

Оглавление

1	Краткий обзор StarUML	2
2	Основные концепции	4
3	Управление проектом	6
3.0.1	Управление элементами с помощью моделей, подсистем и пакетов	9
4	Моделирование с помощью StarUML	11

Глава 1

Краткий обзор StarUML

StarUML™ — программный инструмент моделирования, который поддерживает UML (Унифицированный язык моделирования). StarUML ориентирован на UML версии 1.4 и поддерживает одиннадцать различных типов диаграмм, принятых в нотации UML 2.0. Он активно поддерживает подход MDA (Модельно-управляемая архитектура), реализуя концепцию профилей UML. Среда разработки StarUML™ превосходно настраивается в соответствии с требованиями пользователя и имеет высокую степень расширяемости, особенно в области своих функциональных возможностей. Использование StarUML™, одного из ведущих программных инструментов моделирования, гарантирует достижение максимальной производительности и качества ваших программных проектов.

Инструмент UML, который адаптируется к пользователю

StarUML™ предоставляет максимальную степень адаптации среды разработки пользователя, предлагая настройку параметров, которые могут влиять на методологию разработки программного обеспечения, проектную платформу и язык.

Превосходная расширяемость и гибкость

StarUML™ обеспечивает превосходную расширяемость и гибкость. Он предоставляет механизм аддинов, чтобы расширять свои функциональные возможности. Этот механизм разработан специально, чтобы предоставлять доступ ко всем функциям модели/мета-модели посредством COM Automation и расширять меню и набор свойств элементов. Также, пользователи могут создавать собственные подходы и механизмы согласно своим собственным методологиям. Программа может также быть интегрирована с любыми внешними инструментальными средствами.

Системные требования

Ниже указаны минимальные системные требования, необходимые для функционирования StarUML™.

- Intel ® Pentium ® 233MHz или выше
- Windows ® 2000, Windows XP TM, или выше
- Microsoft ® Internet Explorer 5.0 или выше
- 128 Мбайт RAM (256 МБ рекомендуется)
- 110 Мбайт на жестком диске (150 МБ рекомендуется)
- Устройство CD-ROM
- SVGA или монитор с более высокой разрешающей способностью (1024x768 рекомендуется)
- Мышь или другое устройство позиционирования

Глава 2

Основные концепции

Эта глава вводит фундаментальные концепции, которые требуется знать для эффективного использования StarUML™. Она содержит описание моделей, визуальных элементов и диаграмм, проектов, секций, подходов, фреймворков, модельных фрагментов, их различий относительно разных профилей UML.

- Модель,
- Представление (view) и Диаграмма
- Проект и проектная секция (unit)
- Модуль (module)

Модель, Представление и Диаграмма

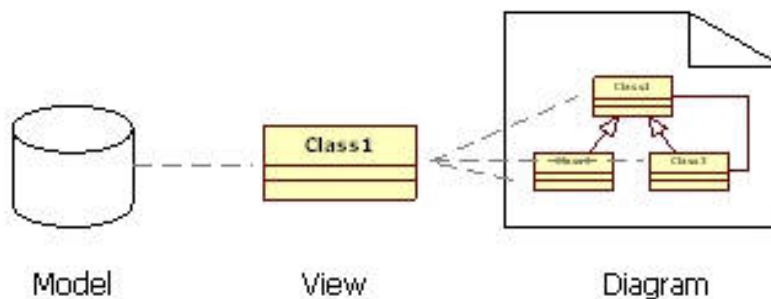


Рис. 2.1: Модель, представление, диаграмма

StarUML™ предполагает ясное понимание концептуального различия между моделями, представлениями и диаграммами. Модель — элемент, который содержит всю информацию о модели программы. Представление — визуальное выражение информации, содержащейся в модели, а Диаграмма — коллекция визуальных образов, которая отображает определенные аспекты проекта.

Проект и проектная секция

Проект

Проект – основная структурная единица в StarUML™. Проект может содержать одну или более программных моделей. Проект - корневой пакет верхнего уровня, который всегда существует в любой программной модели. В общем случае, один проект сохраняется в одном файле.

Структура проекта

Проект содержит следующие суб-элементы:

Модель — элемент, который соответствует одной программной модели.

Подсистема — элемент, который соответствует модели подсистемы.

Пакет — самый общий элемент для группировки других элементов.

Проектный файл

Проектные файлы сохраняются в формате XML и имеют расширение *".UML"*. Все модели, представления и диаграммы, созданные в StarUML™ сохраняются в одном проектном файле. Проект может также быть разделен и сохранен в нескольких проектных секциях. Проектный файл содержит следующую информацию:

- профиль UML, используемый в проекте
- файлы секций, на которые ссылается проект
- информация по всем моделям, содержащимся в проекте
- информация по всем диаграммам и представлениям, содержащимся в проекте

Глава 3

Управление проектом

Эта глава подробно описывает операции по управлению проектом: создание нового проекта, размещение части проекта в секции, создание и импорт фрагментов модели, импорт фреймворков, подключение и исключение профилей UML.

- Управление проектом
- Управление секциями
- Работа с фрагментами модели
- Импорт фреймворка
- Работа с профилями UML

Управление проектом

Создание нового проекта

Чтобы начать разработку программного обеспечения, нужно инициировать новый проект. Вы можете начать абсолютно пустой проект или инициализировать новый проект согласно определённому подходу.

Процедура создания нового проекта #1 — New Project:

1. выберите меню [File] -> [New Project].
2. Новый проект будет создан в соответствии с подходом по умолчанию, ранее выбранным пользователем. В зависимости от подхода могут быть подгружены определённые профили и/или инструментарии.

Процедура создания нового проекта #2 — Select Select New Project

1. Выберите меню [File] -> [Select New Project...]. диалоговом окне New Project будет отображен список доступных подходов.

2. Выберите нужный из списка и нажмите кнопку [OK].
3. Новый проект будет создан и инициализирован согласно указанному подходу. В зависимости от подхода могут быть подгружены определённые профили и/или фреймворки.

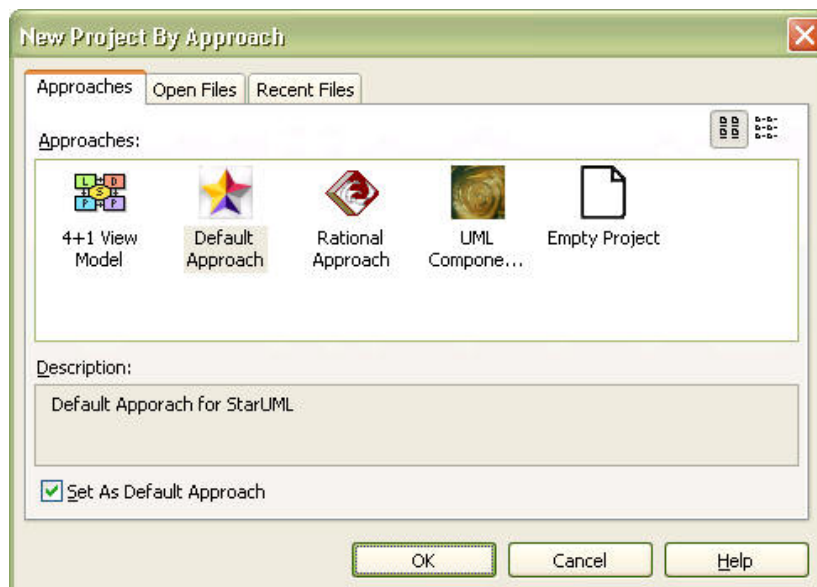


Рис. 3.1: Новый проект

Примечание

- Список доступных подходов зависит от установленной среды разработки пользователя.
- Чтобы изменить заданный по умолчанию подход, откройте диалоговое окно "Select New Project" выберите нужный подход и затем установите опцию "Set As Default Approach"(Использовать как подход по умолчанию).

Открытие проекта

Чтобы начать работать с существующим проектом, его проектный файл нужно открыть. Если проект включает более одной секции, все связанные секции будут также загружены вместе с проектом.

Процедура открытия проекта:

1. Выберите меню [File] -> [Open...].
2. В диалоговом окне "Open Project" выберите файл проекта (.UML) и нажмите кнопку [Open].
3. Выбранный проектный файл будет открыт.

Примечание Проекты могут также быть открыты через диалоговое окно "Select New Project".

Сохранение проекта

Чтобы изменения, сделанные в проекте, не пропали, проектный файл должен быть должным образом сохранен. Ваша работа может быть сохранена в существующий проектный файл или в новый проектный файл. Когда проектный файл сохраняется, то вместе с ним сохраняются данные из связанных с ним секций.

Процедура сохранения проекта:

1. Выберите меню [File] -> [Save].
2. Если имя файла проекта не было определено, появится диалоговое окно "Save Project". Введите имя файла, и нажмите кнопку [Save].
3. Проектный файл будет сохранен.

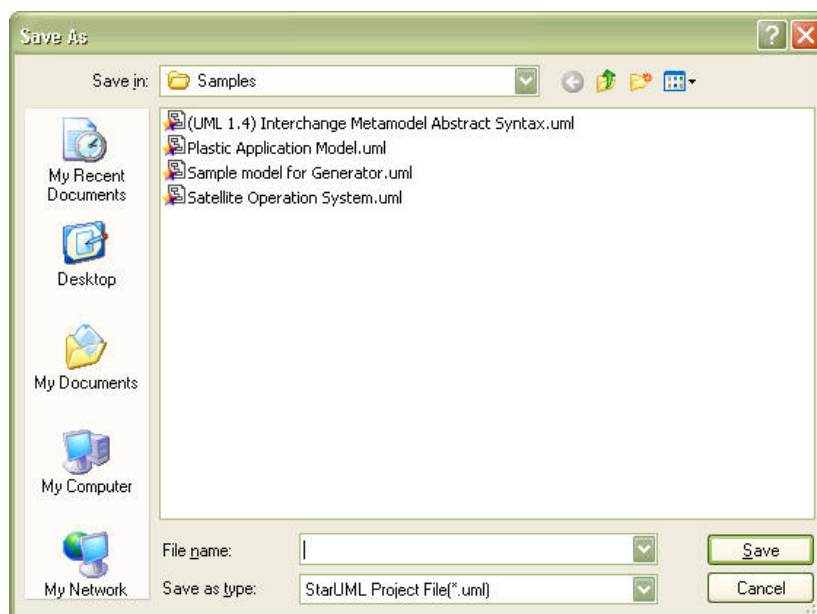


Рис. 3.2: Сохранение проекта

Процедура сохранения проекта в новом файле:

1. Выберите меню [File] -> [Save As...].
2. В диалоговом окне "Save As" введите новое имя файла, и нажмите кнопку [Save].
3. Проект будет сохранен в указанном файле.

Примечание

- Если проект содержит одну или более секций, и есть изменённые секции, диалоговое окно будет запрашивать подтверждение на сохранение каждой измененной секции. Выберите [Yes], чтобы сохранить измененную секцию вместе с проектом.

Заккрытие проекта

Проект может быть закрыт, если больше не требуется его редактирование.

Процедура закрытия проекта:

1. Выберите меню [Файл]->[Close].
2. Если проект не был сохранен после внесения изменений, пользователю будет предложено сохранить изменения. Пользователь может выбрать «да», «нет» или «отмена».
3. После закрытия проект становится недоступным для редактирования.

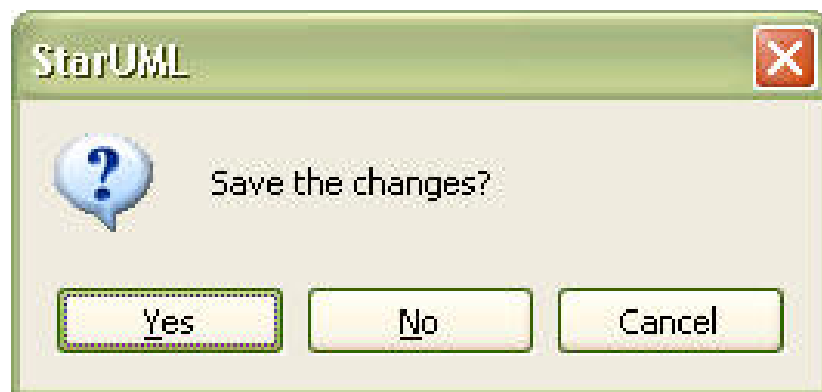


Рис. 3.3: Сохранение изменений

3.0.1 Управление элементами с помощью моделей, подсистем и пакетов

Программная модель состоит из многих элементов и диаграмм. Правильная группировка этих элементов и диаграмм очень важна для эффективного управления проектом. StarUML™ поддерживает три типа группирующих элементов (модели, подсистемы и пакеты), которые пользователь может использовать соответственно их назначению. Способы группировки элементов, реализованные в StarUML™



Модель

Модель выражает физическую систему в определенном аспекте. Например, это может быть аспект анализа, аспект проекта, пользовательский аспект, и т.д.



Подсистема

Подсистема группирует элементы, которые составляют полную физическую систему или её части.



Пакет

Пакет логически группирует и содержит модельные элементы. Это чрезвычайно обобщенный элемент, который может использоваться только для того, чтобы как-то организовать модельные элементы.

Глава 4

Моделирование с помощью StarUML

Эта глава подробно описывает, как создавать и редактировать элементы диаграммы, включая способы организации структуры модели с помощью навигатора модели.

- Редактирование элементов и диаграмм
- Организация структуры модели

Редактирование элементов и диаграмм

Создание новой диаграммы

StarUML TM поддерживает 11 типов диаграмм UML. Пользователь может свободно создавать и манипулировать диаграммами различных типов, как ему необходимо.

Процедура создания новой диаграммы:

1. Выберите в навигаторе модели или на диаграмме элемент, который будет содержать новую диаграмму.
2. Щелкните правой кнопкой мыши и выберите [Add Diagram]. Новая диаграмма будет создана после выбора типа диаграммы.

Доступные типы диаграмм

- **Диаграмма классов (Class diagram)**

Диаграмма классов - визуальное отображение различных статических отношений между класс-подобными элементами. Диаграмма классов может содержать не только классы, но также и интерфейсы, перечислимые типы, пакеты, различные отношения, экземпляры и их связи.

- **Диаграмма прецедентов (Use case diagram)**

Диаграмма прецедентов - отображение отношений между вариантами использования (прецедентами) определенной системы или объекта и внешними акторами. Вариант использования отображает функции системы и то, как эти функции взаимодействуют с внешними акторами.

- **Диаграмма сообщений (Sequence Diagram)**

Диаграмма сообщений отображает взаимодействие инстанций. Она является прямым отображением множества взаимных воздействий (InteractionInstanceSet) между элементами множества инстанций (CollaborationInstanceSet). В то время как Диаграмма сообщений роли ориентирована на классификаторы-роли, обычная Диаграмма сообщений - на инстанции.

- **Диаграмма сообщений роли (Sequence Role Diagram)**

Диаграмма сообщений роли отображает взаимодействия в концепции ролей. Она является прямым отображением Interaction (множества взаимных сообщений между классификаторами-ролями) в пределах Collaboration. В то время как Диаграмма сообщений - отображение инстанций, Диаграмма сообщений роли - отображение классификаторов-ролей.

- **Диаграмма коллаборации (Collaboration Diagram)**

Диаграмма коллаборации отображает взаимодействие между инстанциями. Она является прямым отображением модели взаимодействия инстанций, входящих в CollaborationInstanceSet. В то время как диаграмма коллаборации ролей - отображение классификаторов-ролей, обычная диаграмма коллаборации - отображение инстанций.

- **Диаграмма коллаборации ролей**

Диаграмма коллаборации ролей отображает взаимодействия между ролями. Она является прямым отображением модели взаимодействия классификаторов-ролей внутри коллаборации. В то время как обычная диаграмма коллаборации ориентирована на отображение инстанций, диаграмма коллаборации ролей - отображение классификаторов-ролей.

- **Диаграмма состояний (Statechart Diagram)**

Диаграмма состояний выражает статическое поведение определенного объекта через состояния и переходы состояний. Хотя диаграмма состояний обычно используется, чтобы выразить поведение инстанций классов, она может также использоваться, чтобы выражать поведение и других элементов.

- **Диаграмма действий (Activity Diagram)**

Диаграмма действий - специальная форма диаграммы состояний, которая является подходящей для того, чтобы отображать поток выполнения действий. Диаграмма действий в общем случае используется для отображения любых потоков обработки, но чаще всего применительно к объектам подобным классам,

пакетам и операциям.

- **Диаграмма компонентов (Component Diagram)**

Диаграмма компонентов отображает зависимость между программными компонентами. Элементы, которые составляют программные компоненты и элементы, которые реализуют эти компоненты, могут быть отображены на диаграмме компонентов.

- **Диаграмма развертывания (Deployment Diagram)**

Диаграмма развертывания отображает аппаратные элементы компьютера, другие устройства и программные компоненты, а также процессы и объекты, которые им назначены.

- **Композиционная структурная диаграмма (Composite Structure Diagram)**

Композиционная структурная диаграмма - диаграмма, выражающая внутреннюю структуру классификатора. Она показывает его точки зрения взаимодействия с другими частями системы.

Примечание

- Типы доступных диаграмм изменяются при переходе от одного типа элемента к другому.

Создание элемента на диаграмме

Чтобы создать на диаграмме новый элемент, диаграмму сначала нужно открыть. Палитра элементов содержит различные типы элементов, доступных для создания в зависимости от типа диаграммы. Список доступных элементов изменяется при переходе от диаграммы одного типа к диаграмме другого типа.

Процедура создания элемента из палитры элементов:

1. Выберите тип создаваемого элемента на палитре элементов.
2. Щёлкните желаемое место для нового элемента на диаграмме, чтобы создать там элемент. (Перетаскивайте указатель мыши, чтобы определить область и размер нового элемента. При создании элемента, который соединяет два других элемента, убедитесь, что соединение сделано правильно.)

Процедура одновременного создания нескольких однотипных элементов:

1. Выберите тип создаваемого элемента на палитре элементов.
2. Нажмите [Lock] на палитре или тот же тип элемента еще раз.
3. Создайте несколько элементов подряд.

4. Снова нажмите элемент в палитре, когда создание группы элементов будет закончено.

Примечание

- Создание элемента на диаграмме с помощью палитры элементов фактически означает создание как собственно модельного элемента, так и его визуального образа.