

ШАМАЕВ А. В., КАДАНЦЕВ А. А.

**РАЗРАБОТКА ДОПОЛНИТЕЛЬНОГО ПРОГРАММНОГО МОДУЛЯ
ДЛЯ PLM-СИСТЕМЫ TEAMCENTER (SIEMENS)**

Аннотация. В статье описан процесс создания дополнительного программного модуля для системы управления жизненным циклом изделия промышленного производства Teamcenter, автоматизирующего процесс создания структуры документа «Ведомость конфигурации». Рассмотрены этапы моделирования, проектирования и реализации данного модуля.

Ключевые слова: жизненный цикл, ведомость конфигурации, изделие, кастомизация программы, Teamcenter, Eclipse, код.

SHAMAIEV A. V., KADANTSEV A. A.

**DEVELOPMENT OF ADDITIONAL SOFTWARE MODULE
FOR TEAMCENTER PLM SYSTEM (SIEMENS)**

Abstract. This article considers the process of creating an add-on software module for the Teamcenter industrial product lifecycle management system, which automates the process of creating the structure of the «Configuration Sheet» document. The authors describe the stages of modeling, design and implementation of this module.

Keywords: product lifecycle, configuration sheet, machine-building product, program customization, Teamcenter, Eclipse, code.

Введение. Аббревиатура PLM (Product Lifecycle Management) – это общепринятое сокращенное название совокупности технологий, применяемых для организации управления жизненным циклом продукта (изделия). PLM-системой называют организационно-техническую систему, которая обеспечивает управление всей информацией о продукте и связанных с ним процессах на протяжении всего жизненного цикла этого продукта, начиная с его проектирования и производства и завершая прекращением его эксплуатации.

PLM-системы доказали свою эффективность, предлагая эффективные способы хранения и обработки данных, а также комплексный набор бизнес-решений, предназначенных для поддержки процессов создания, управления, распределения и использования интеллектуальных активов предприятия. Описывая процессы проектирования и подготовки производства, производитель PLM-системы (которая представляет собой специализированный комплекс программ) предоставляет максимально детализированные шаблоны, формы и предлагает инструменты для их настройки. Однако всегда существует ряд задач, к которым невозможно применить готовые шаблоны. Для их решения

производители таких систем предоставляют пользователям возможность интеграции стороннего программного обеспечения, благодаря чему предприятие, применяющее такую систему, может вести разработку дополнительных программных модулей, которые позволят увеличить скорость работы, минимизировать количество ошибок при выполнении операций, что в итоге приведёт к снижению затрат на выполнение проекта.

Описание PLM-системы Teamcenter. Teamcenter представляет собой многофункциональную платформу, которая, помимо набора стандартных приложений Siemens PLM Software, предназначенных для проектирования и анализа, может включать другие специализированные программы для проектирования продуктов машиностроения, приборостроения, строительства и других областей. Она организует управление инженерными данными и производственными процессами на протяжении всего жизненного цикла продукта. Прежде всего данная платформа предназначена для создания единой корпоративной информационной системы, организующей управление компанией или группой компаний, филиалы которой географически удалены друг от друга [1, с. 32; 2, с. 8].

Ниже перечислены основные возможности Teamcenter, как единой системой управления данными о продукте [1, с. 32]:

- оценка влияния каждого изменения на общую конфигурацию всего продукта;
- создание специально согласованной структуры продукта, которая может послужить основой для анализа и учета изменений;
- проведение экспертизы поставленных и выполненных задач;
- уведомление пользователей, о том, когда должны быть реализованы задачи по внедрению изменений;
- ведение истории изменений продукта на всем протяжении его жизненного цикла;
- проверка сохранности функциональных характеристик после внесения изменений в конфигурацию продукта;
- сравнение ожидаемых и фактически полученных влияний от изменений, внесенных в продукт.

Архитектура Teamcenter. Многоуровневая архитектура системы Teamcenter изображена на рисунке 1.

Синим цветом на рисунке выделены самые низкие (серверные) уровни. Они расположены ближе к базе данных и томам с файлами. Зеленым цветом на рисунке выделены самые высокие (клиентские) уровни. Они расположены ближе к пользователю [3, с. 13].

Существует два варианта развертывания системы Teamcenter, основанные на четырёхуровневой и двухуровневой архитектуре, соответственно.

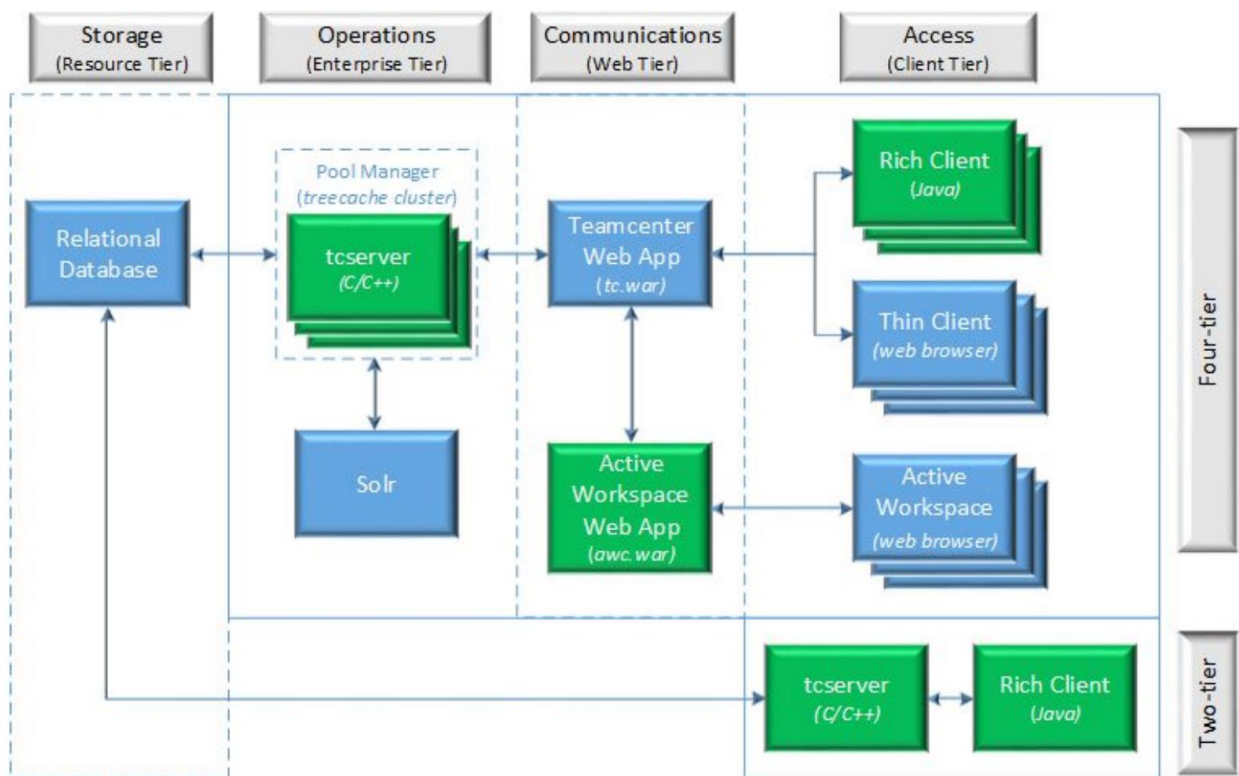


Рис. 1. Многоуровневая архитектура Teamcenter.

Двухуровневая архитектура в большей степени подходит для исследования задачи интеграции сторонних модулей в систему Teamcenter, поэтому в данной работе использовалась именно она.

При применении такой архитектуры серверная часть (бизнес-логика) системы и «толстый» клиент (далее просто клиент) запускаются непосредственно на компьютере клиента. Клиент и сервер могут быть подвержены кастомизации, под которой понимается расширение функционала программы.

Определение ключевых прецедентов программного модуля. Среди совокупности задач, которые можно решить с помощью стандартных методов Teamcenter, есть задачи, требующие создания дополнительных программных модулей. Одной из таких задач является задача формирования ведомости конфигурации.

Каждый сложный продукт подвергается декомпозиции и должен быть разбит на системы, подсистемы и объекты конфигурации.

Ведомость конфигурации (ВК) – проектный документ, который полностью и однозначно описывает в требуемом формате архитектуру и состав конечного продукта в его конкретной конфигурации в соответствии с вариантным правилом и правилом модификации.

В системе Teamcenter, а именно в приложении «Менеджер структуры», отсутствуют готовые инструменты, автоматизирующие процесс формирования данного документа и потому он формируется соответствующими специалистами вручную. Ручное создание

ведомости конфигурации представляет собой длительный и трудоемкий процесс обработки тысячи компонентов, часто приводящий к ошибкам и получению неверного результата. Поэтому задачей данной работы являлась частичная автоматизация этого процесса.

Инструменты для разработки модуля. Перед началом разработки дополнительного программного модуля в рамках Teamcenter версии 11.2 были установлены следующие компоненты [4, с. 13]:

1. Программа Teamcenter с двухуровневым вариантом развертывания.
2. Язык программирования Java и его комплект разработчика приложений (Java Development Kit).
3. Интегрированная среда разработки Eclipse.

Рассмотрим реализацию второго и третьего пункта пошагово:

а) Необходимо установить JDK той же версии, которая используется вместе с клиентом Teamcenter, чтобы обеспечить его полноценное взаимодействие с разрабатываемым модулем. Выберем версию, указанную в документации по настройке клиента на основе версии Teamcenter. В нашем случае это JDK 1.8.0_181.

б) Для работы полнофункционального клиента должна быть установлена и настроена среда разработки Eclipse 3.8 Software Development Kit с версией, указанной в документации по настройке клиента Teamcenter версии 11.2. Настройка клиента включает в себя настройку определенных параметров проекта и запуск клиента Teamcenter через Eclipse.

Подготовка и настройка программного обеспечения. После загрузки и установки вышеуказанных инструментов разработки в каталоге, где размещена среда разработки Eclipse, необходимо создать bat-файл, с помощью которого через параметры командной строки JDK будут устанавливаться значения переменных среды, запуск сервера Teamcenter и среды разработки Eclipse. Пример содержания этого файла приведен ниже [4, с.149].

```
set TPR="TC_ROOT\portal"
if not defined FMS_HOME set FMS_HOME=" TC_ROOT \tccs"
set JAVA_HOME="jre-install-directory"
set JRE_HOME="jre-install-directory"
set ORIGINAL_PATH="%PATH%"
set PATH="%SYSTEMROOT%\system32; %FMS_HOME%\bin; %FMS_HOME%\lib; %TPR%;
%AUX_PATH%"
start "TAO ImR" /min cmd /c "TC_ROOT\iiopservers\start_imr.bat"
Eclipse-install-directory\eclipse.exe -vm "jdk-install-directory\bin\javaw.exe"
```

При выполнении созданного bat-файла среда разработки Eclipse потребует задать путь к рабочему каталогу, в котором будут храниться создаваемые проекты и связанная с ними

информация. В результате выполнения описанных выше шагов среда Eclipse успешно запустится и будет готова к написанию кода программы.

Настройки проекта. При настройке проекта были указаны пути к виртуальной машине Java – Java Runtime Environment (JRE), а также целевая платформа, для которой разрабатывался дополнительный модуль. В нашем случае целевой платформой является клиент Teamcenter, поэтому также нужно указать каталог, в котором находятся все java-пакеты, используемые при работе клиента. Порядок дальнейших действий по настройке проекта подробно описан в [4, с.150].

Результатом выполненных действий является проект Eclipse, настроенный для работы с целевой платформой Teamcenter. Запуск клиента Teamcenter из Eclipse необходим для проверки работы разработанного модуля без необходимости обновления существующих клиентских модулей системы Teamcenter.

Разработанный модуль загружается динамически и отображается в списке установленных на данный момент модулей. Очевидными плюсами такого запуска является возможность быстрой проверки разработанного приложения и возможность его отладки во время работы. Процедура запуска разработанного модуля совместно с модулями Teamcenter подробно описана в [4 с. 151].

Таким образом, после завершения настройки конфигурации проекта можно использовать разработанный модуль совместно с установленными модулями клиента Teamcenter как в обычном режиме, так и в режиме отладки.

Разработка диаграммы последовательности. Структура и логика работы программного модуля определяется его архитектурой. Для её описания, руководствуясь методологией RUP, будем использовать диаграммы языка UML. Начнем с диаграммы последовательности, которая представлена на рисунке 2. Она поэтапно описывает взаимодействие пользователя системы Teamcenter с данным модулем.

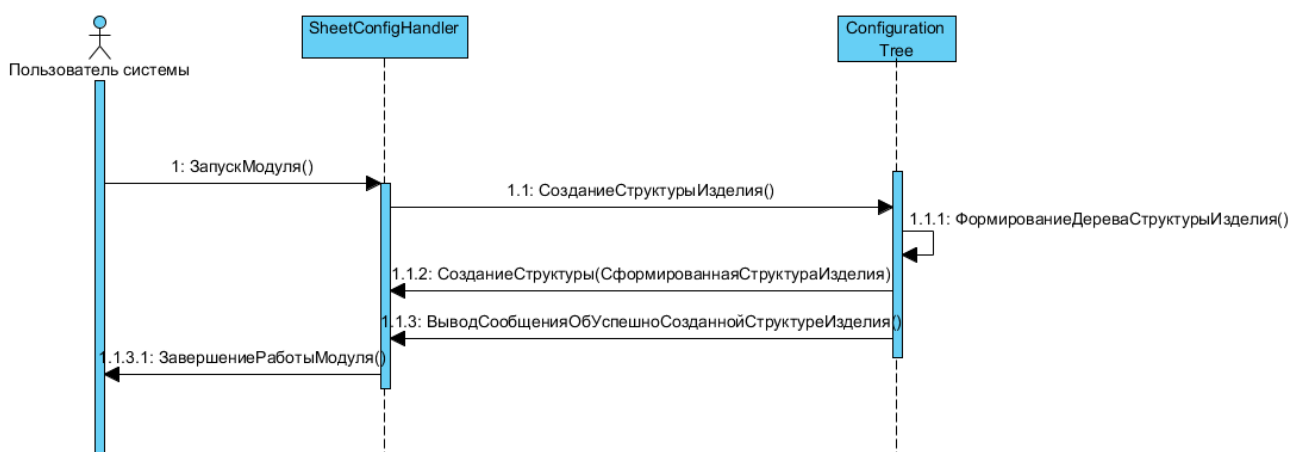


Рис. 2. Диаграмма последовательности.

Разработка диаграммы классов. На рисунке 3 приведена диаграмма классов разработанного программного модуля. Рассмотрим её с точки зрения реализации, то есть дадим краткое описание созданных классов:

- *Activator* – это класс, управляющий жизненным циклом модуля. Он должен реализовывать интерфейс. Среда разработки Eclipse вызывает метод *start()* этого интерфейса сразу после загрузки модуля и метод *stop()* непосредственно перед его выгрузкой;

- *SheetConfigHandler* – это класс-обработчик, который запускает программу при выборе соответствующего пункта меню «Создать Ведомость конфигурации...» из раздела «Отчеты» главного окна приложения «Менеджер структуры» системы Teamcenter;

- *ConfigurationTree* – это класс, который формирует структуру выбранного объекта и включает набор методов, каждый из которых отвечает за определенный этап создания конечного документа;

- *Node* – это класс, который содержит полный набор параметров объекта электронной структуры продукта и методы, применяемые для мониторинга ошибок в нем. Этот класс также реализует стандартный интерфейс языка Java *Comparable*, который позволяет классифицировать объекты по их идентификаторам. Объект класса *Node* может ссылаться как на своего родителя, так и на потомков данного класса.

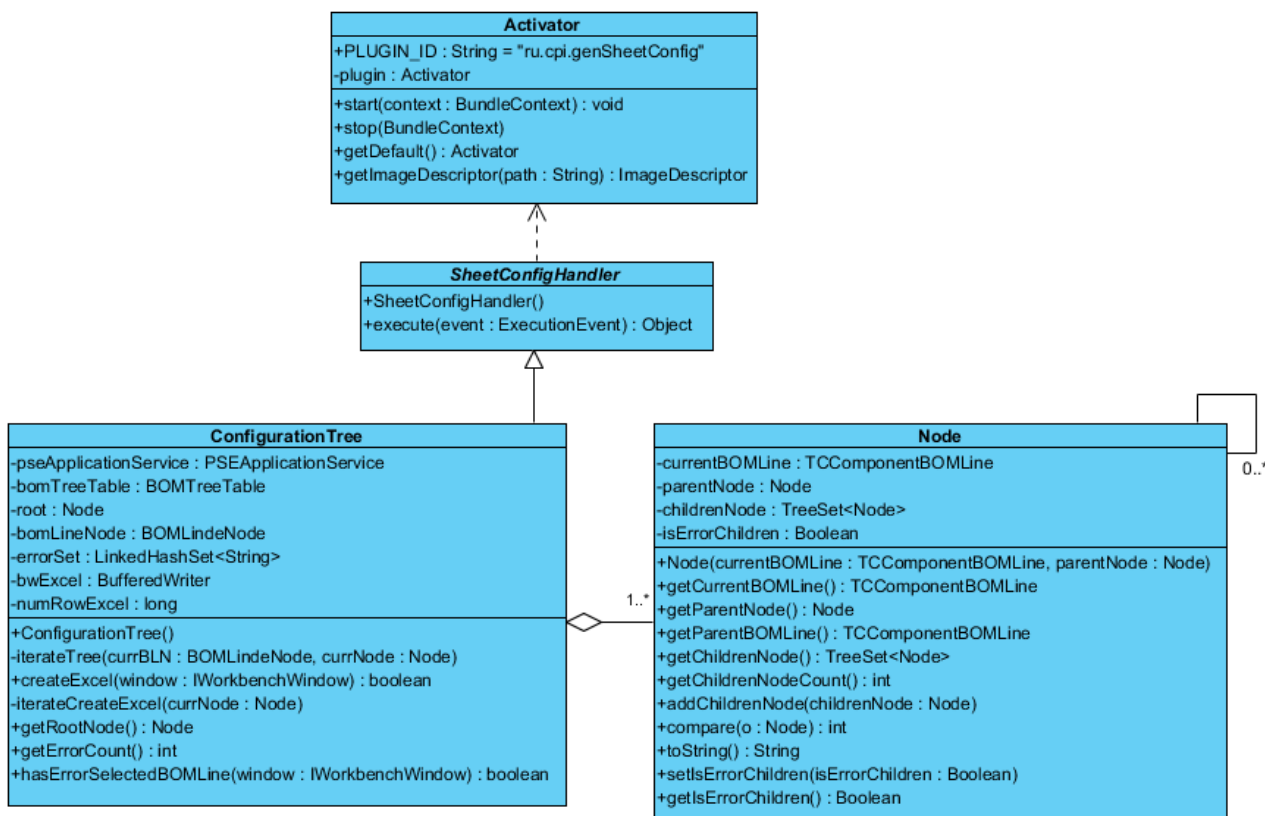


Рис. 3. Диаграмма классов.

Автоматизированное создание структуры документа. Формирование структуры документа «Ведомость конфигурации», выполняемое в PLM-системе Teamcenter с помощью разработанного дополнительного программного модуля, автоматизирующего данный процесс, включает в себя следующие этапы:

1. Запуск PLM-системы Teamcenter и авторизация в ней;
2. Запуск приложения «Менеджер структуры»;
3. Выбор пункта «Добавить структуру...» из раздела «Создание структуры» панели меню главного окна приложения «Менеджер структуры» системы Teamcenter;
4. Ввод необходимого количества уровней и элементов на каждом уровне;
5. Вывод информационного сообщения об успешном создании структуры.

Заключение. Результатом разработки является созданный дополнительный программный модуль, автоматизирующий процесс формирования структуры документа в PLM-системе Teamcenter. Он расширяет функциональность системы Teamcenter и позволяет:

- исключить выполнение рутинных операций;
- раскрыть структуру выбранного продукта до необходимого уровня;
- сократить время на создание структуры ведомости конфигурации (при работе с большими по объему структурами);
- уменьшить количество ошибок в структуре ведомости конфигурации при ее создании до нуля.

СПИСОК ЛИТЕРАТУРЫ

1. Стародубов В. А. Управление конфигурацией: задачи, стандарты и реализация // CAD/CAM/CAE Observer. – 2006. – № 28. – С. 30–33.
2. Тороп Д. Н., Терликов В. В. Teamcenter. Начало работы. – М.: ДМК Пресс, 2011. – 280 с.
3. Teamcenter Getting Started with Customization Publication Number PLM00003 – Global Technical Access Center (GTAC) [Электронный ресурс]. – Режим доступа: https://docs.plm.automation.siemens.com/data_services/resources/tc/11.2/PDF/ru_RU/tdocExt/pdf/getting_started_with_customization.pdf (дата обращения 12.09.2020).
4. Teamcenter 11.2 Rich Client Customization Programmer's Guide Publication Number PLM00075 [Электронный ресурс]. – Режим доступа: https://docs.plm.automation.siemens.com/data_services/resources/tc/11.2/PDF/ru_RU/tdocExt/pdf/client_customization_programmers_guide.pdf (дата обращения 12.09.2020).