

**Реинжиниринг геоинформационной системы  
компании ООО «Владлинк»**

**Бессонов Илья Игоревич**

*студент 5 курс, Институт информационных технологий,*

**Богданова О.Б.,** *старший преподаватель кафедры ИТС,*

*Владивостокский государственный университет экономики и сервиса,*

*Владивосток, Россия*

*E-mail: iliyabessonov@gmail.com*

*В статье описывается задача реинжиниринга геоинформационной системы компании ООО «Владлинк». Приведены особенности архитектуры приложения и возможные подходы для выполнения реинжиниринга.*

**Ключевые слова и сочетания:** *реинжиниринг, подходы реинжиниринга, геоинформационная система, функции ГИС, создание новой информационной системы, модификация существующей информационной системы, адаптация готовой информационной системы.*

Основная деятельность компании ООО «Владлинк» заключается в предоставлении услуг Интернета и телевидения абонентам. В ходе осуществления своей деятельности, компания выполняет задачи по модернизации, ремонту и территориальному расширению сетевой инфраструктуры. Такие же задачи стоят и перед всеми остальными интернет-провайдерами, предоставляющими телекоммуникационные услуги. Особенностью этих задач является необходимость использования географических данных для планирования и управления сетями. Применение геоинформационных систем (ГИС) позволяет провайдерам оценивать состояние сетевой инфраструктуры, улучшать качество предоставляемых услуг и планировать мероприятия по модернизации оборудования и расширению зоны покрытия.

В компании ООО «Владлинк» отделом информационных технологий, была разработана собственная ГИС. В результате разработки была получена интерактивная визуальная модель инфраструктуры сети. К функциям ГИС можно отнести следующие: проектирование сетевой инфраструктуры компании; просмотр данных об оборудовании на конкретных участках сети, количестве абонентов в домах и состоянии подключаемых домов к общей сети, а также выполняемых в них работах на текущий момент.

Несмотря на широкие функциональные возможности системы, в ходе её эксплуатации были выявлены следующие недостатки:

- необходимость постоянного ручного обновления данных проектировщиком при изменениях на каких-либо участках сети;
- отсутствие данных о выполненных действиях пользователями.

Для устранения указанных недостатков было принято решение о выполнении реинжиниринга системы состоящего в добавлении нового функционала.

При осуществлении реинжиниринга может применяться один из следующих подходов[1]:

- создание новой информационной системы (ИС) взамен существующей;
- модификация существующей ИС;
- адаптация готовой ИС стороннего разработчика.

Положительные и отрицательные стороны использования перечисленных подходов представлены в таблице 1.

Таблица 1. Положительные и отрицательные стороны подходов реинжиниринга

Название подхода	Плюсы	Минусы
Создание новой ИС	<ul style="list-style-type: none"> <li>– достаточная изученность процесса разработки;</li> <li>– более точная прогнозируемость по времени, стоимости и другим ресурсам;</li> <li>– возможность отказаться от устаревших технологий и применить новые технологии</li> </ul>	<ul style="list-style-type: none"> <li>– высокая ресурсоёмкость;</li> <li>– необходимость наличия резерва времени;</li> <li>– высокая вероятность повторного решения задач, выполненных на приемлемом уровне</li> </ul>
Модификация существующей ИС	<ul style="list-style-type: none"> <li>– небольшая ресурсоёмкость;</li> <li>– малые затраты на переобучение пользователей;</li> <li>– возможность широкого использования компонентов существующей информационной системы;</li> <li>– отсутствие необходимости полной остановки системы</li> </ul>	<ul style="list-style-type: none"> <li>– сложность прогнозирования процесса, оценки необходимых ресурсов;</li> <li>– сложность структурной модели, описывающей реинжиниринг ИС;</li> <li>– высокая вероятность «волны изменений»</li> </ul>
Адаптация готовой ИС	аналогично подходу модификации существующей ИС	необходимо наличие аналогичной ИС

Для выполнения реинжиниринга ГИС наиболее оптимальным является подход модификации существующей информационной системы, так как он удовлетворяет следующим критериям:

- ограниченность срока на выполнение реинжиниринга;
- отсутствие необходимости остановки существующей системы, в период осуществления реинжиниринга.

Помимо этого, добавляемый функционал будет являться расширением функционала существующей системы, то есть большая часть компонентов может быть заимствована из действующей системы.

В результате выполнения проектирования информационной системы с учётом добавляемых функциональных возможностей системы была составлена модель прецедентов. Проектирование осуществлялось с применением объектно-ориентированного подхода по методике ICONIX[2, 3].

Все прецеденты были сгруппированы в 2 пакета, представленные на рисунке 1. В пакет «Работа с картой», представленный на рисунке 2, были включены прецеденты, в которых пользователь выполняет действия непосредственно на слоях карты. Пакет «Работа с сетью», представленный на рисунке 3, содержит прецеденты добавления, удаления муфт или кроссов и соединений волокон в них.

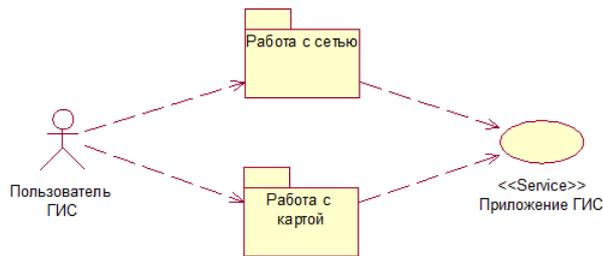


Рисунок 1 – Диаграмма пакетов с прецедентами

В ходе реинжиниринга недостатки системы были устранены путём добавления нового прецедента «Обновить данные о кабелях» (пакет «Работа с картой») и операции создания записей о выполненных действиях.

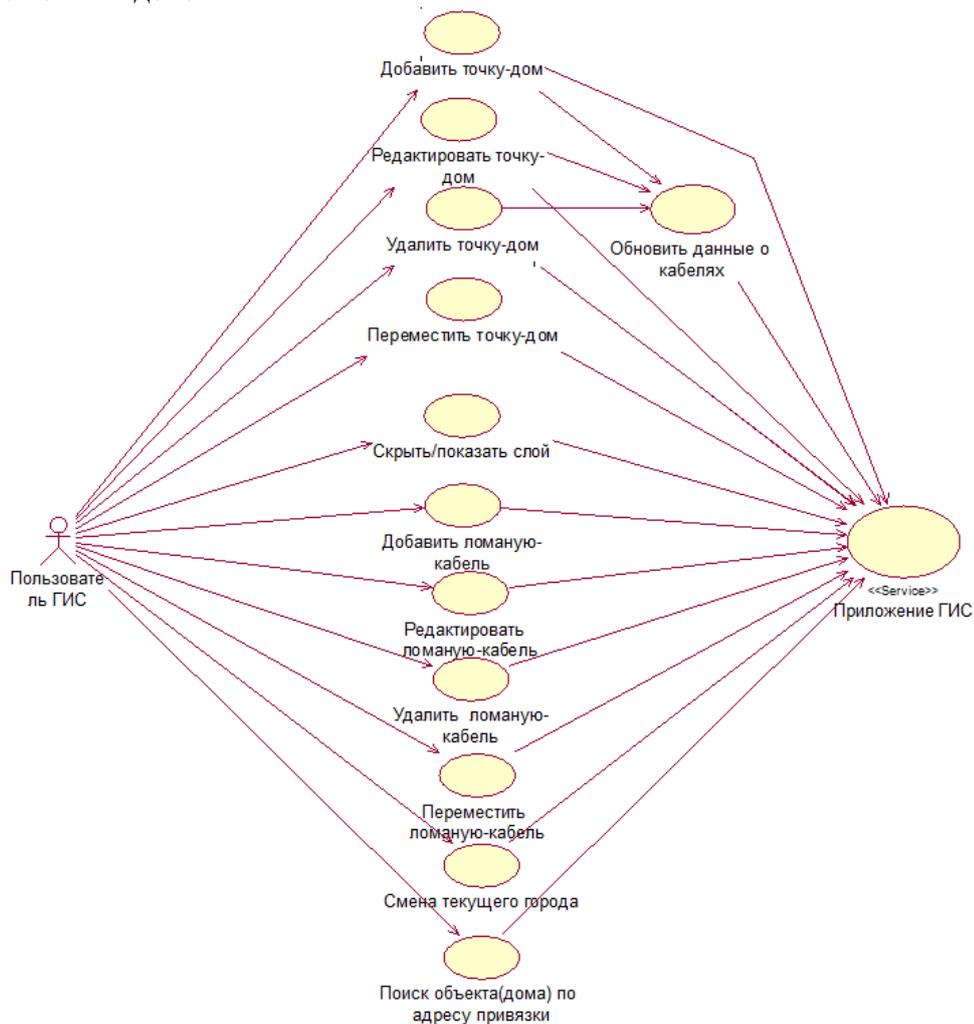


Рисунок 2 – Диаграмма прецедентов пакета «Работа с картой»

Прецедент «Обновить данные о кабелях» вызывается, как автономно - с определённым интервалом времени, так и после выполнения прецедентов «Добавить точку-дом», «Редактировать точку-дом», «Удалить точку-дом». Назначение этого прецедента заключается в импорте данных из корпоративной информационной системы в ГИС. Непосредственно в корпоративной информационной системе происходит создание нарядов на монтаж и демонтаж оборудования. Таким образом, в базе данных корпоративной информационной системы содержится актуальная информация о текущем состоянии сетевой инфраструктуры компании. Алгоритм обновления данных о кабелях, реализованный в одноимённом прецеденте, представлен на рисунке 4.



Рисунок 3 – Диаграмма прецедентов пакета «Работа с сетью»

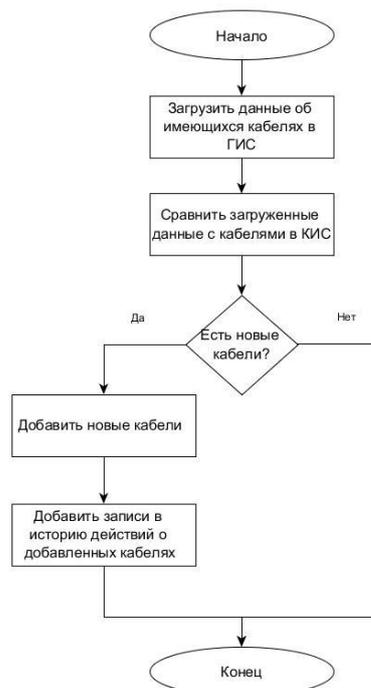


Рисунок 4 – Алгоритм обновления данных о кабелях

Операция создания записей о выполненных действиях была добавлена в прецеденты, вносящие изменения в модель сетевой инфраструктуры. Рассмотрим наиболее часто используемый прецедент «Добавить точку-дом», представленный диаграммой пригодности на рисунке 5. В этот прецедент была добавлена новая сущность «История действий пользователей» и вызов прецедента «Обновить данные о кабелях».

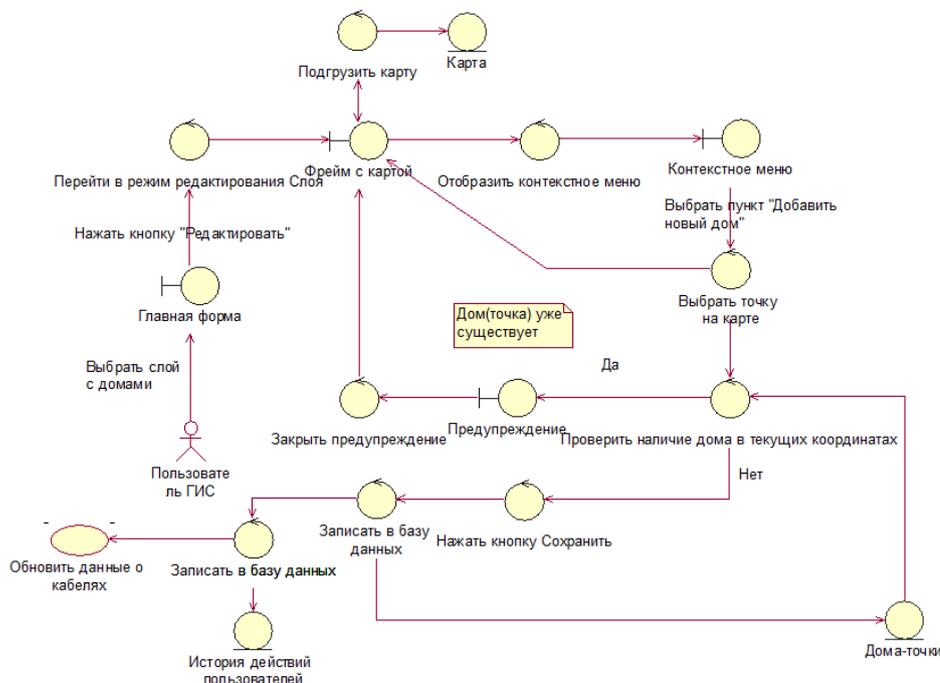


Рисунок 5 – Диаграмма пригодности прецедента «Добавить точку-дом»

Используя модели, полученные в результате проектирования, была выполнена программная модификация ГИС.

ГИС представляет собой клиент-серверное приложение. Функционал клиентской части выполняют html-страницы и js-скрипты, полученные после ретрансляции GWT-компилятором. Функционал серверной части является объединением трех различных подсистем реализованных на Java, Perl, Python языках.

Подсистема «Вывод детализированной информации», написанная на языке Python, создана на основе веб-фреймворка Django. Она отвечает за вывод всплывающих окон с детализированной информацией об объектах, при наведении курсора мыши на них.

Подсистема «Операции с объектами на карте», написанная на языке Perl, реализует серверный функционал, связанный с выборкой, созданием, редактированием и удалением объектов на карте.

Подсистема «Основные функции ГИС», написанная на языке Java, отвечает за следующий функционал:

- выборка адресов улиц и номеров домов;
- загрузка требуемых областей карты;
- создание и удаление слоёв с объектами;
- создание, редактирование, удаление муфт, кроссов и соединений волокон в них;
- поиск объектов по адресу привязки.

Добавление новых функций к ГИС системе является сложным процессом, если эти функции связаны с несколькими серверными подсистемами одновременно. В случае внесения изменений в подсистему, написанную на Java, потребуется перекомпиляция GWT-проекта целиком.

Помимо этого, необходимо поддерживать актуальность версий используемых технологических решений, на что в данном случае затрачивается много времени, так как серверные подсистемы созданы на основе различных технологий.

Таким образом, перед добавлением нового функционала, необходимо реализовать переход серверных подсистем к единой технологии. В качестве целевой технологии был выбран веб-фреймворк Mojolicious. Его использование позволяет выполнить перенос серверного функционала ГИС системы за достаточно короткое время, и не требует глубокого изучения самого фреймворка перед началом непосредственного написания кода приложения.

Перенос каждой из подсистем имеет свои технологические особенности. В случае с подсистемой «Вывод детализированной информации», разработанной на платформе Django, все обрабатываемые запросы и используемые модели могут быть получены из «Менеджера URL»[4]. После выполнения анализа запросов и моделей, осуществляется перенос подсистемы на платформу Mojolicious.

Подсистема «Операции с объектами на карте» реализована на основе технологии CGI (Common Gateway Interface) с использованием библиотеки «CGI::Simple». Данная подсистема обрабатывает четыре типа запросов - get, post, put, delete, и для каждого типа вызывается различная функция из пакета «WFS». Для каждого типа запросов также выполняется анализ исходного кода используемых функций с последующей переработкой под платформу Mojolicious.

Подсистема «Основные функции ГИС» имеет модульную структуру и разработана с использованием GWT-технологии. Для выполнения переноса этой подсистемы кроме анализа исходного кода серверных модулей необходимо проанализировать код клиентских модулей.

Взаимодействие клиентских и серверных модулей реализовано посредством библиотеки «AsyncCallback» поставляемой вместе с фреймворком GWT[5].

Библиотека «AsyncCallback» позволяет разрабатывать код приложения ориентированно на данные, обеспечивая высокий уровень абстракции. Для формирования шаблона строки запроса используется подстрока «grc», определённая в модуле «AppRequestTransport.java». Строка запроса дополняется названием клиентского модуля с названием вызываемого метода и передаваемыми аргументами, затем выполняется запрос.

После выполнения анализа соответствующих клиентских и серверных модулей, функции серверных модулей перерабатываются под платформу Mojolicious.

В полученную серверную подсистему были добавлены: программная реализация прецедента «Обновить данные о кабелях»; функция записи действий пользователей.

Реинжиниринг состоял из следующих этапов: проектирования модели прецедентов с учётом добавляемого функционала; перехода серверной подсистемы к единой технологии; добавления программной реализации требуемого функционала.

При проектировании в состав модели прецедентов был добавлен новый прецедент «Обновить данные о кабелях», автономно вносящий изменения в кабельную инфраструктуру сети без участия проектировщика. Также в прецеденты, вносящие изменения в модель сетевой инфраструктуры, была добавлена операция записи действий пользователей регистрирующая все действия пользователей.

Переход серверной подсистемы к единой технологии был обоснован особенностью её архитектуры. Использование трёх различных технологических решений является неоправданно трудоёмким как при добавлении необходимого функционала, так и в ходе дальнейшей поддержки системы.

Заключительным этапом реинжиниринга стало добавление программной реализации прецедента «Обновить данные о кабелях» и функции записи действий пользователей.

## Литература

1. Ахтырченко К.В, Методы и технологии реинжиниринга ИС [Электронный ресурс]// Труды Института системного программирования РАН: [сайт].[2003]. URL: <http://www.citforum.ru/SE/project/isr>. (дата обращения: 20.04.2015).
2. Мейер Б. Объектно-ориентированное конструирование программных систем. – М.: Русская Редакция, 2004. – 1204 с.
3. Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов. – М.: ДМК Пресс, 2002. – 160 с.
4. Документация Django [Электронный ресурс]. – Режим доступа: <http://djbook.ru/rel1.4/topics/http/urls.html>.
5. Документация разработчика по технологии Google Web Toolkit [Электронный ресурс]. – Режим доступа: <http://www.gwtproject.org/doc/latest/tutorial/buildui.html>.