол операции;
- получить перечень отбракованных записей в виде отчета (с возможностью просмотра и сохранения в файл);
- оценить риск использования информации без устранения обнаруженных ошибок.

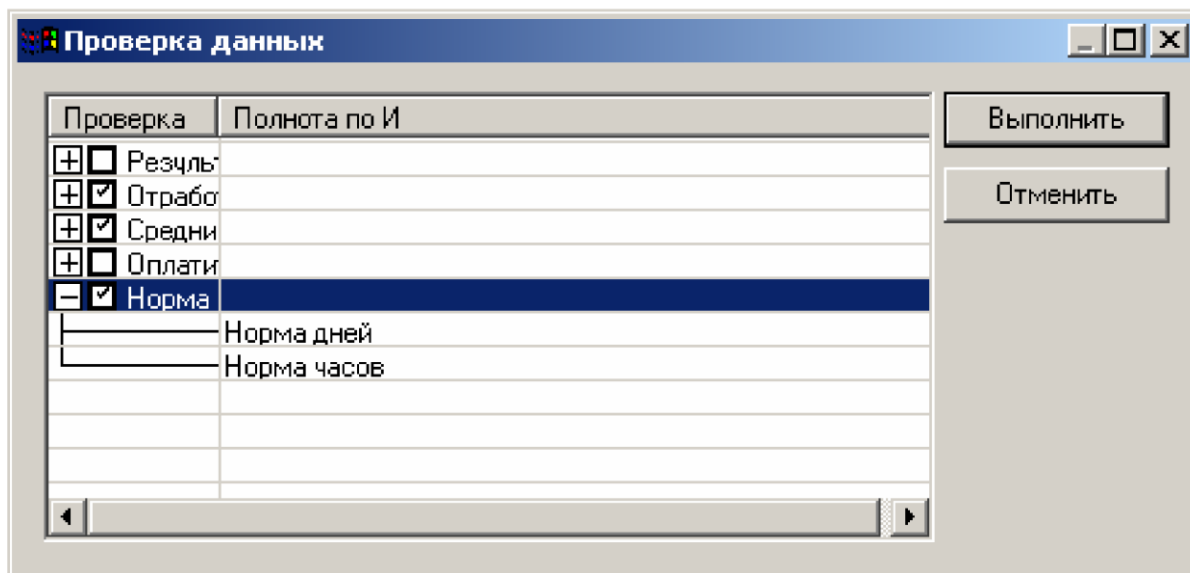


Рисунок 16 – Интерфейс выбора условий проверки качества данных

В результате каждой проверки формируется отчет, содержащий информацию о данных, не удовлетворивших условиям проверки (см. Рисунок 17).

Полнота 'И' по полям: Отработано дней и Отработано часов - Отчет

	Отработано дней	Отработано часов	Сумма начислений	Средний дневной заработок	Средний часовой заработок	Оплатить дней	Оплатить часов	Начи
▶	0,00	0,00	0,00	0,00	0,00	19,00	151,00	
	0,00	0,00	0,00	0,00	0,00	10,00	0,00	
	0,00	0,00	0,00	0,00	0,00	10,00	0,00	

Полнота 'И' по полям: Средний дневной заработок и Средний часовой заработок - Отчет

	Средний дневной заработок	Средний часовой заработок	Сумма начислений	Отработано дней	Отработано часов	Оплатить дней	Оплатить часов	Начи
▶	0,00	0,00	0,00	0,00	0,00	19,00	151,00	
	0,00	0,00	0,00	0,00	0,00	10,00	0,00	

Полнота 'И' по полям: Норма дней и Норма часов - Отчет

	Норма часов	Сумма начислений	Отработано дней	Отработано часов	Средний дневной заработок	Средний часовой заработок	Оплатить дней	Оплатить часов
▶	0,00	0,00	0,00	0,00	0,00	0,00	19,00	151,00
	0,00	0,00	0,00	0,00	0,00	0,00	10,00	0,00
	0,00	0,00	0,00	0,00	0,00	0,00	10,00	0,00
	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00
	0,00	0,00	0,00	0,00	315,00	31,50	19,00	151,00
	0,00	0,00	0,00	0,00	0,00	0,00	5,00	40,00
	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Рисунок 17 – Протоколы результатов проверки

7.4. Доступ к ресурсам и авторизация действий в Интеграторе

7.4.1.Разграничение доступа к ресурсам модели

Интегратор обладает механизмом разграничения доступа к ресурсам единого информационного хранилища данных. Для этого используются следующие способы:

- проектирование рабочих столов пользователей, соответствующих их должностным обязанностям;
- регистрация пользователей в системе и назначение прав доступа;
- разграничение доступа к данным по принадлежности пользователя организационно-структурной единице организации.

7.4.2.Проектирование рабочих столов

Формирование рабочих столов как исполняемых моделей позволяет разграничить доступ к функциональным ресурсам системы (см. Рисунок 18).

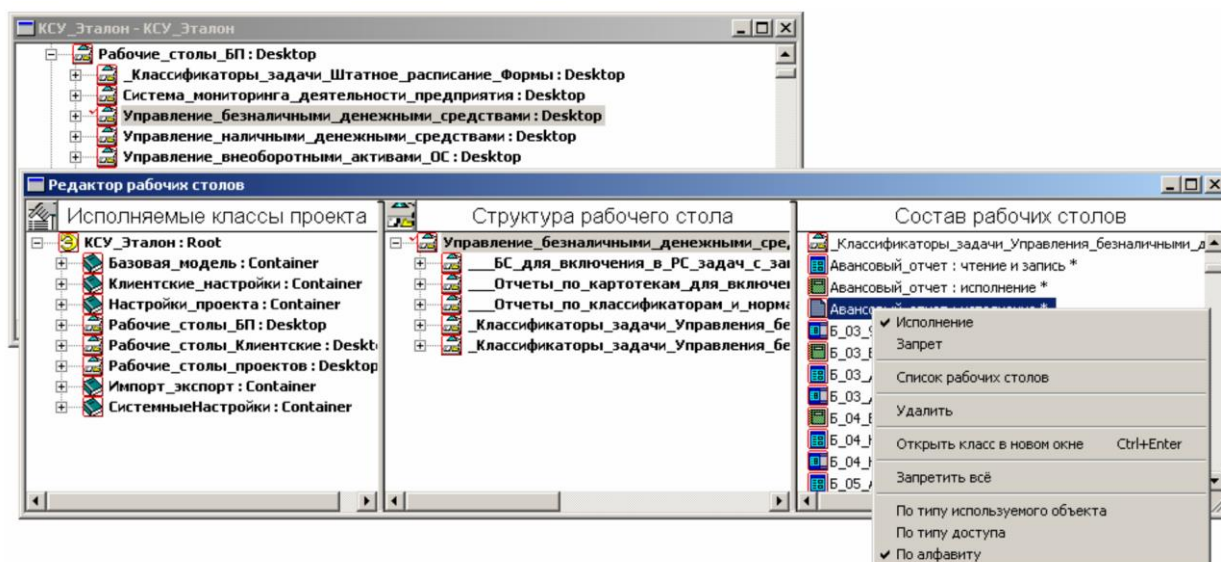


Рисунок 18 – Редактирование исполняемой модели в редакторе рабочих столов

7.4.3.Регистрация пользователей в системе и назначение прав доступа

Регистрация пользователя и назначение ему права доступа к ресурсам выполняется администратором в диалоге Управление пользователями:

- назначение системных привилегий на странице Пользователи (см. Рисунок 19);
- установка доступные функции при работе с исполняемыми классами рабочего стола, назначенного на странице Рабочий стол (см. Рисунок 20).

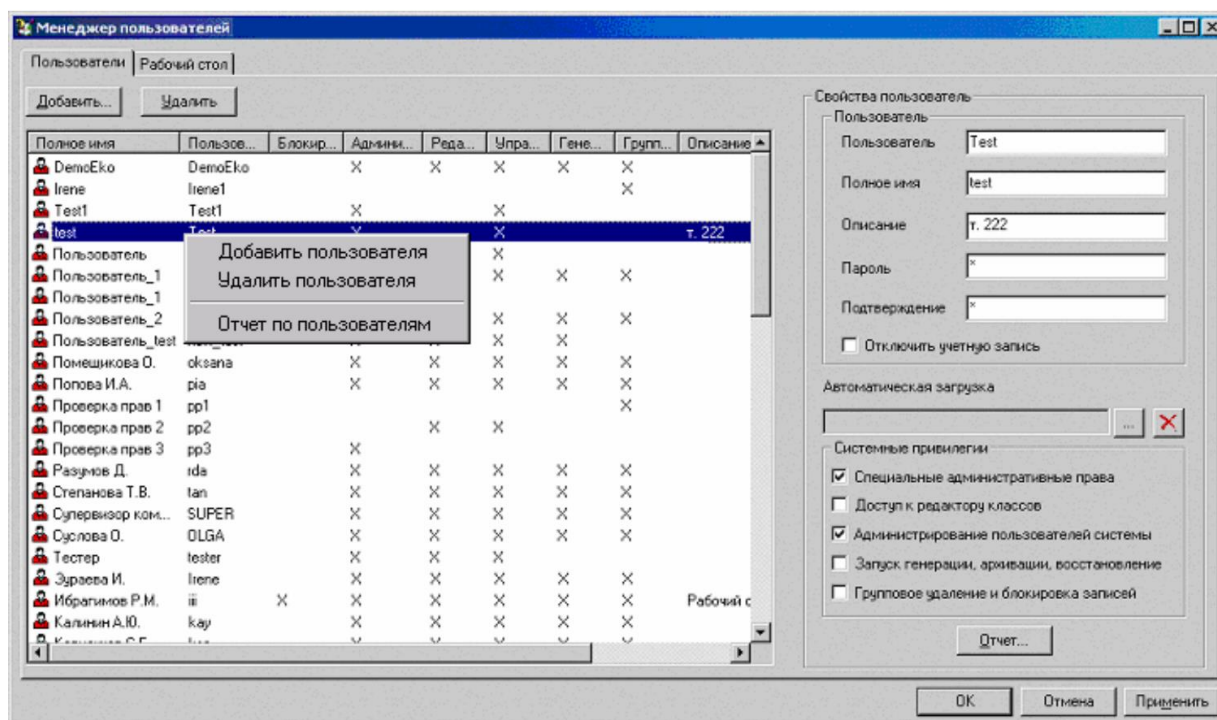


Рисунок 19 – Регистрация пользователя в системе на странице Пользователи диалога Управление пользователями

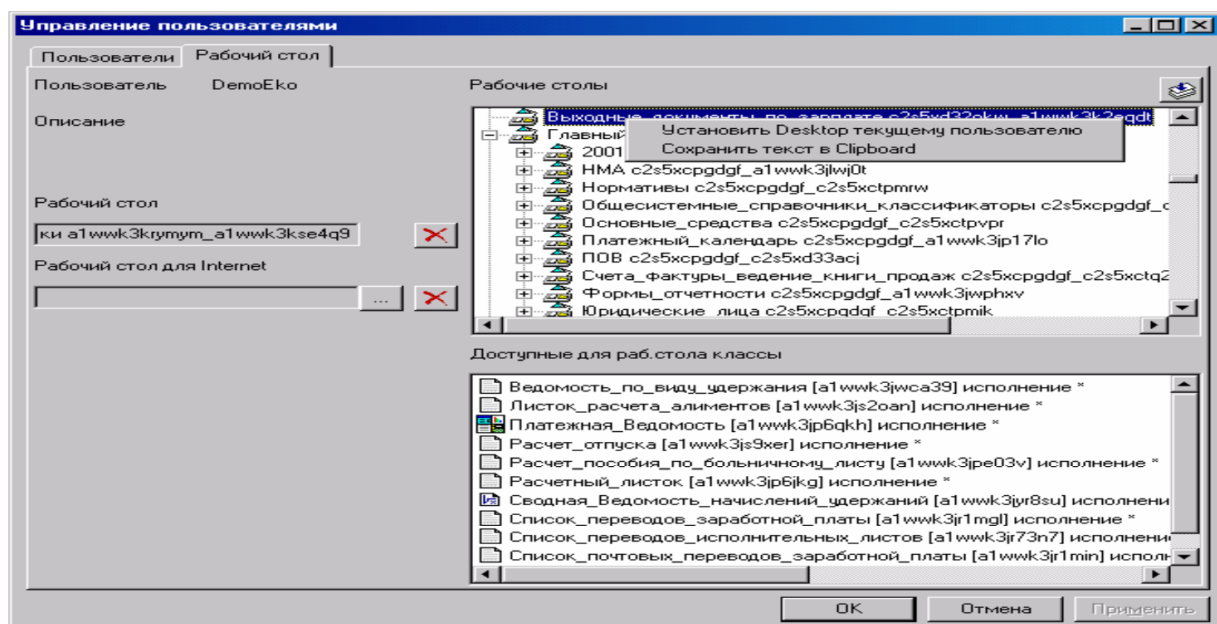


Рисунок 20 – Назначение пользователю рабочего стола на странице Рабочий стол диалога Управление пользователями

7.4.4. Работа пользователя в среде Интегратора

Пользователь при входе в систему попадает на назначенный ему рабочий стол, сформированный в соответствии с его должностными обязанностями. Функционал рабочего стола определяет доступ к данным единого хранилища (см. Рисунок 21).

Рисунок 21 – Вид рабочего стола пользователя на *RunTime*

7.5. Авторизация действий

Контроль и авторизация действий пользователя и проектировщика в #Гиперграф:Интегратор осуществляется, благодаря:

- – ведению системного протокола;
- – наличию в таблицах базы данных специальных полей об авторе и корректоре записей;
- – настраиваемому доступу пользователя к информации базы данных в соответствии с его принадлежностью той или иной структурной единице организации.

7.6. Навигация по информационному пространству

Информационное пространство #Гиперграф:Интегратор -Интегратора представляет набор вложенных контейнеров проектирования, используемых для развития модели, и исполнимых контейнеров, описывающих функциональную наполненность рабочих столов пользователей (см. Рисунок 21).

Системный аналитик, проектируя новые задачи, использует интерфейсные окна (см. Рисунок 22).

7.7. Оперативная работа с данными единого хранилища

7.7.1. Проектирование интерфейсов

Для визуализации данных унаследованных программных комплексов в среде Интегратора проектируются различные формы ввода и корректировки данных, отчеты, документы и т.п.

Для этого используется набор базовых классов и специализированные редакторы – встроенные Case-средств Интегратора.

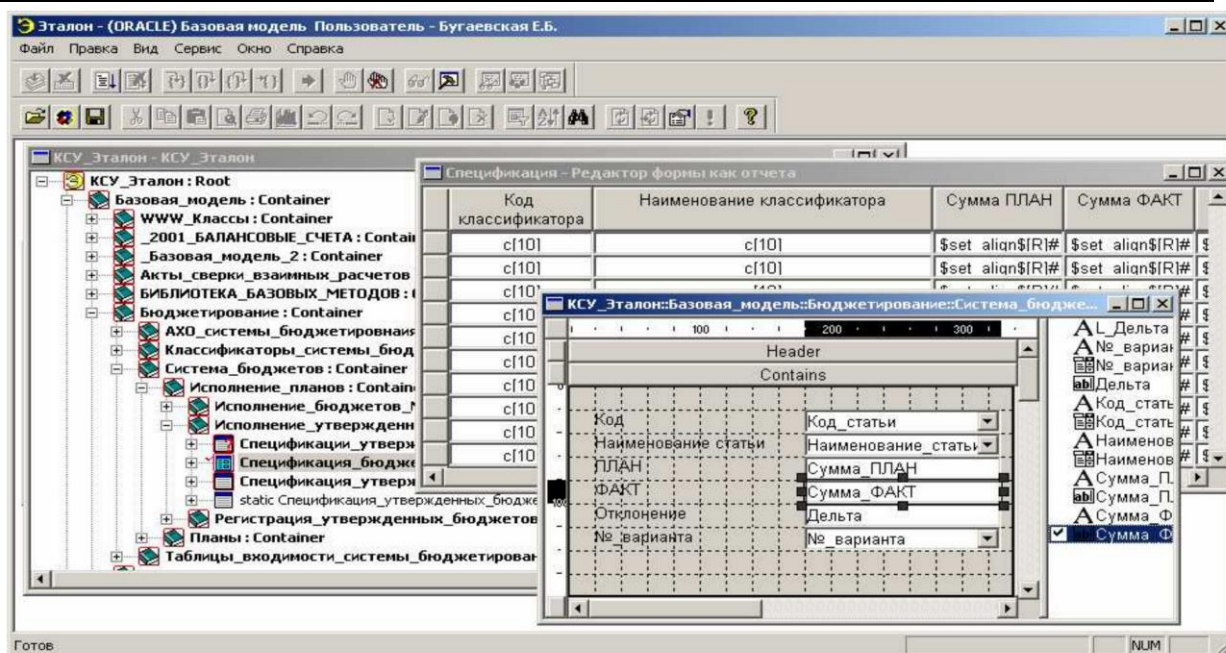


Рисунок 22 – Пример использования встроенных Case-средств для развития модели

7.7.2. Оперативная работа с данными единого хранилища, их анализ

Конечный пользователь имеет возможность подготовить, распечатать и сохранить желаемый аналитический отчет, составленный по информации баз данных программных комплексов, объединенных Интегратором в единое информационное пространство.

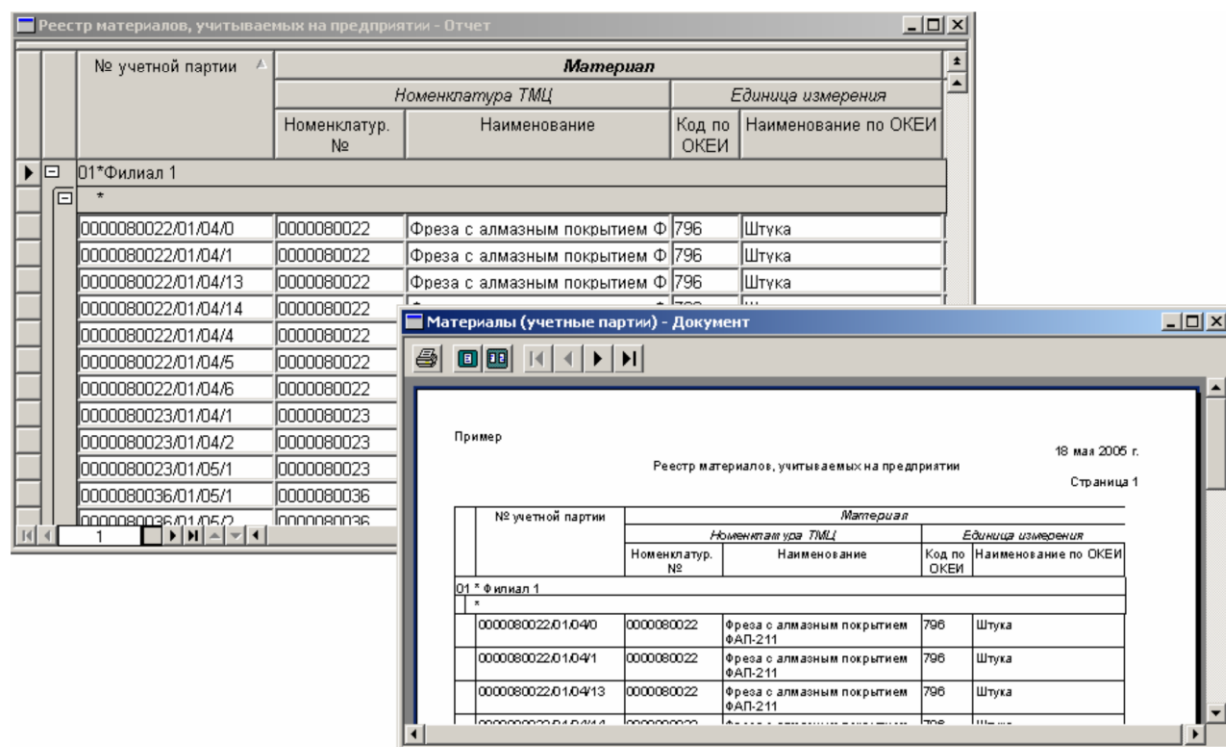


Рисунок 23 – Пример получения отчетных материалов пользователем

7.8. Развитие модели организации на базе единого хранилища данных

7.8.1. Проектирование новых задач на базе единого хранилища данных

Интегратор позволяет эволюционно развивать единую модель организации за счет проектирования новых бизнес-процессов, использующих информацию баз данных различных программных комплексов, и, таким образом, формировать единую систему аналитической обработки данных, управления потоками работ и управления знаниями.

При этом проектирование ведется исключительно в Интеграторе, без доработок в унаследованных программных комплексах.

7.9. Обмен данными в едином информационном пространстве

7.9.1. Импорт и экспорт данных

Если нет возможности организации оперативного доступа (on-line) к компонентам, включенным Интегратором в единую информационную модель организации, т.е. они территориально разнесены, отсутствуют каналы связи или единая сеть, то используется реализованный в Интеграторе механизм импорта/экспорта или репликации данных. На рисунке (см. Рисунок 24) приведена схема использования механизмов импорта/экспорта или репликации для обмена данными.

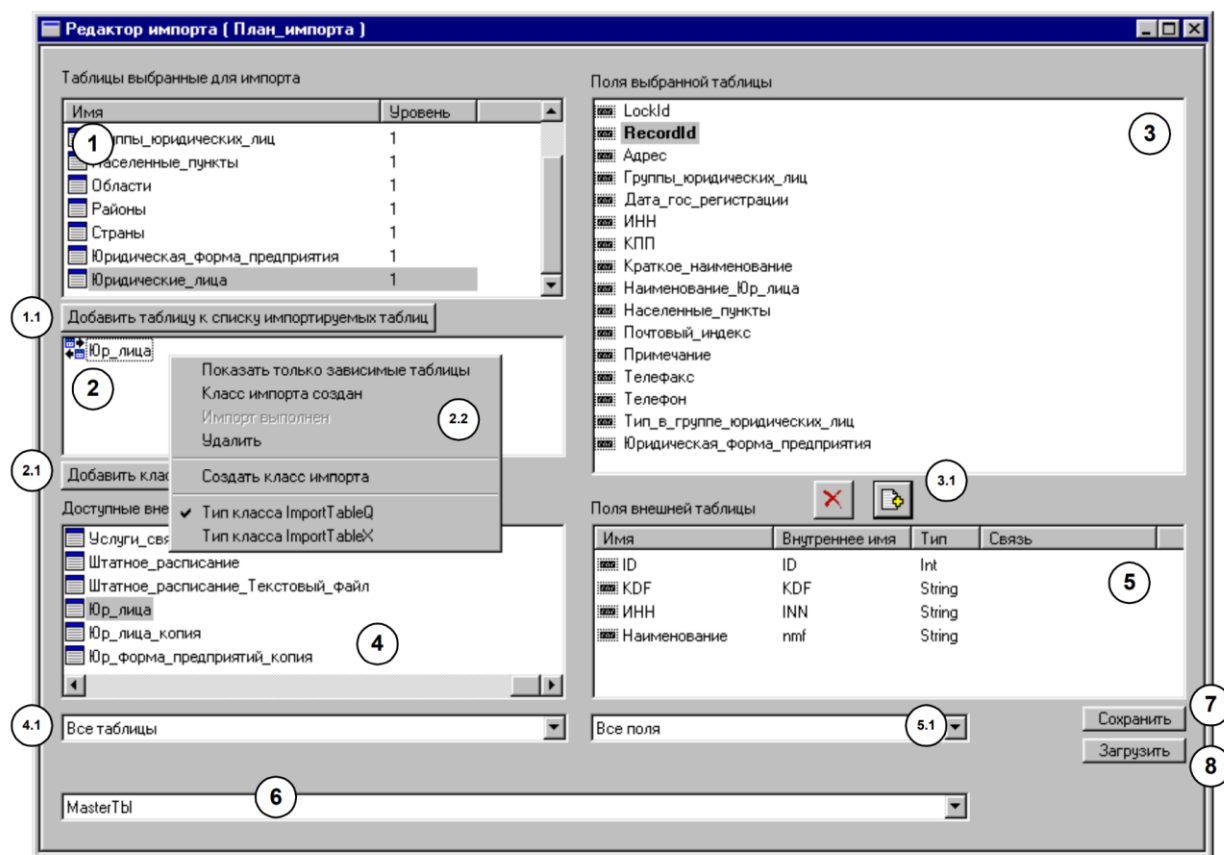


Рисунок 25 – Вид окна редактора импорта

Редактор импорта, позволяет сформировать контейнер «План импорта», в который помещаются контейнеры и классы для импорта (наследники базовых классов ImportTableQ и ImportTableX).

Импорт осуществляется при запуске на выполнение созданного класса процедуры импорта.

7.9.2.Репликация данных

Механизм репликации данных используется для согласования баз данных, сформированных на различных серверах Интегратора. Репликация повышает эффективность использования информации.

Пользователь может определить объем и состав реплицируемых данных, установив соответствующий режим репликации:

- – перенести все записи, заменив старые;
- – перенести только измененные;
- – перенести добавленные;
- – перенести записи, добавленные или измененные за определенный период времени.

Допускается репликация данных только указанных полей таблиц.

Репликация данных включает в себя этап экспорта данных таблиц Интегратора в файлы и последующий этап импорта данных из этих файлов в таблицы Интегратора. Репликация данных выполняется пакетом для нескольких таблиц.

Для проектирования репликации используются специально разработанные в Системе базовые классы:

- – ReplSrcData для формирования пакета при экспорте данных таблиц;
- – ReplTable для определения параметров репликации данных отдельной таблицы;
- – ReplTrgData для выполнения импорта данных таблиц;
- – EMail для определения параметров почтового клиента, если репликация осуществляется по электронной почте.

7.9.2.1. Экспорт данных при репликации

Для экспорта данных таблиц создается класс-наследник базового класса ReplSrcData, в котором для каждой экспортируемой таблицы создается объект базового класса ReplTable. Ссылка TableX@ refTable в этом объекте инициализируется статическим объектом экспортируемой таблицы, а выбор значения свойства Replication_mode определяет способ замены записей при импорте данных.

Пакет экспорта данных можно подготовить в специальном редакторе.

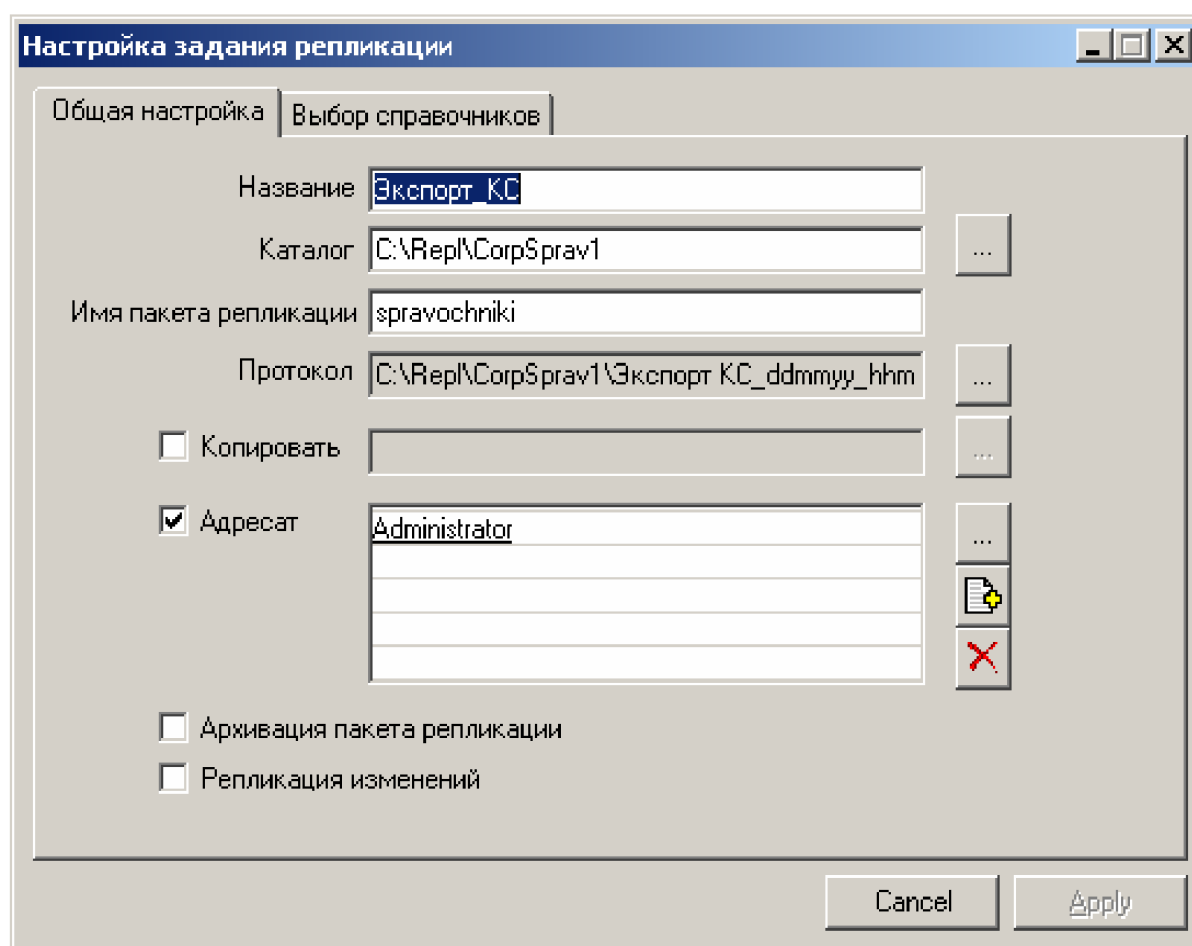


Рисунок 26 – Настройка задания репликации (закладка «Общая настройка»)

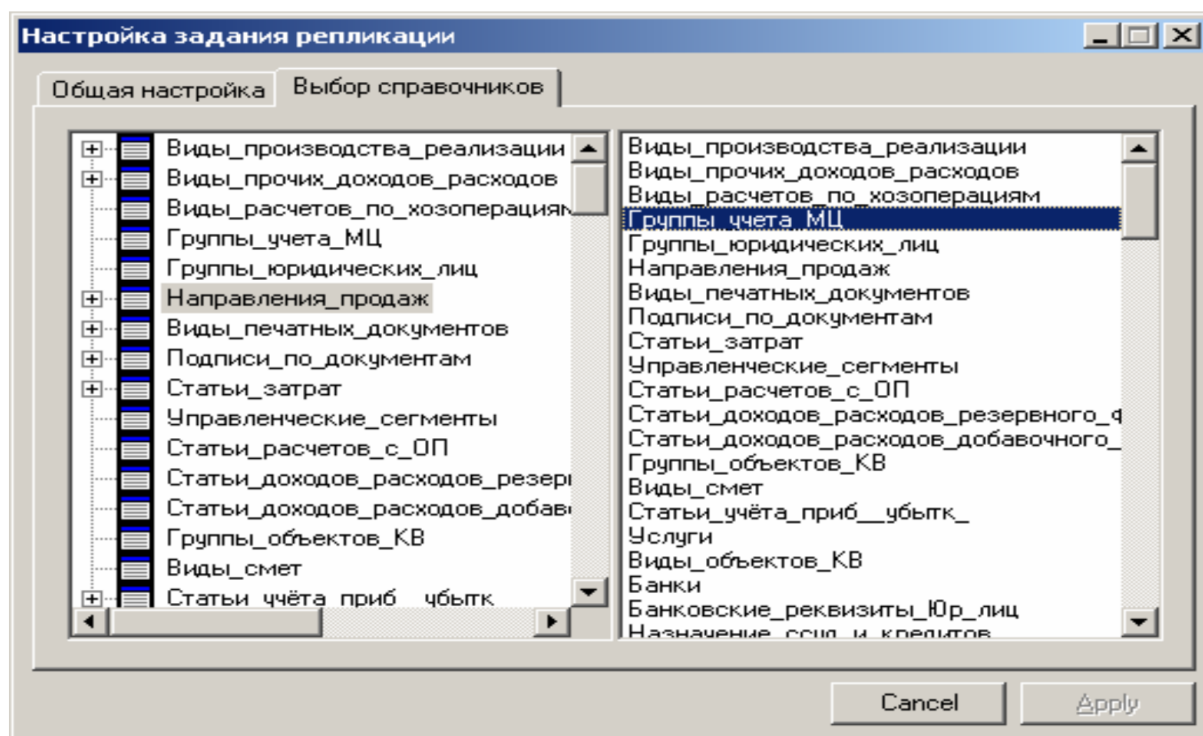


Рисунок 27 – Настройка задания репликации (закладка «Выбор справочников»)

Ход выполнения экспорта можно проследить в строке состояния, где отображается имя обрабатываемой пользовательской таблицы с указанием её порядкового номера в пакете репликации и индикатор процесса.

По результатам выполнения экспорта раскрывается окно с протоколом.

В результате выполнения экспорта будут созданы:

- – текстовый файл (*.rep), описывающий структуры экспортированных таблиц;
- – файл переносимых данных (Table_*.dbf) для каждой таблицы;
- – текстовый файл протокола.

7.9.2.2. Импорт данных при репликации

Импорт данных таблиц производится запуском на выполнение наследника базового класса ReplTrgData.

В окне Открыть файл (аналог окна «Сохранение файла») следует указать путь и имя нужного файла *.rep с описанием структуры импортируемых таблиц в текстовом формате. После этого в раскрывшемся окне (см. Рисунок 28) будет приведен список импортируемых данных.

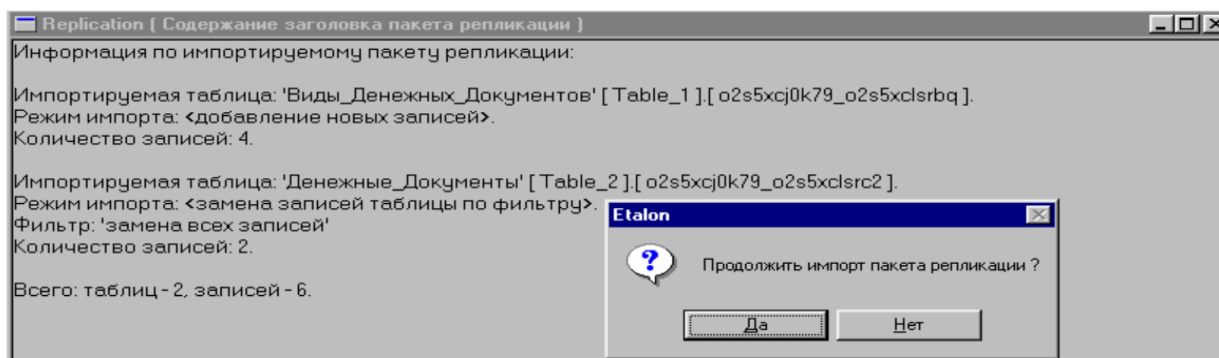


Рисунок 28 – Вид окна с информацией по импорту данных таблиц

Предлагаемое описание структуры можно изменить, внося исправления в текстовый файл описания *.rep и/или файлы данных таблиц Table_*.dbf.

По результатам выполнения импорта формируется протокол. Протоколы репликации

можно сохранять в файле. Результат репликации, время её проведения и имя пользователя регистрируются в таблице «Системный_протокол».

Для выполнения репликации с использованием электронной почты необходимо добавить в класс-наследник базового класса ReplTrgData объект базового класса Email и определить его свойства.

7.9.3. Работа с буферными базами данных

Интегратор позволяет в единую информационную модель включать только часть структур и часть данных унаследованных программных комплексов. Для этого создаются специальные буферные базы данных. На рисунке (см. Рисунок 29) представлена схема проецирования в единую модель части базы данных программного комплекса.

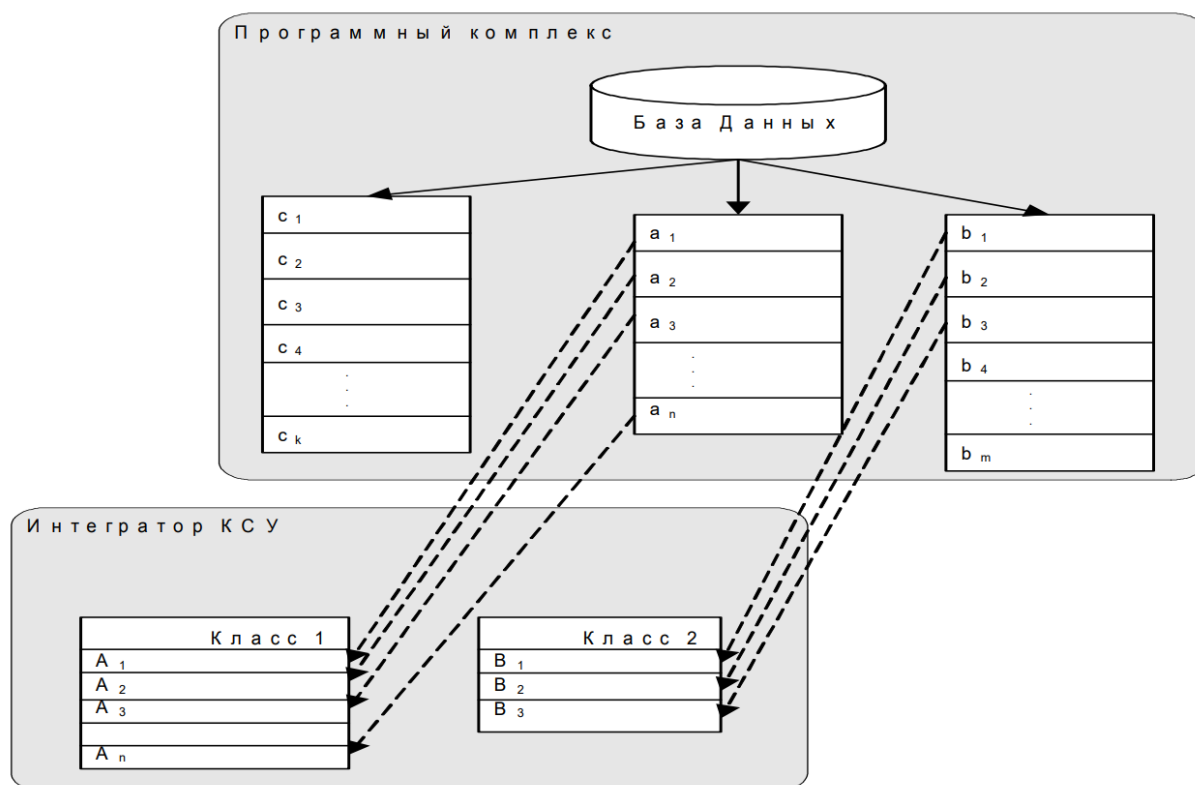


Рисунок 29 – Проецирование в единую модель части базы данных программного комплекса

7.9.4. Разработка и исполнение регламентов обмена данными

Любой класс Интегратора, как реализация конкретного алгоритма обработки данных, может запускаться из командной строки. Это предоставляет возможность планировать (определять) моменты запуска Интегратора. Описание последовательности и периодичности запуска определяется регламентами.

Регламенты в технологии Интегратора реализованы в нескольких вариантах:

- принудительный запуск класса;
- периодический запуск процедуры, согласно установленному расписанию;
- событийный запуск процедуры осуществляется по наступлению того или иного определенного ранее события.

7.9.4.1. Принудительный запуск

Запуск класса Интегратора может быть выполнен из подготовленного командного файла или из командной строки.

7.9.4.2. Периодический запуск

Программа Агент запуска (sagent.exe) включается в комплект поставки Интегратора и используется для автоматического повременного запуска подготовленных исполняемых модулей различного назначения. В комплект Агента запуска включается два файла:

- sagent.exe (программа установки и исполняемый файл сервиса);
- sagent.ini (шаблон файла настроек сервиса).

Для работы программы Агент запуска предварительно следует в файле sagent.ini сформировать список заданий.

Запуск сервиса Etalon Start Agent можно выполнить через специальную панель управления или из командной строки.

Для формирования в файле sagent.ini списка заданий, запускаемых автоматически программой «Агент запуска», можно использовать специальную сервисную программу, загружаемую по команде «Сервис ⇒ Запуски по расписанию» (см. Рисунок 30).

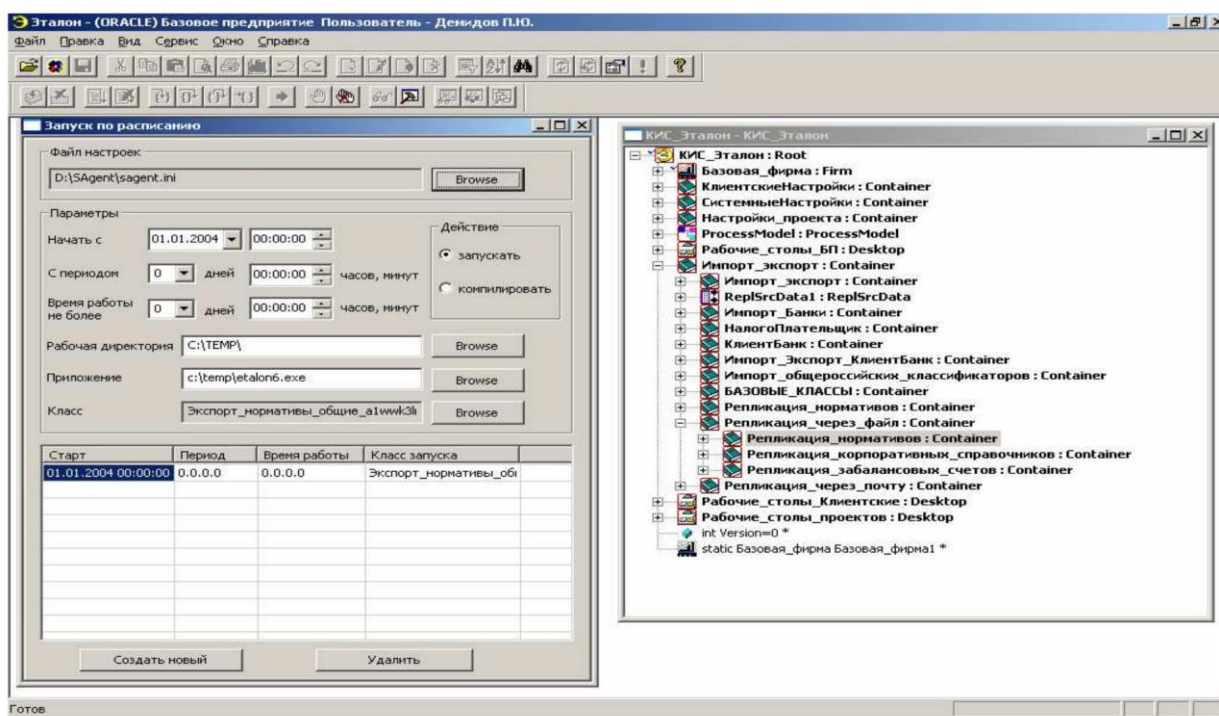


Рисунок 30 – Вид окна для задания параметров запуска по расписанию

Созданные настройки записываются и хранятся в файле заданий sagent.ini, который используется программой «Агент запуска».

Настройки содержат информацию о времени первого запуска, периодичности запуска, продолжительности работы агента и т.п.

7.9.4.3. Событийный запуск

Событийный запуск процедуры в Интеграторе осуществляется по наступлению того или иного определенного ранее события.

Событийный запуск проектируется как:

- – отдельное сервисное приложение, запускаемое из Интегратора;
- – специальный класс, описывающей механизм проверки наступления события и запуск процедуры.

Интегратор может использоваться в качестве сервера приложений, который периодически опрашивает разделяемые ресурсы и при изменении их состояния запускает необходимый класс (по событию). Такой вариант целесообразно применять, если сервер приложений размещается на выделенном компьютере.

7.10. Управления мастер-данными

Данные можно классифицировать по различным признакам: по их принадлежности, организации, актуальности и по многим другим. Очевидно, что одни и те же данные могут присутствовать в нескольких категориях одновременно.

Мастер-данными (Master Data, основные данные, общие данные) будем называть данные, общие для нескольких используемых информационных систем. Мастер-данные являются основой интеграции, эффективной обработки, качественного анализа деятельности. Часто понятие мастер-данных отождествляется с понятием нормативно-справочной информации. Это не всегда корректно, НСИ является одним из видов мастер-данных, хоть и наиболее существенным.

Высокое качество мастер-данных значительно повышает эффективность бизнес процессов. А включение технологий MDM в существующую инфраструктуру организации позволяет:

7.10.1. Базовое интегрированное пространство

#Гиперграф: Интегратор позволяет создать, поддерживать актуальное состояние и обеспечить доступ всех программных продуктов к базовому интегрированному пространству, содержащему общие данные, необходимые для расчетов в различных программных продуктах:

- – международные классификаторы и нормативы;
- – общенациональные (общероссийские) классификаторы и нормативы;
- – общесистемные классификаторы, нормативы и картотеки.

7.10.2. Библиотеки моделей интеграции

Развитие технологий Эталон-Интегратора позволяет создавать библиотеки решений (библиотеки моделей интеграции) как по использованию и портации данных, так и по разработке единого информационного и функционального пространства для унаследованного мультиплатформенного программного обеспечения организации.

В настоящее время созданы и постоянно пополняются библиотеки моделей интеграции для информационных и программных продуктов, созданных органами муниципального, регионального и государственного управления, например:

Федеральная служба государственной статистики:

- – Общероссийские классификаторы;
- – Каталог статистических показателей форм федерального государственного статистического наблюдения;
- – Формы финансовой и бухгалтерской отчетности.

Федеральная налоговая служба:

- – Программный продукт «Оазис», «Контур»;
- – Программный продукт «Налогоплательщик»;
- – Классификатор адресов РФ (ФАИС);
- – Единый государственный реестр юридических лиц (ЕГРЮЛ).

Другое программное обеспечение:

- – 1С;
- – Инфин;
- – Парус;
- – Галактика;
- – R/3;
- – Oracle Applications;
- – Ахпта;
- – IFS;
- – Navision;
- – Системы «Клиент-банк»;
- – Биллинговые системы.

8. ОПЫТ ПРИМЕНЕНИЯ

Компания «ГиперГрафГрупп» и её отраслевые и региональные партнеры за последние несколько лет реализовали в на #Гиперграф:Платформе более 700 успешных проектов Информационных Систем Управления для крупных предприятий и органов государственной власти федерального и регионального уровней.

При этом, обеспечивалась эффективная семантическая интеграция с множеством унаследованных функциональных информационных систем, находящихся в эксплуатации. И если на начальном этапе каждого проекта формировалось взаимодействие #Гиперграф:Платформы с более 50 унаследованными системами с выделением общих данных и функций, то через 3-9 месяцев промышленной эксплуатации #Гиперграф:Платформы оставались интегрированными лишь 2-3 незамещенных системы, которые можно классифицировать следующим образом:

- специализированные системы реального времени, являющиеся частью оборудования.
- сертифицированные специальным образом информационные системы, сохраненные из-за субъективных обстоятельств.
- графо-центричные технологии применялась, в основном, для реализации масштабных разнофункциональных и межотраслевых проектов как в целях исследования и поиска ограничений технологии, так и с целью развития и наращивания единой Модели Знаний.