

Техническое описание
Инструментальной платформы *OpenCar*

Оглавление	
1. Общие сведения	3
2. Функциональное назначение	4
2.1. Архитектура	4
2.2. Системные и сервисные функции	5
2.3. Управление компонентами прикладной бизнес-логики	5
2.4. Редактор бизнес-процессов	6
2.5. Подсистема исполнения бизнес-процессов	6
2.6. Управление структурами бизнес-сущностей	6
2.7. Подсистема формирования пользовательского интерфейса	7
2.8. Создание прикладного решения	8
2.9. Подсистема формирования отчетности	9
2.10. Средства оповещения пользователей	10

1. Общие сведения

Платформа **Openlcar** представляет собой инструментарий для создания прикладного решения на основе процессного подхода.

Платформа **Openlcar** и решения на ее основе представляют собой Java EE приложения. Основные используемые компоненты:

- Oracle Java SE 7 (Oracle Binary Code License) / OpenJDK 7 (GPL 2)- стандартная версия платформы Java и среда исполнения приложений (для WildFly 10 требуется Java SE версии 8);
- JBoss Application Server 7.1.2.Final (LGPL 2.1) / WildFly 10.0.0.Final (LGPL 2.1) -JavaEE сервер приложений с открытым исходным кодом;
- JBoss Seam 2.3.1.Final (LGPL) -каркас (фреймворк) для разработки web приложений;
- JBoss jBPM 5.3.0.Final (Apache Software License 2.0) -процесный движок,
- поддерживающий моделирование и выполнение бизнес -процессов в нотации BPMN 2.0;
- JBoss jBPM Designer 2.4.2 (Apache License 2.0) -веб-ориентированный редактор процессов основанный на Oryx/Wарата (MIT License);
- Jasper Reports 3.7.4 (LGPL 3) -компонент для формирования отчетов с широкими возможностями;
- Vaadin UI 7.6.4 (Apache License 2.0) -java фреймворк для построения пользовательского интерфейса современных web-приложений;
- СУБД -в настоящий момент поддерживается работ а с реляционными СУБД (может использоваться одна из перечисленных ниже СУБД):
 - Firebird 2.5+ (Mozilla Public Licence 1.1);
 - PostgreSQL 9+ (PostgreSQL License);
 - Oracle 9+ (Oracle Technology Network License);
 - H2 1.3.176+ (Mozilla Public License Version 2.0 / Eclipse Public License 1.0).

*Средства разработки: **Eclipse -IDE среда для разработки на Java, Apache Maven** -средство сборки и управления зависимостями.

2. Функциональное назначение

2.1. Архитектура.

Платформа **OpenIcar** является основой для реализации решений со следующими характеристиками:

- Многозвенная архитектура. Использование многозвенной архитектуры позволяет повысить эффективность работы корпоративной информационной системы и оптимизировать распределение ее программно-аппаратных ресурсов. При этом, благодаря слиянию технологий Internet/intranet и "клиент/сервер", существенно упрощается процесс внедрения и сопровождения системы при сохранении высокой эффективности и простоте совместного использования информации.
- Использование современных и общепринятых платформ и средств разработки. Решение основано на платформе JavaEE (Java Enterprise Edition), которая является де-факто промышленным стандартом при построении многозвенных информационных систем предприятия и поддерживается как ведущими мировыми производителями программного обеспечения, так и огромным количеством независимых разработчиков.
- Web-интерфейс пользователей. Реализация интерфейса пользователя в форме Web-интерфейса позволяет минимизировать требования к аппаратному и программному обеспечению рабочих мест пользователей и является кросс-платформенным решением, что устраняет привязку к определенной операционной системе и позволяет использовать открытые ОС (Linux)
- Использование компонент с открытым исходным кодом (opensource). Решение максимально использует компоненты с открытым исходным кодом, находящиеся в открытом доступе (сервер БД, сервер приложений и т.п.), что для заказчика означает нулевую стоимость лицензии, а также позволяет:
 - Достичь большей надежности: эти компоненты используются большим числом компаний и, как показывает практика, лучше оттестированы и более надежны, чем созданное с нуля специализированное решение
 - Снизить трудозатраты на разработку
- Процессный подход. Решение основано на исполняемых бизнес-процессах (workflow / обеспечение жизненного цикла содержимого), что в первую очередь позволяет многократно увеличить гибкость системы и значительно снижает затраты на адаптацию решения при конкретном внедрении и последующие временные и ресурсные затраты на поддержку изменений бизнеса
- Открытость и сервисная ориентированность. Решение является открытым (сервисы доступны "снаружи"), что позволяет эффективно использовать его при построении SOA-ориентированной IT-инфраструктуры и в результате

воспользоваться преимуществами SOA-подхода, в т.ч.:

- Использование текущих инвестиций.
- Уменьшение рисков, связанных с внедрением проекта
- Возможность последовательного и непрерывного улучшения предоставляемых услуг.

2.2. Системные и сервисные функции

Платформа **OpenIcar** предоставляет набор системного сервисного функционала, такого как:

- средства для управления правами доступа пользователей
- средства для выполнения задач по расписанию
- средства по оповещению пользователей по email с формированием текста сообщения по шаблону
- административные сервисы (настройки приложения, миграция экземпляров бизнес-процесса на новые версии определений процессов, выполнение отладочных запросов и скриптов)
- средства по управлению обновлениями системы, в т.ч. для отражения изменений в структурах данных и бизнес-процессах

2.3. Управление компонентами прикладной бизнес-логики

Не весь прикладной функционал можно (или целесообразно) реализовывать в виде бизнес-процессов, часто в рамках решения прикладных задач бывает необходимо реализовать сервисы/утилиты с помощью разработки программных компонент, в т.ч. и для использования их в ходе выполнения бизнес-процессов. OpenIcar основан на каркасе разработки jboss seam и при реализации прикладных решений предоставляет прикладным программистам возможности разрабатывать прикладную бизнес-логику, используя языки программирования java и groovy, и использовать разрабатываемые прикладные компоненты для:

- использования их в бизнес-процессах
- использования их при выполнении операций(действий) над бизнес-сущностями
- непосредственного выполнения методов прикладных компонент с помощью автоматически генерируемого пользовательского интерфейса в тех случаях, когда создание бизнес-процесса для этого нецелесообразно (например, в случае процессов из одного шага) Разработку прикладных компонент целесообразно (и удобно) вести с применением современных средств разработки (eclipse), однако в случае необходимости изменения могут быть внесены с использованием простого текстового редактора. При этом в случае использования groovy изменения могут быть внесены без необходимости пересборки приложения (но в режиме промышленной эксплуатации потребуется перезапуск приложения после изменения прикладных компонент)

2.4. Редактор бизнес-процессов

Платформа **Openlcar** и решения на его основе содержат встроенный редактор бизнес-процессов, имеющий web-интерфейс и позволяющий:

- создавать и редактировать бизнес-процессы в нотации BPMN 2.0
- использовать при описании данных бизнес-процесса и при описании экранных форм шагов процесса описания прикладных бизнес-сущностей (для организации выбора из справочников и вывода или редактирования экранных форм объектов при выполнении шагов процесса)
- использовать компоненты системного уровня и прикладной бизнес-логики в шагах бизнес-процесса для вызова необходимых сервисов
- редактировать бизнес-процессы в режиме "черновика", без немедленного изменения непосредственно исполняемых процессов
- загружать процессы из внешних файлов
- осуществлять "развертывание" процесса (загружать новую версию процесса для непосредственного исполнения системой).

2.5. Подсистема исполнения бизнес-процессов

Платформа **Openlcar** и решения на его основе включает в себя движок исполнения бизнес-процессов jbrm (версии 5.x), позволяющий непосредственно исполнять процессы в нотации BPMN 2.0. При этом выполнение процесса производится в контексте работающего прикладного решения, и т.о. во время выполнения доступны вся системные сервисы (по управлению данными, проверке прав доступа и т.п.) и прикладная бизнес-логика. Для интеграции движка бизнес-процессов в прикладное решение, в т.ч. организации взаимодействия с пользователями, предоставляет специализированные экранные формы, которые:

- позволяют осуществлять запуск новых экземпляров бизнес-процессов в соответствии с полномочиями пользователей
- формируют и выводят списки пользовательских персональных заданий, которые поступили к ним в очередь в соответствии с шагами бизнес-процессов и ролями, выполняемыми пользователем
- предоставляют пользователям возможность ознакомиться с персональными заданиями из списков, ввести необходимые данные на данном шаге процесса данные и принять решения по дальнейшему выполнению бизнес-процесса.

2.6. Управление структурами бизнес-сущностей

Платформа **Openlcar** содержит функционал, позволяющий без изменения исходных кодов, без необходимости перезапуска приложения, силами прикладных специалистов-аналитиков, описать прикладные бизнес-сущности, которые моделируют автоматизируемую предметную область. При этом возможно:

- определение структуры данных сущности (набор полей)

- описание взаимоотношений между сущностями, в т.ч:
 - описание связей "многие к одному" при использовании справочников и классификаторов
 - описание связей "один ко многим" при описании сложных объектов, содержащих в т.ч. списки других объектов
- описание правил доступности и видимости элементов структуры данных
- декларативное (не визуальное) описание правил формирования экранных форм бизнес-сущности
- определение операций (действий), доступных над описываемым объектом
- описание возможных параметров фильтрации при выводе список объектов

На основании такого описания бизнес-сущностей по время выполнения автоматически производится:

- формирование структуры БД
- формирование экранных форм списков объектов с учетом полномочий пользователей
- формирование экранных форм объектов, в т.ч. при использовании этих объектов в экранных формах шагов бизнес-процессов.

2.7.Подсистема формирования пользовательского интерфейса

В настоящее время в решениях можно выделить следующие типы экранных форм :

№	Тип Экранной формы	Способ формирования	Источник метаданных	Управляемо ИТ-службой заказчика
1	Экранная форма заданий (шагов бизнес-процесса)	Автогенерация	Описание бизнес-процесса	Да
2	Экранная форма системных объектов, в т.ч. списки, фильтры	Автогенерация	design-time модели (csm)	Нет
3	Экранная форма прикладных объектов (бизнес-сущности), в т.ч. списки, фильтры	Автогенерация	модели сущностей	Да
4	Экранная форма параметров отчетов	Автогенерация	Описание запроса для отчета	Да
5	Специальные общесистемные Экранные формы (Мои документы, Мои задания и т.п.)	Кодируется	Метаданные не используются	Нет

6	Экранная форма параметров при вызове утилит	Автогенерация	Аннотации вызываемого метод	Да, для утилит, реализуемых прикладными компонентами
---	---	---------------	-----------------------------	--

Для каждого типа автогенерируемых экранных форм существует свои метаданные, обычно как часть метаданных собственно объекта, экранная форма которого генерируется. Платформа **Openlcar** использует подход, когда основная масса экранных форм формируется путем автогенерации на основе метаданных.

Преимущества автогенерации Экранных форм

- Скорость разработки - после описания структуры объекта его экранная форма доступна сразу
- Универсальность - все экранные формы генерируются единообразно, по единым правилам. Для добавления функционала/исправления ошибки необходимо внести изменения в генератор на основе метаданных, и эти изменения будут отражены сразу во всех экранных формах такого типа.

Прочие особенности

Метаданные, используемые для генерации экранных форм, в настоящее время неотделимы от структуры объекта (сущности, шага бизнес-процесса), это не позволяет форме и содержанию "разъезжаться". За поведение экранной формы каждого типа отвечает свой (невизуальный) системный компонент, который контролирует, как и откуда должны быть загружены данные, отображаемые на экранной форме, куда и как их следует сохранять, какая должна быть реакция на действия пользователя (нажатия на кнопки и т.п.). В текущее время эта логика тесно связана с логикой генерации экранных форм по метаданным.

2.8.Создание прикладного решения

При реализации конкретного прикладного решения можно выделить следующие компоненты:

- Бизнес-процессы - определяют взаимодействие человеческих ресурсов и программных средств при выполнении ими бизнес-задач (human workflow, оркестровка сервисов разл. приложений в случае гетерогенного решения (поддержка SOA))
- Бизнес-сущности - объекты, специфичные для конкретной предметной области (документы, справочники и т.п.)
- Прикладная бизнес-логика - оперирует бизнес-сущностями и определяет их поведение (правила нумерации документов, ...)
- Общесистемные сервисы - повторно используемые, не относящиеся к

конкретной предметной области (управление правами доступа, работа с эл. почтой, протоколирование, механизмы построения отчетов, средства импорта/экспорта)

2.9. Подсистема формирования отчетности

Формирование отчетов основывается на описании запроса, использовании одного из поддерживаемых отчетных движков (шаблонизаторов) и подготовке шаблона, поддерживаемого выбранным отчетным движком.

Подсистема выпуска отчетов оперирует следующими объектами:

- Запрос — отвечает за подготовку данных (система поддерживает запросы различных типов: sql, ejb ql, groovy script, при описании запроса необходимо указать выходные поля запроса, входные параметры запроса, текст запроса и тип запроса)
- Отчет — связывает запрос, отчетный движок и шаблон документа;
- Отчетный движок — системный программный компонент, который формирует документ на основе шаблона и набора данных, предоставленных запросом.
- Шаблон - представляет собой один или несколько файлов, подготовленных специальным образом в зависимости от типа шаблона (отчетного движка)

Платформа **OpenIcar** поддерживает следующие типы отчетов (шаблонизаторов):

- JasperReports (отчетный движок jasperReportsEngine) - для отчетов общего назначения, с возможностью вывода в различных форматах
- RTF-шаблоны в связке с библиотекой RTFTemplate - для подготовки текстовых документов с возможностью последующего редактирования
- XLT-шаблоны (отчетный движок jUniPrintReportsEngine) - для подготовки табличных документов в формате MS Excel с возможностью последующего редактирования

Каждый тип из перечисленных выше имеет свой способ и формат описания шаблонов, поддерживает вывод отчета в том или ином формате. При разработке отчета в зависимости от типа движка используется свой инструмент для подготовки шаблона.

Создание отчета состоит из след. этапов:

- Разработка шаблона - производится аналитиком или специалистом по разработке отчетов с использованием спец. инструментов
- Разработка запроса - производится аналитиком или прикладным программистом средствами платформы **OpenIcar**
- Описание параметров

- Отладка
- Установка в работающую систему.

2.10. Средства оповещения пользователей

Платформа *OpenIcar* содержит возможности:

- выполнения периодических заданий
- оповещение пользователей по email. На основе этих компонент в прикладных проектах реализуются рассылки пользователям с оповещениями.