Известия Гомельского государственного университета
имени Ф. Скорины, №5(62), 2010

УДК 681.3

Использование шаблонов объектно-ориентированного проектирования в программировании имитационной модели технологических процессов производства с иерархической структурой

В. Д. Левчук, П. Л. Чечет

Введение

Несмотря на широкое распространение визуальных систем моделирования, в настоящее время остается актуальным разработка программ имитационных моделей (ИМ) с использованием специализированного или универсального языков моделирования. Наибольшими возможностями в построении программы ИМ обладают те языки, в которых присутствует объектно-ориентированный подход к написанию программы. К таким языкам относится универсальный язык программирования C++, на основе которого реализована система моделирования MICIC 4 [1]. Построение объектно-ориентированного кода программы ИМ является нетривиальной задачей, не существует единого универсального решения по проектированию множества классов объектной программы ИМ. В этом случае исследователь, зачастую опираясь в основном на свой опыт и интуицию, пытается разработать удачную структуру классов, которая сможет качественно представлять функционирование сложной системы в программе ИМ. В теории программирования подобная проблема возникла достаточно давно и для её решения были предложены так называемые шаблоны (паттерны) объектно-ориентированного проектирования [2], которые затрагивают определенные типовые ситуации взаимодействия компонентов некоторой системы, предлагая наиболее рациональные, проверенные и эффективные решения по проектированию структуры классов объектно-ориентированной программы.

Использование шаблонов объектно-ориентированного проектирования в имитационном моделировании

Попытки использовать некоторые шаблоны объектно-ориентированного проектирования при разработке имитационных моделей [3, 4] показали эффективность и перспективность использования шаблонов объектно-ориентированного проектирования в моделировании. Использование шаблонов в разработке объектно-ориентированной программы ИМ обладает следующими преимуществами:

– сокращение времени на разработку программы ИМ, так как существующие шаблоны покрывают большое количество типовых ситуаций по взаимодействию компонентов исследуемой сложной системы;

– упрощение отладки и верификации программы ИМ, так как взаимосвязи между классами, устанавливаемые шаблонами, неоднократно проверены и вероятность возникновения ошибок при их задании минимальна;

– высокая реентерабельность получаемой программы ИМ, так как шаблоны объектно-ориентированного проектирования позволяют легко изменять программу ИМ, сводя к минимуму необходимое для этого перепрограммирование.

С учетом данных преимуществ становится ясно, что при разработке программы ИМ с использованием языка, поддерживающего объектно-ориентированный подход, шаблоны объектно-ориентированного проектирования использовать можно и нужно. Опыт разработки имитационных моделей показал, что более эффективным является использование не стандартных шаблонов объектно-ориентированного проектирования, описанных, например, в [2],

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

а шаблонов объектно-ориентированного проектирования, разработанных специально для программирования ИМ.

При разработке объектно-ориентированной программы ИМ технологических процессов производства с иерархической структурой [5] с использованием системы моделирования MICIC 4 были дополнительно разработаны следующие шаблоны объектно-ориентированного проектирования.

Шаблон формализации предназначен для построения основных функциональных классов программы ИМ. Задача этого шаблона – преобразование трехуровневой формальной модели [5] во множество связанных классов в программе ИМ. В результате применения данного шаблона на выходе получается множество классов программы модели, сгруппированных по логическим уровням, как показано на рисунке 1.

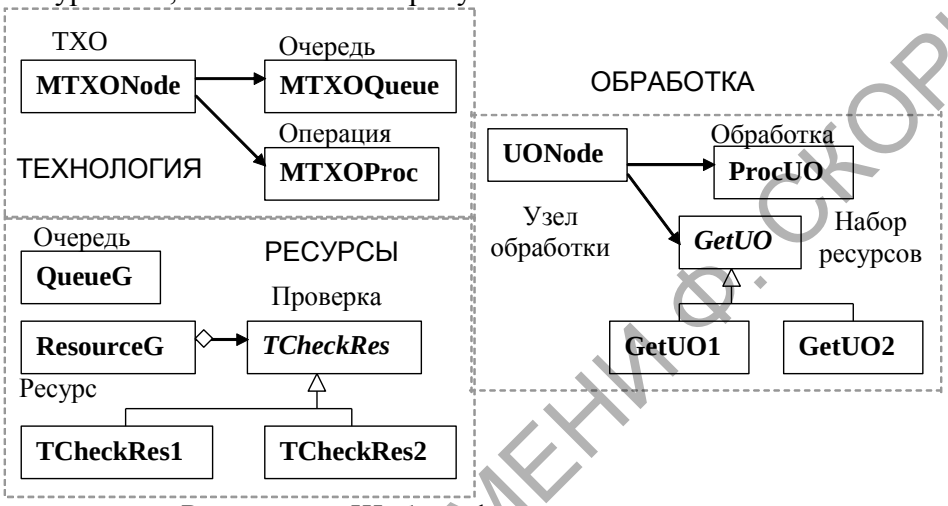

Рисунок 1 – Шаблон формализации

Уровень технологии в программе ИМ представляется тремя классами. Класс операции MTXONode представляет собой узел технологической операции, который внутри содержит очередь к операции и собственно операцию. Собственно операция реализована классом операции MTXOProc. Очередь к операции реализована классом MTXOQueue. Между классом узлом технологической операции и классами операции и очереди устанавливаются отношения осведомленности. Уровень обработки в программе ИМ представлен классом узла обработки UONode. С ним в отношении осведомленности находятся класс набора ресурсов GetUO и класс обработки ProcUO. Класс набора ресурсов является абстрактным. Его наследники (на рисунке 1 – GetUO1, GetUO2) реализуют различные стратегии набора ресурсов. Эта реализация осуществлена с использованием шаблона объектно-ориентированного проектирования «Strategy». Уровень ресурсов представлен классом очереди к ресурсу QueueG и классом ресурса ResourceG. Ресурс агрегирует абстрактный класс TCheckRes, наследники которого используются для создания групп локальных и глобальных ресурсов. Реализация взаимодействия между классами ResourceG и классом TCheckRes реализована с использованием шаблона объектно-ориентированного проектирования «State».

Шаблон иерархий динамических элементов предназначен для упрощения реализации иерархий динамических элементов технологических процессов производства и позволяет реализовывать эффективный код в каждом конкретном случае (размерность, конфигурация ЭВМ). Более того, использование шаблона иерархий динамических элементов позволяет изменять реализацию иерархий даже во время выполнения программы ИМ. Шаблон иерархий динамических элементов разработан на основе шаблона объектно-ориентированного проектирования «Bridge» (рисунок 2). Промежуточный класс MyTransaction (MyDevice), наследованный от базового класса CM MICIC 4 Transaction (Device), агрегирует абстрактный класс реализации иерархий TTnsFiling. Класс TTnsFiling предоставляет интерфейс для работы с вложенными операндами с помощью методов добавления, получения, удаления вложенных операндов. Класс является абстрактным, поэтому никакой реализации хранения операндов он не задает. Конкретные подклассы (TTnsF1 и TTnsF2 на рисунке 2), класса

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

TTnsFiling реализуют конкретный способ хранения вложенных операндов. В частности, это могут быть реализации, оптимизированные для малых и больших объемов вложенных операндов, оптимизированные по скорости выполнения или требовании к определенным ресурсам ЭВМ (памяти, процессору). Так как все подклассы реализации иерархий имеют общего родителя, то появляется возможность подменять их в процессе выполнения программы ИМ. Возможность наследования позволяет легко модифицировать существующие реализации иерархий под каждую конкретную программу ИМ И-процессов.

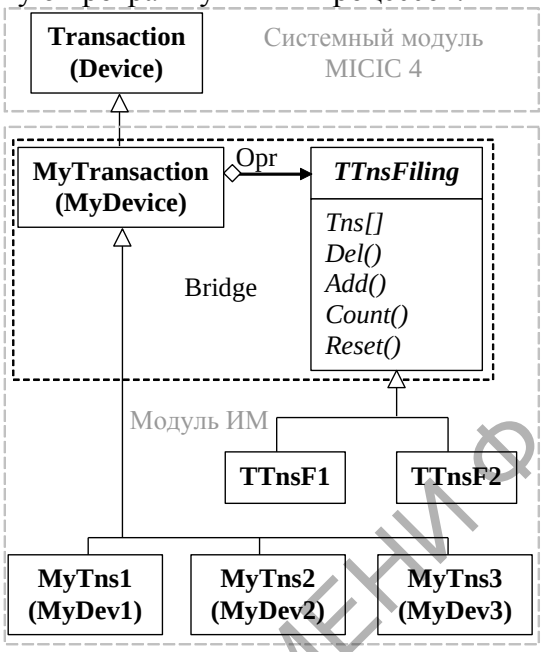

Рисунок 2 – Шаблон иерархий динамических элементов

Класс MyTransaction (MyDevice) используется в программе ИМ в качестве базового при создании транзакта (устройства) с поддержкой иерархий динамических элементов вместо базового класса СМ MICIC 4 Transaction (Device). Работа с этим классом не отличается от работы с базовым классом СМ. Наследники этого класса MyTnsn (MyDevn) используются для представления конкретных транзактов (устройств) в программе ИМ.

Шаблон состояния компонентов предназначен для эффективной реализации поддержки состояния компонентов технологического процесса производства с иерархической структурой. Это может быть изменение режима обработки в технологической операции, изменение маршрута перемещения операнда, изменение стратегии набора ресурсов и др. Этот шаблон основан на шаблоне объектно-ориентированного проектирования «State» (рисунок 3).

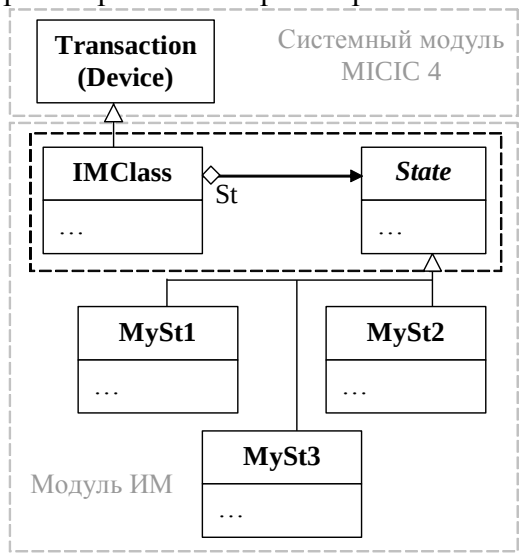

Рисунок 3 – Шаблон состояний компонентов

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

Суть применения этого шаблона заключается в вынесении в отдельные классы состояний объекта. При этом в программу вводится абстрактный класс, который выступает в качестве родительского для классов состояний. Объект не хранит в себе номер состояния или другую аналогичную информацию, а вместо этого агрегирует класс состояния, которому делегирует те операции, которые зависят от его состояния. Такая реализация позволяет независимо разрабатывать классы состояний, даже после того, как класс компонента уже реализован. Каждый класс состояния отвечает только за одно свое состояние, что упрощает его программирование и отладку. Возможность наследования позволяет строить иерархии классов состояний, легко модифицируя уже существующие. Вынос состояний в отдельные классы позволяет использовать одни и те же классы для представления состояний различных независимых компонентов технологического процесса производства с иерархической структурой.

Шаблон сбора статистик предназначен для реализации классов, осуществляющих вычисление откликов в процессе функционирования ИМ. СМ MICIC 4 содержит встроенные средства для накопления результирующих значений и последующего вычисления различных видов откликов по ним. Однако для накопления этих значений в алгоритмы активностей требуется вставлять участки кода, осуществляющие их вычисление и сохранение. Такой подход при всей своей простоте использования приводит к усложнению чтения кода алгоритмов активностей и затрудняет поиск, отладку и изменение кода, осуществляющего сбор статистик. Для построения эффективного кода вычисления откликов ИМ эффективным будет разделение в программе кода, осуществляющего сбор статистик, от кода, моделирующего функционирование активностей модели. Для этого и предназначен шаблон сбора статистик (рисунок 4). Этот шаблон предназначен для проектирования сбора откликов как по уровням иерархических технологических процессов (ТП), так и по отдельным ТП. Идея этого шаблона заключается во введении в программу ИМ глобального объекта, который контролирует любые перемещения транзактов и изменения состояний компонентов в ИМ. Любые движения между элементами технологических процессов производства приводят к вызову метода сбора статистик класса TRespTPP. Виртуальный метод UpdResponce() этого класса анализирует тип транзакта, который переместился между элементами технологического процесса производства. Далее метод UpdResponce() пытается восстановить полную тройку связанных информационного, управляющего и ресурсного транзактов. Если перемещение было на уровне технологии, то восстанавливается только информационный транзакт. Если перемещение было на уровне обработки, то восстанавливаются информационный и управляющий транзакты. Если перемещение было на уровне ресурсов, то восстанавливается полная тройка из информационного, управляющего и ресурсного транзактов. Далее по положению информационного транзакта, характеризующего текущую технологическую операцию обработки операнда, вычисляются номер и уровень иерархии ТП. Следующим шагом выполняется передача информации в объекты классов TRespN и TRespL согласно номеру и уровню ТП соответственно.

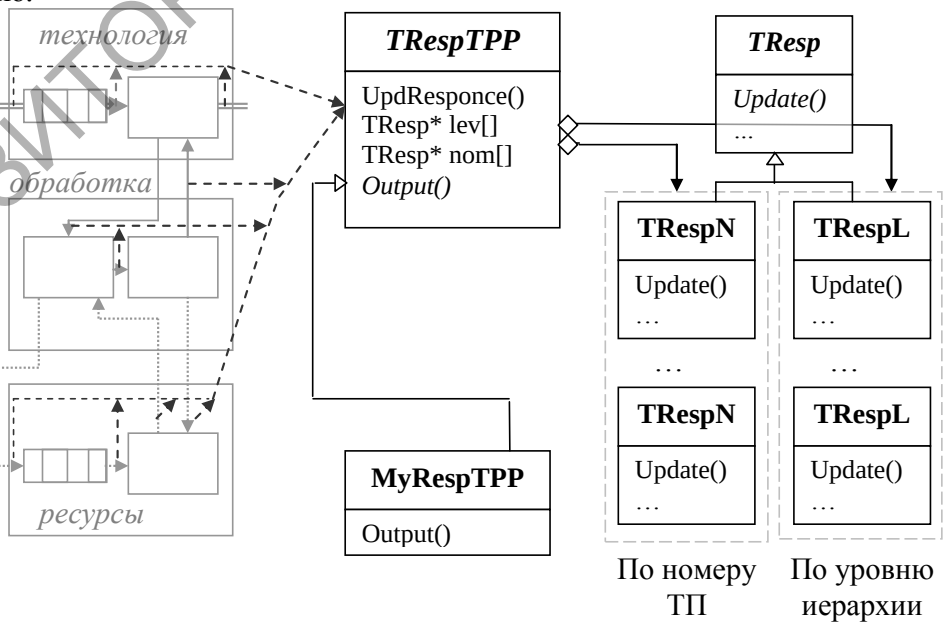

Рисунок 4 – Шаблон сбора статистик

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

За счет наличия одного абстрактного метода Output(), класс TRespTPP является абстрактным и в его наследнике нужно переопределить этот метод. Код, помещенный в этот метод, должен организовать вывод накопленной информации в требуемом для пользователя виде на экран, в файл и др. Агрегируемые объекты классов TRespN и TRespL, наследованные от базового абстрактного класса TResp, предназначены для сбора статистик по номеру и уровню ТП соответственно.

Заключение

На основе разработанных шаблонов объектно-ориентированного проектирования была реализована программа ИМ технологических процессов производства с иерархической структурой. Использование шаблонов объектно-ориентированного проектирования позволило получить код программы, который может быть легко модифицирован исследователем, владеющим только основами объектно-ориентированного программирования. Шаблон формализации позволяет быстро перейти от формальной модели технологических процессов производства к программе ИМ. Шаблон сбора статистик упрощает вычисление откликов в модели. Шаблоны состояния и иерархий динамических элементов упрощают внутреннюю реализацию программы модели, позволяя одновременно выполнить ее эффективно и быстро. Предложенные шаблоны могут быть использованы и при разработке других ИМ.

Резюме. В статье рассмотрен актуальный вопрос повышения эффективности разработки имитационных моделей при использовании объектно-ориентированных языков программирования. Для упрощения этапа перехода от формальной модели к ее программе предлагается использовать шаблоны объектно-ориентированного проектирования. Демонстрируется техника применения таких шаблонов для разработки имитационной модели технологических процессов производства с иерархической структурой.

Abstract. The pressing question of working out imitating model efficiency increase using object-oriented programming languages is considered in the article. For simplification of transition from formal model to its program stage it is offered to use object-oriented designing templates. The technique of such template application for working out of imitating model programming of technological manufacture processes with hierarchical structure is shown.

Литература

1. Левчук, В.Д. Базовая схема формализации системы моделирования MICIC4 // Проблеми програмування. – №1, 2005. – С. 85–96.

2. Приёмы объектно-ориентированного проектирования. Паттерны проектирования: серия «Библиотека программиста» / Э. Гамма [и др.]; пер. А. Слинкин – СПб: Питер, 2001. – 368 с.

3. Чечет, П.Л. Использование паттернов проектирования в разработке имитационных моделей / П.Л. Чечет, В.Д. Левчук // Известия ГГУ им. Ф.Скорины. – Гомель. – 2007. – №6. – С. 131–135.

4. Чечет, П.Л. Актуальные шаблоны программирования имитационных моделей сложных систем / П.Л. Чечет, В.Д. Левчук // 3-я Всероссийская научн.-практ. конференция «Имитационное моделирование. Теория и практика» ИММОД-2007, Санкт-Петербург, 17-19 октября 2007 г. – Режим доступа: http://www.gpss.ru/immod07/doklad/27.html. – дата доступа: 17.12.2007.

5. Чечет, П.Л. Взаимодействие подсистем и элементов имитационной модели производственного процесса с иерархической структурой / П.Л. Чечет // Сборник материалов IV международной межвузовской научно-технической конференции студентов, магистрантов и аспирантов, Гомель, 8-9 апреля 2004г. / УО "Гомельский государственный технический университет им. П.О.Сухого". – Гомель, 2004. – С. 305–308.

Гомельский государственный
университет имени Ф. Скорины

Поступило 20.04.10

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ