

Интернет-журнал «Наукоедение» ISSN 2223-5167 <http://naukovedenie.ru/>

Том 8, №3 (2016) <http://naukovedenie.ru/index.php?p=vol8-3>

URL статьи: <http://naukovedenie.ru/PDF/139TVN316.pdf>

DOI: 10.15862/139TVN316 (<http://dx.doi.org/10.15862/139TVN316>)

Статья опубликована 30.06.2016.

Ссылка для цитирования этой статьи:

Олейник Д.С., Оглоблин Д.И. Исследование принципов построения программы PLM Teamcenter с целью расширения её функционала // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 8, №3 (2016) <http://naukovedenie.ru/PDF/139TVN316.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/139TVN316

УДК 658.5

Олейник Дмитрий Сергеевич

ГОУ ВПО «Московский государственный технический университет имени Н.Э. Баумана», Россия, Москва¹
Студент 4 курса
E-mail: dimitr95@mail.ru

Оглоблин Дмитрий Игоревич

ГОУ ВПО «Московский государственный технический университет имени Н.Э. Баумана», Россия, Москва
Кандидат технических наук, доцент
E-mail: ogldi@mail.ru
РИНЦ: http://elibrary.ru/author_profile.asp?id=797288

Исследование принципов построения программы PLM Teamcenter с целью расширения её функционала

Аннотация. Расширение функционала программы PLM способствует ее адаптации под условия конкретного предприятия, ускорению решения специфических задач в области проектирования, подготовки производства, технического обслуживания выпускаемой продукции и других этапов жизненного цикла изделий. Авторами изучена структура программы PLM Teamcenter и предложено решение проблемы создания и запуска сторонних приложений (плагинов) в рабочем пространстве программы. Использован полный клиент двухслойной архитектуры PLM Teamcenter и дополнительный каркас в виде платформы Eclipse, синхронизированный с работой полного клиента. Проведено исследование возможностей совместной работы этих сред, предполагающее запуск полного клиента Teamcenter из Eclipse. Рассмотрены проблемы сочетания версии PLM Teamcenter, комплекта разработчика приложений на языке Java (Java Development Kit) и интегрированной среды разработки Eclipse и задания настроек проекта. Создана в рамках виртуальной машины, отлажена и исследована среда запуска сторонних модулей на основе указанных выше частей. Представлен фрагмент Java-кода обработчика, запускаемый в процессе сессии работы программы Teamcenter и успешно отображаемый в пространстве Teamcenter. Таким образом, получен инструмент для встраивания сторонних приложений, позволяющий расширить возможности программы при адаптации к конкретному предприятию и при ограничении связей с разработчиком.

¹ 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1

Аннотация должна отражать не только актуальность темы, а содержать краткое изложение материала статьи. Сделайте, пожалуйста, упор на это. Употребляйте выражения «Автором представлен ...», «В статье ...» и т.д.

Ключевые слова: программа PLM; расширение функционала; клиент-серверная архитектура; среда разработки Eclipse; инструмент для встраивания стороннего модуля (плагины); настроенный проект Eclipse на целевой платформе Teamcenter

Введение

Программы PLM доказали свою эффективность, предлагая хранение данных, их обслуживание, а также непрерывный набор бизнес-решений для поддержки создания, управления, распределения и использования интеллектуальных активов предприятия. PLM Teamcenter компании Siemens PLM Software признана в России одной из лучших и используется на крупных сборочных предприятиях и в КБ в авиастроительной, автомобильной, машиностроительной отраслях. Описывая процессы проектирования, подготовки производства, производитель программы формирует шаблоны, формы в общем виде с максимальной подробностью и предлагает инструменты внутренней настройки. Однако, всегда имеются специфические задачи, которые не укладываются в известные шаблоны. Это приводит к необходимости расширения функционала путем программирования в среде Teamcenter, исходя из конкретных нужд предприятия.

В отличие от других производителей подобных программ, Siemens PLM Software выпускает руководства по «кастомизации» [1] и предоставляет некоторые сведения о строении программ. Например, известны возможности программирования с целью расширения функционала в программе CAD NX от Siemens [2]. Здесь, в основном, решаются вопросы геометрических построений при помощи создания в среде Visual Studio приложений, запуска их из текущей сессии в пространстве процессов NX, а также отображения результатов работы в графическом окне NX.

В данной работе ставится задача исследования архитектуры программы PLM Teamcenter с целью расширения функционала, а также поиск путей формирования и запуска приложений стороннего разработчика в пространстве процесса текущей сессии Teamcenter.

Архитектура программы Teamcenter и средства расширения функционала

Программа PLM Teamcenter представляет собой клиент-серверную архитектуру, условно изображенную авторами на рис. 1.

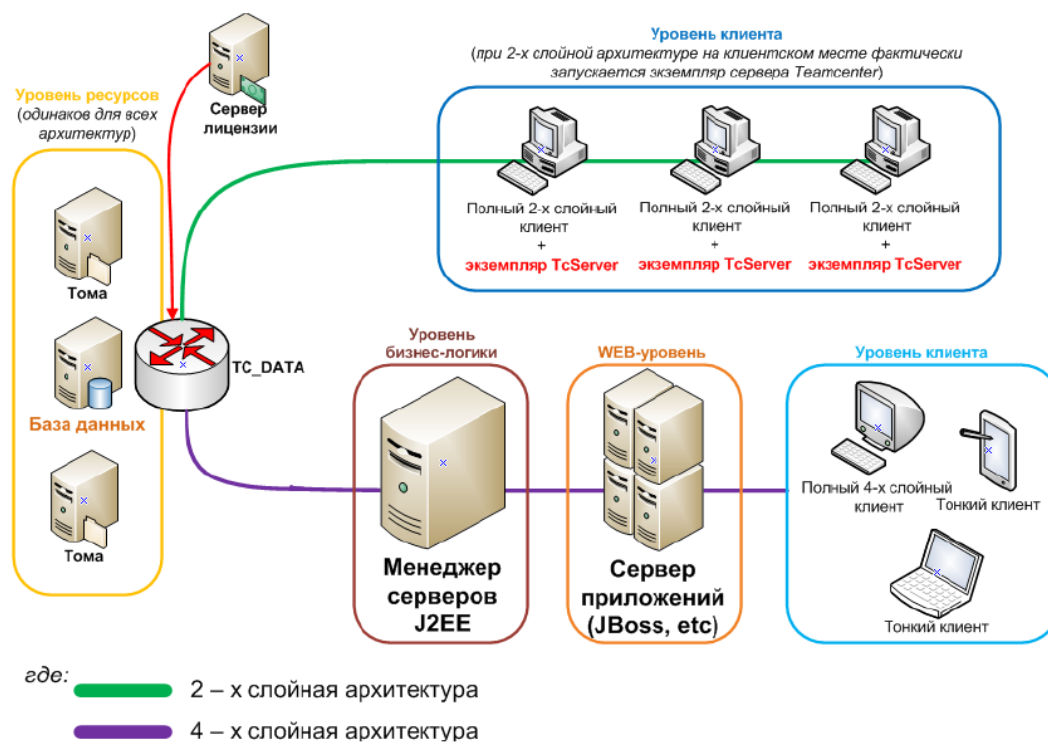


Рисунок 1. Архитектура программы Teamcenter

Существует два варианта развертывания Teamcenter: 4-х слойная архитектура, включающая уровень ресурсов, уровень бизнес-логики, WEB-уровень, уровень клиентов (рис. 1) и 2-х слойная, включающая в себя только уровень клиента, состоящий из толстого клиента и сервера Teamcenter с необходимыми исполняемыми и библиотечными файлами, и уровень ресурсов, включающий СУБД и файловые хранилища. Двухслойная архитектура наиболее подходит для исследования задачи запуска сторонних приложений, поэтому она и рассматривается далее.

В двухслойной архитектуре серверная часть (бизнес-логика) Teamcenter и толстый клиент запускаются непосредственно на клиентском месте. Толстый клиент и сервер могут быть подвержены кастомизации, под которой понимается расширение функционала программы.

Серверный уровень представляет собой библиотеки, консольные утилиты и может модернизироваться с использованием библиотек ИТК (Integration Tool Kit - это некий интеграционный инструмент, позволяющий передавать и обрабатывать данные между двумя системами) и языка C/C++. Клиентский уровень представляет собой интерфейсную часть и может модернизироваться средствами языка Java. Полный клиент имеет единообразный пользовательский интерфейс, систему динамического конфигурирования установки, механизмы доступа и обмена информацией через интернет, открытый программный интерфейс (API) для интеграции приложений в каркас полного клиента. Информация производителя, раскрывающая состав клиент-серверной архитектуры и использованные языки программирования, приведена на рис. 2 [7].

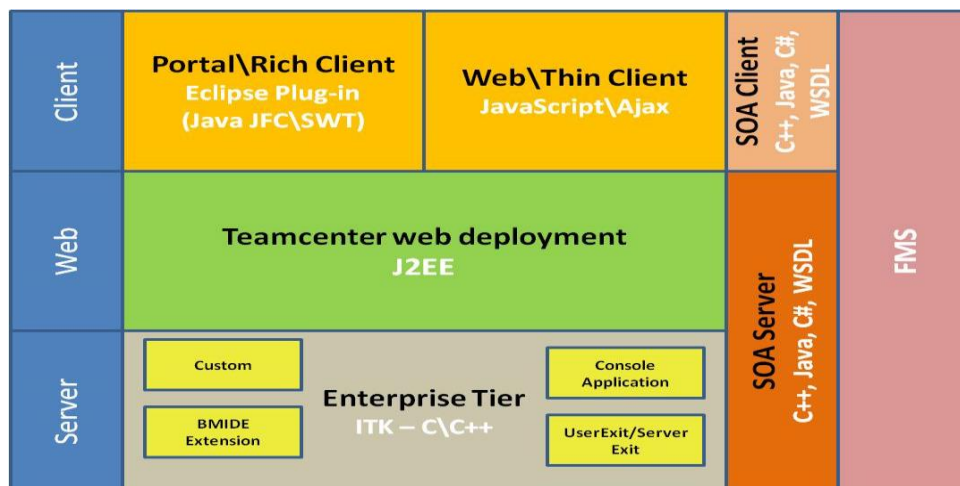


Рисунок 2. Лингвистическое обеспечение PLM Teamcenter

Полный клиент является порталом к данным и процессам и предоставляет каркас для встраивания дополнительного функционала и его запуска, которым является платформа Eclipse.

Итак, средствами программной разработки можно решать задачи по расширению функционала системы Teamcenter [1], используя: а) библиотеку ИТК и язык C/C++ для расширения функционала серверных компонентов Teamcenter; б) язык программирования Java и среду разработки Eclipse для расширения функционала полного клиента Teamcenter; в) сервис-ориентированную архитектуру (SOA) для написания дополнений к Teamcenter как в составе полного клиента, так и самостоятельных приложений.

Исследовательская часть работы

Подготовительная часть

Перед началом формирования собственного приложения в рамках Teamcenter версии 8.3 необходимо установить следующие компоненты:

1. Программу Teamcenter с 2-х слойным вариантом развертывания. Чтобы иметь возможность восстановить систему, в которой ведется разработка программного обеспечения, а также ограничить доступ к рабочей системе, наилучшим образом подойдет создание виртуальной машины.
2. Комплект разработчика приложений на языке Java (JDK, Java Development Kit).
3. Интегрированную среду разработки Eclipse и настроить её для запуска полнофункционального клиента. Последнее предполагает задание определенных настроек проекта и запуск полного клиента Teamcenter из Eclipse.

Рассмотрим третий пункт подробнее:

- а) Необходимо установить JDK той же версии, которая использована для написания клиента Teamcenter, что обеспечивает в будущем полноценное взаимодействие разработанного приложения с Teamcenter. Можно использовать версию, указанную в документации по кастомизации с учетом версии Teamcenter. В нашем случае это JDK 1.6.0_14.

- b) Установить среду разработки Eclipse SDK в версии, которая указывается в документации по кастомизации конкретной версии Teamcenter. В нашем случае это Eclipse 3.5 Software Development Kit.

После скачивания нужно распаковать архив с Eclipse в желаемую директорию и создать bat-файл, который устанавливает переменные среды, запускает сервер и запускает Eclipse. Пример файла приведен ниже.

```
setFMS_HOME=TC_ROOT\fcc
setJAVA_HOME=TC_ROOT\portal\jre
setJRE_HOME=TC_ROOT\portal\jre
setCLASSPATH=TC_ROOT\portal
setPATH=%FMS_HOME%\bin;%FMS_HOME%\lib;TC_ROOT\portal;%
PATH%
start "TAOImR"/mincmd/c "TC_ROOT\iopservers\start_imr.bat"
Eclipse-install-directory\eclipse.exe-vmjdk-install-directory\bin\javaw
```

При запуске созданного bat-файла Eclipse запросит рабочую директорию, где будут находиться создаваемые проекты и связанная с ними информация. И после ее указания откроется среда Eclipse.

Задание настроек проекта

При настройке проекта необходимо указать путь к JRE, чтобы Eclipse мог запускать Java-приложения, а также целевую платформу, под которую мы будем разрабатывать дополнительные модули (плагины). В нашем случае целевой платформой является клиент Teamcenter, поэтому необходимо указать директорию, в которой находятся все java-пакеты, используемые при работе клиента. Примерный порядок действий описан в [1, с. 13].

В результате получен настроенный проект Eclipse на целевой платформе TC, в который может быть подгружено стороннее приложение как в обычном режиме, так и в режиме отладки.

Запуск полного клиента Teamcenter из Eclipse необходим для проверки работы написанных модулей без обновления модулей действующего клиента. Плагин загружается динамически и виден в списке действующих установленных модулей. Плюсами такого запуска является быстрая проверка написанного приложения и возможность его отладки в ходе работы. Процедура подобного запуска описана в [1, с. 14].

Создание проекта в Eclipse и выполнение его в пространстве процессов Teamcenter

Полный клиент Teamcenter создан на основе Eclipse, поэтому уточним, как устроена структура Eclipse. Платформа Eclipse RCP предоставляет каркас для модульного расширения функционала приложения за счет написания собственных модулей (плагинов) [3, 4]. При установке полного клиента устанавливаются пакеты Java (JAR-файлы), которые являются дополнительными модулями (плагинами) к Eclipse. Пакеты располагаются в директории TC_ROOT\portal\plugins, а имена пакетов начинаются с **com.teamcenter**. Эти файлы составляют основу полного клиента и содержат ресурсы необходимые для его запуска. При добавлении нового модуля он добавляется в эту директорию.

Большинство задач кастомизации требуют написания Java-кода и его подключения к Eclipse [5, 6]. На основе примера – запуска приложения, выводящего на экран программы Teamcenter надпись «Hello, Eclipse word» [1, с. 67], создадим проект и рассмотрим ключевые файлы в структуре проекта.

После установки дополнительных модулей к Eclipse создается проект (рис. 3).

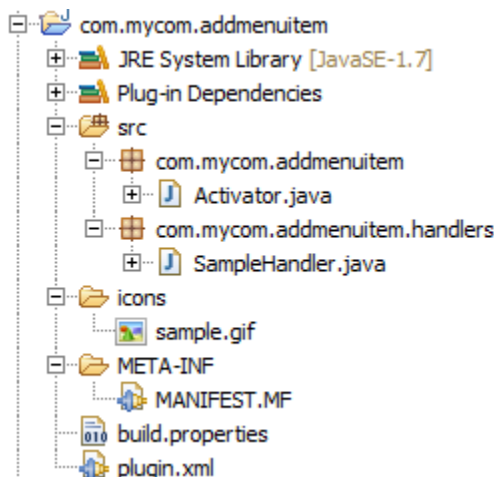


Рисунок 3. Структура тестового проекта в Eclipse

Чтобы проверить его работоспособность и наличие, нужно его запустить, нажать Run. Открывается клиент Teamcenter. Данное тестовое приложение добавляет в вкладку меню с названием «SampleMenu» внутри которой расположена команда «SampleCommand» (рис. 4), а также вносит эту команду в панель инструментов обозначенную иконкой (рис. 5).

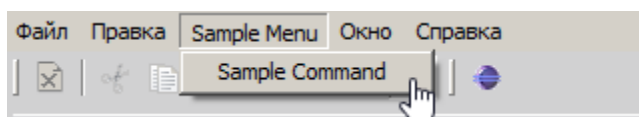


Рисунок 4. Команда в меню



Рисунок 5. Команда в панели инструментов

Выбор данной команды выводит информационное окно с сообщением «Hello, Eclipse word» и подтверждает работоспособность созданного приложения (рис. 6).

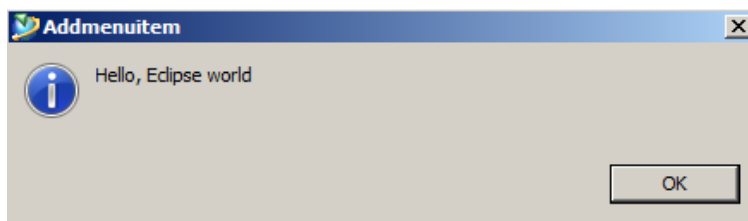


Рисунок 6. Информационное окно

Для более сложных проектов способом проверки наличия добавленного приложения является просмотр установленных модулей. В меню выбирается Справка -> О программе -> Сведения об установке. Значение столбцов добавленного приложения в открывшейся таблице (рис. 7) соответствует значениям, указанным в файле проекта MANIFEST.MF, который содержит все настройки проекта (рис. 8).

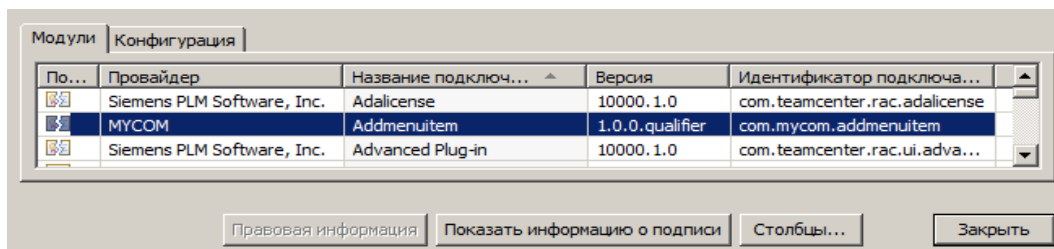


Рисунок 7. Сведения об установленных модулях Teamcenter

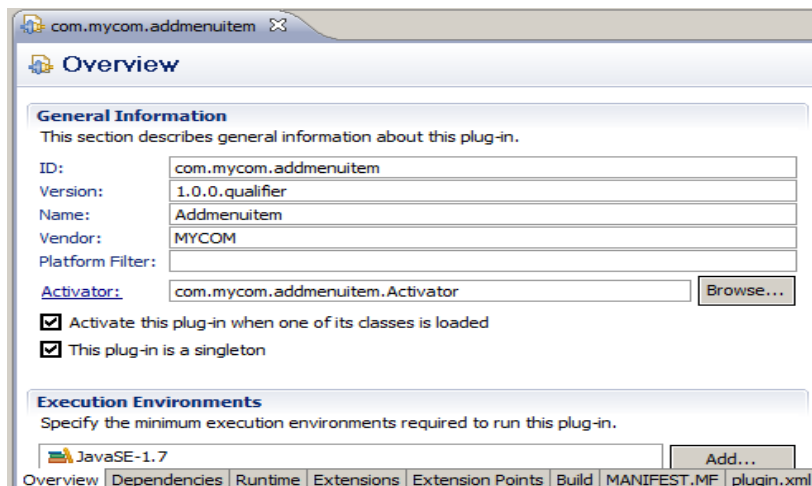


Рисунок 8. Настройки файла MANIFEST.MF

Файл MANIFEST.MF позволяет редактировать общие настройки проекта (рис. 8), подключать java-пакеты других модулей (рис. 9), настраивать сборку проекта (рис. 10) и многое другое.

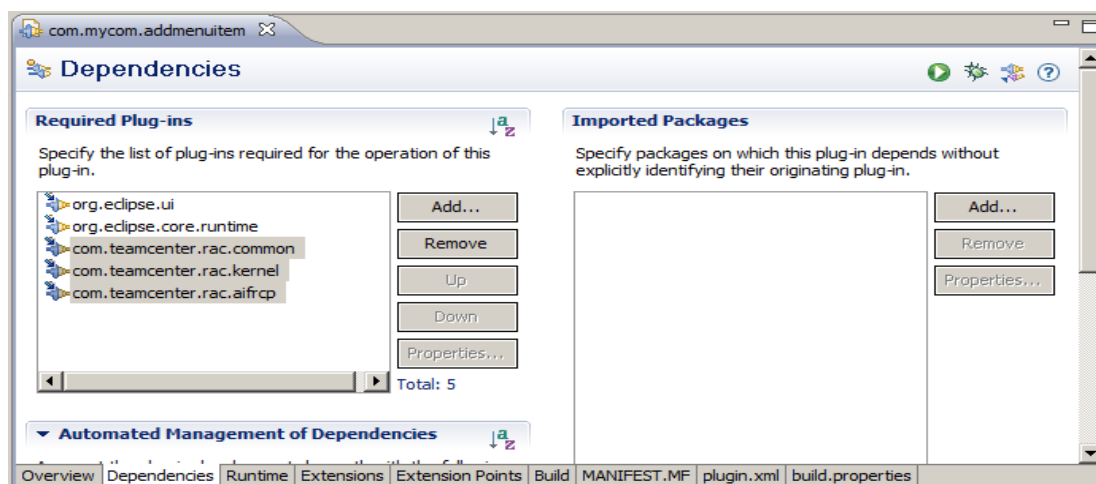


Рисунок 9. Подключение дополнительных модулей

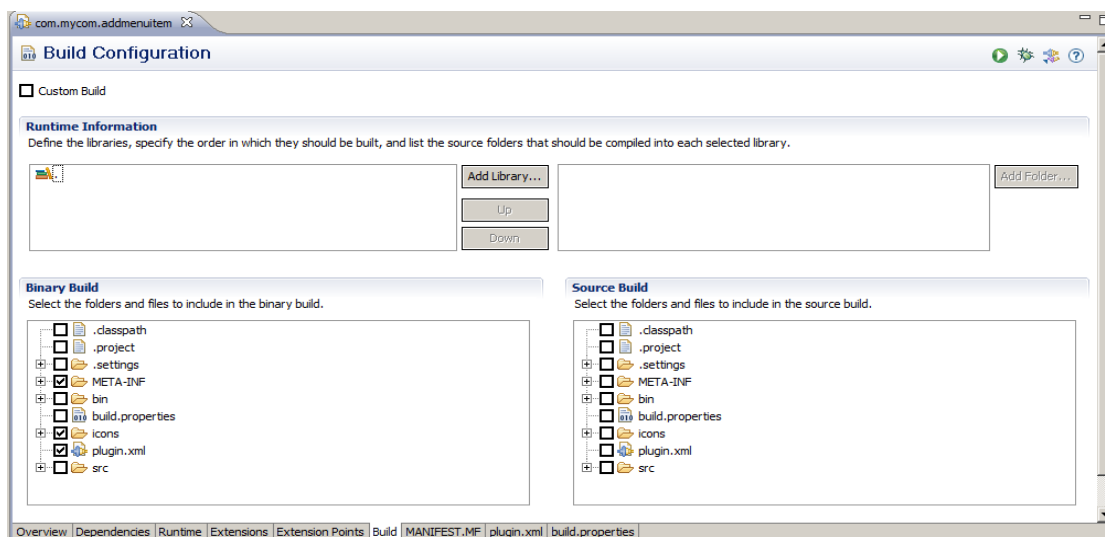


Рисунок 10. Настройка сборки проекта

Файл plugin.xml позволяет подключать расширения и управлять ими (рис. 11).

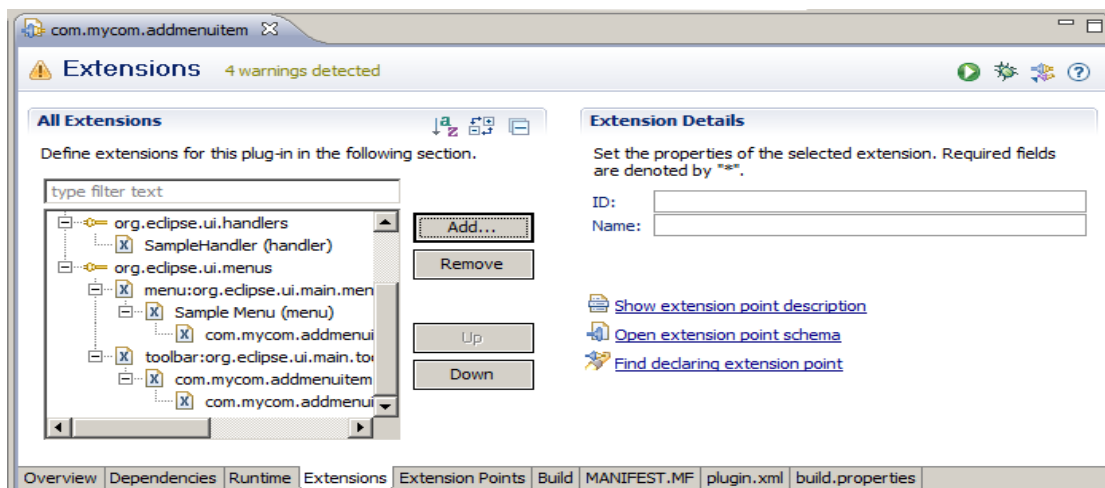


Рисунок 11. Работа с расширениями

Java-код обработчика, именуемый в данном примере SampleHandler.java и запускаемый по нажатию на созданную команду, находится в пакете **com.mycom.addmenuitem.handlers** (рис. 12), где на основе абстрактного класса обработчика AbstractHandler реализован свой класс обработчика SampleHandler.

Выполнение программы начинается с вызова функции execute().

```
package com.mycom.addmenuitem.handlers;

import org.eclipse.core.commands.AbstractHandler;
import org.eclipse.core.commands.ExecutionEvent;
import org.eclipse.core.commands.ExecutionException;
import org.eclipse.ui.IWorkbenchWindow;
import org.eclipse.ui.handlers.HandlerUtil;
import org.eclipse.jface.dialogs.MessageDialog;

/**
 * Our sample handler extends AbstractHandler, an IHandler base class.
```



```
* @see org.eclipse.core.commands.IHandler
* @see org.eclipse.core.commands.AbstractHandler
*/
publicclass SampleHandler extends AbstractHandler {
    /**
     * The constructor.
     */
    public SampleHandler() {
    }

    /**
     * the command has been executed, so extract extract the needed information
     * from the application context.
     */
    public Object execute(ExecutionEvent event) throws ExecutionException {
        IWorkbenchWindow window = HandlerU-
        til.getActiveWorkbenchWindowChecked(event);
        MessageDialog.openInformation(window.getShell(), "AddmenuItem", "Hello,
        Eclipse world");
        returnnull;
    }
}
```

Рисунок 12. Реализация класса обработчика SampleHandler

Ядро (Kernel) – часть полного клиента, которая выполняет функции из ИТК в среде полного клиента. «Сердцем» ядра является объект «Сессия». Можно получить ссылку на сессию или из приложения или из объекта, используя метод getSession(). Приложение может работать с многими типами объектов в течение процесса своего выполнения – сессии.

Немного изменив код функции execute(), получаем ссылку на объект текущей сессии (рис. 13). В качестве примера используем имя пользователя текущей сессии и выведем его в информационном сообщении.

```
public Object execute(ExecutionEvent event) throws ExecutionException {
    IWorkbenchWindow window = HandlerU-
    til.getActiveWorkbenchWindowChecked(event);

    //Получаемтекущуюсессию
    TCSession tcSession=(TCSession) AIFUtility.getCurrentApplication().getSession();
    //записываемимятекущегопользователя
    String userName=tcSession.getUserName();

    MessageDialog.openInformation(window.getShell(), "AddmenuItem", "Hello,
    "+userName+"!");

    returnnull;
}
```

Рисунок 13. Получение ссылки на объект текущей сессии

Для проверки войдем в Teamcenter под именем пользователя infodba и, запустив добавленную команду, получим сообщение (рис. 14).



Рисунок 14. Информационное сообщение

Заключение

Совместная работа полного клиента Teamcenter и дополнительного каркаса в виде платформы Eclipse, позволила сформировать инструмент для встраивания стороннего модуля (плагины) и его запуска в рамках текущей сессии Teamcenter (рис. 15). Этот инструмент позволяет «дописать» программу PLM Teamcenter для выполнения новых неизвестных функций, автоматизировать повторяющиеся операции и т.п. вне внутренних настроек.

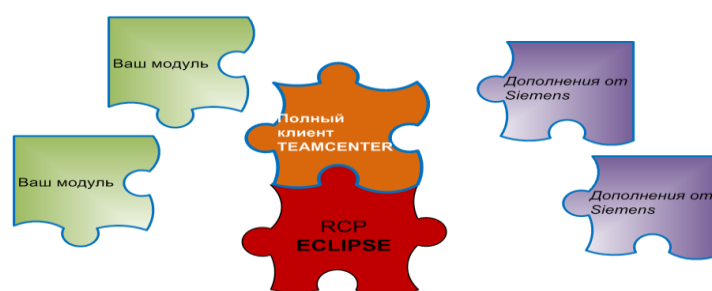


Рисунок 15. Полученный инструмент запуска стороннего модуля

Получена методика создания и запуска сторонних модулей (плагинов) в среде Eclipse, выполняющихся в рамках целевой платформы Teamcenter и предназначенных для расширения функционала данной программы PLM.

ЛИТЕРАТУРА

1. Teamcenter 8.3 Rich Client Customization Programmer's Guide Publication Number PLM00075 F [Электронный ресурс] Global Technical Access Center (GTAC). URL: https://support.industrysoftware.automation.siemens.com/docs/teamcenter/8.3/PDF/pdf/rich_client_customization_programmers_guide.pdf (дата обращения: 03.05.2016).
2. Тихомиров В.А. Разработка приложений для Unigraphics на языке C. Владивосток, Дальнаука, 2011. 422 с.
3. URL: <https://www.ibm.com/developerworks/ru/library/os-eclipse-plugindev1/> (дата обращения 30.05.2016).
4. URL: <https://www.ibm.com/developerworks/ru/library/os-eclipse-plugindev2/> (дата обращения 30.05.2016).
5. URL: <https://m.habrahabr.ru/post/139340/> (дата обращения 30.05.2016).
6. Машнин Т.С. Eclipse: разработка RCP-, Web-, Ajax- и Android-приложений на Java. СПб.: БХВ-Петербург, 2013. 384 с.
7. URL: <http://teamcenterplm.blogspot.ru/2013/08/introduction-to-teamcenter-customization.html> (дата обращения 30.05.2016).

Oleynic Dmitry Sergeevich

Bauman Moscow state technical university, Russia, Moscow
E-mail: dimitr95@mail.ru

Ogloblin Dmitry Igorevich

Bauman Moscow state technical university, Russia, Moscow
E-mail: ogldi@mail.ru

A study the principles of the Teamcenter PLM programs in order to enhance the functionality

Abstract. The extension of the functionality of the program PLM facilitates its adaptation to the specific conditions of the enterprise, accelerate solve specific problems in the field of design, production preparation, maintenance of products and other stages of the life cycle of products. The Authors studied the structure of the program and PLM Teamcenter provides a solution to the problem of creating and running third-party applications (plugins) in the workspace. Used the rich client, double-layered architecture PLM Teamcenter and additional frame within the Eclipse platform that are synchronized with the rich client. Conducted a feasibility study of coexistence between these environments, involving the launch of the rich client Teamcenter from Eclipse. The problems of combination version PLM Teamcenter development kit applications in Java (Java Development Kit) and IDE Eclipse setting project. Created within the virtual machine, debugged and investigated the environment to run third-party modules based on the above parts. A fragment of Java code in a handler that is triggered during the session of operation of the program Teamcenter and successfully displayed in the space of Teamcenter. Thus, the obtained tool for embedding third-party applications, allowing to expand opportunities of the program during adaptation to a particular company and by limiting relations with the developer.

Keywords: program PLM; the extension functionality; the client-server architecture; development environment Eclipse; tools for embedding third-party module (plug-in); configured Eclipse project on the target platform Teamcenter

REFERENCES

1. Teamcenter 8.3 Rich Client Customization Programmer's Guide Publication Number PLM00075 F [Elektronnyy resurs] Global Technical Access Center (GTAC). URL: https://support.industrysoftware.automation.siemens.com/docs/teamcenter/8.3/PDF/pdf/rich_client_customization_programmers_guide.pdf (data obrashcheniya: 03.05.2016).
2. Tikhomirov V.A. Razrabotka prilozheniy dlya Unigraphics na yazyke C. Vladivostok, Dal'nauka, 2011. 422 s.
3. URL: <https://www.ibm.com/developerworks/ru/library/os-eclipse-plugindev1/> (data obrashcheniya 30.05.2016).
4. URL: <https://www.ibm.com/developerworks/ru/library/os-eclipse-plugindev2/> (data obrashcheniya 30.05.2016).
5. URL: <https://m.habrahabr.ru/post/139340/> (data obrashcheniya 30.05.2016).
6. Mashnin T.S. Eclipse: razrabotka RCP-, Web-, Ajax- i Android-prilozheniy na Java. SPb.: BKhV-Peterburg, 2013. 384 s.
7. URL: <http://teamcenterplm.blogspot.ru/2013/08/introduction-to-teamcenter-customization.html> (data obrashcheniya 30.05.2016).