РЕАЛИЗАЦИЯ СИСТЕМЫ ПРОВЕРКИ ЗАДАНИЙ ПО ВИЗУАЛЬНОМУ МОДЕЛИРОВАНИЮ В QREAL

Храмышкина Ю. С., студент кафедры системного программирования СПбГУ, juliakhramyshkina@gmail.com,

Аннотация

При создании программного обеспечения требуется много времени на проектирование и анализ системы. Для этих целей хорошо подходит визуальное моделирование, однако овладение этим навыком часто вызывает сложности. В данной работе реализована система проверки заданий по визуальному моделированию для того, чтобы улучшать и совершенствовать этот навык.

Введение

Часто во время разработки программного обеспечения много времени нужно тратить на проектирование и анализ требований, а также на то, чтобы достичь взаимопонимания между разработчиками. Для упрощения этих задач был разработан UML. Однако несмотря на существование UML, часто овладение навыками визуального моделирования вызывает затруднения.

Стоит отметить, что обучение визуальному моделированию вызывает интерес у исследователей. Например, в работе [1] был поставлен масштабный эксперимент, доказавший, что студенты, изучавшие и освоившие визуальное моделирование, справляются с выполнением поставленных заданий успешнее, чем те, кто не изучал или изучал, но не освоил.

В СПбГУ на кафедре системного программирования существует научно-исследовательский проект QReal [2]. QReal представляет собой систему для создания предметно-ориентированных визуальных языков. В частности, ранее в QReal была поддержана диаграмма классов UML [3], на основе которой будет создана система проверки заданий по визуальному моделированию. Пользователь сможет нарисовать решение своей задачи и проверить его на правильность. Проверка будет

производиться путем сопоставления реального решения с одним из идеальных, правильных решений данной задачи.

Поиск шаблонов на диаграммах

Для сопоставления реальных решений и идеальных требуется уметь сопоставлять шаблоны на диаграммах. Для того, чтобы решить эту задачу, были проанализированы существующие решения в данной области.

Одним из подходов является — извлечение всех отношений из графа [4]. В этом подходе используется реляционное представление графа. Однако перед этим из графа извлекаются все виды ребер (отношения), после чего для каждого вида ребер строится отдельный граф, содержащий все существующие вершины и все ребра данного вида. В итоге из одного графа получается несколько с различными видами отношений.

Также существует другой подход, в основе которого лежит использование метрик [5]. В этом подходе ключевым является то, что похожесть двух моделей выясняют на основе различных метрик.Для того, чтобы выяснить, насколько похожи две модели, вводятся типы информации о подобии:

- лексическая информация;
- внутренняя информация;
- информация окрестности.

Степень схожести между элементами сравниваемых моделей имеет значение от 0.0 до 1.0 и может быть представлена в двумерной матрице. Лексическая информация используется для сравнения имен классов. Внутренняя информация используется для измерения подобия свойств, атрибутов и поведения сравниваемых элементов. А третий тип информации используется для измерения структурного подобия между соседями сравниваемых элементов.

Помимо перечисленных существует подход — сопоставление на основе правил [6]. В этом подходе ключевым является создание формальных языков для описания шаблонов, с помощью которых они могут быть выражены, а также алгоритм сопоставления на основе заданных правил.

Для элементов каждого типа: классы, методы, различные виды отношений, — создаются отдельные множества, отвечающие за каждый из

видов. А диаграмма представляется как пара, состоящая из конечного набора классов и набора отношений. Далее это трансформируется в более формальный вид, на основе которого запускается алгоритм сопоставления.

Основная особенность в том, что хоть и алгоритм является недетерминированным, в случае, если для части диаграммы он нашел подходящий шаблон, эта часть диаграммы удаляется, что снижает недетерминированность.

Подход

Система проверки учебных заданий, разрабатываемая в контексте данной работы, нацелена, в первую очередь, на закрепление основных навыков и умений у студентов, полученных на лекциях по визуальному моделированию.

Задачи, которые предоставляются студентам для решений, должны иметь лаконичные решения и не являться громоздкими. Однако допускается существование задач, в которых имеется несколько вариантов решений.

Рассмотрим общую концепцию предлагаемого решения. Студент получает задание, результатом выполнения которого должна быть диаграмма классов. Он создает диаграмму классов с помощью редактора UML в среде QReal, после чего нажимает кнопку “проверить решение”, и в ответ получает сообщение о том, справился он с задачей или нет.

Учитель для задания создает базу решений: рисует диаграмму, удовлетворяющую данной задаче. Затем, если диаграмму можно разбить на логические блоки, учитель выделяет их. Для каждого логического блока из списка существующих шаблонов выбираются те, которым этот логический блок может соответствовать.

В случае, если для данного логического блока не существует всех необходимых шаблонов, тогда недостающие создаются и добавляются к уже существующим.

Представление диаграмм осуществляется следующим образом: в случае, когда рассматривается идеальное решение, для каждого логического блока диаграммы хранится набор узлов, соответствующий классам и интерфейсам, совместно с набором отношений, которые принадлежат этим узлам. В случае, когда рассматривается решение,

которое должно пройти проверку, предлагается хранить его точно таким же образом, но без первоначального деления на логические блоки.

Алгоритм сопоставления

Сопоставление реального решения с идеальными производится следующим образом: из представления идеального решения берется один из логических блоков и одна из его вариаций.

На этой стадии происходит сравнение на уровне классов и интерфейсов, содержащихся в логическом блоке и в предлагаемом решении. Если сопоставление прошло успешно, элементы, совпавшие с частью идеального решения, помечаются как принадлежащие этому логическому блоку, а алгоритм переходит к следующему блоку. Если сопоставление прошло неуспешно, то рассматриваются другие вариации этого блока.

Если совпадений не нашлось, то алгоритм либо заканчивает работу с сообщением о том, что решение не является верным, либо если какая-то часть блоков уже была рассмотрена раньше, алгоритм запускается вновь, однако просмотр логических блоков осуществляется не с того блока, который был рассмотрен, а со следующего после него.

После разметки всех узлов, являющихся интерфейсами и классами, происходит сопоставление связей. Связи могут быть внутренними, находящимися внутри блоков, и тогда они помечаются как принадлежащие блоку. Либо связи могут быть внешними, и тогда происходит сопоставлений реального решения с идеальным на уровне блоков.

В идеальном решении каждая внешняя связь соединяет два определенных блока, в случае, если все внешние связи в решении ученика соединяют те же блоки, что и в решении учителя, алгоритм завершается успешно, а решение считается верным, в противном случае — неверным.

Особенности реализации

На основе предлагаемого подхода и алгоритма сопоставления была разработана система проверки заданий по визуальному моделированию.

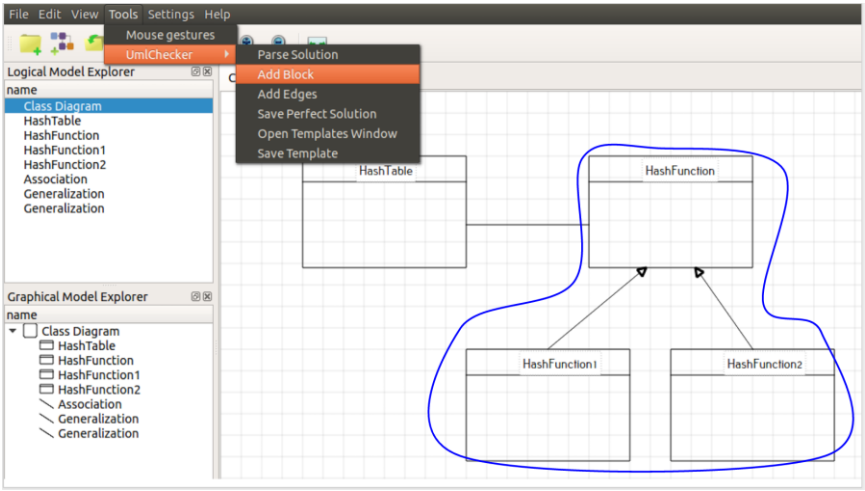

Рис 1. Система проверки заданий

На Рис 1. представлен вариант работы с полученной системой проверки заданий, а именно показана, нарисованная учителем диаграмма классов для задачи “Хеш-таблица со сменными хеш-функциями”. Также на рисунке выделен логический блок, для которого будут выбраны подходящие альтернативы.

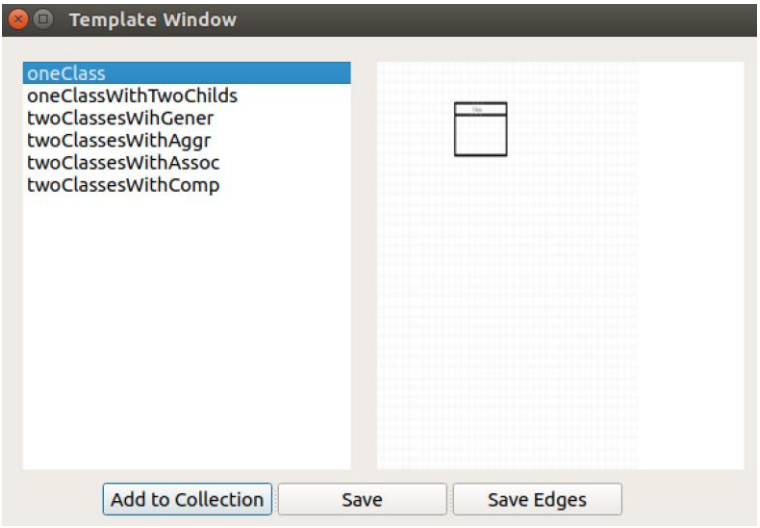

Рис 2. Выбор вариантов для логического блока

На Рис 2. представлено продолжение работы с логическим блоком. Учитель выбирает все подходящие варианты, которым удовлетворяет логический блок. Далее учитель производит разметку всех остальных элементов на диаграмме, после чего сохраняет правильное решение.

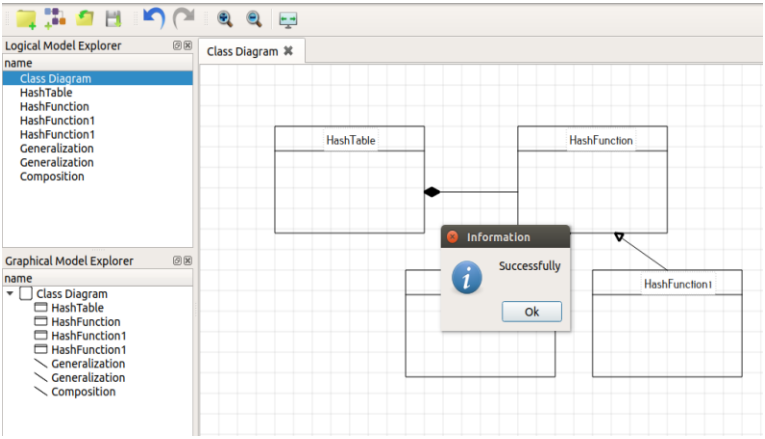

Рис 3. Проверка на правильность решения, предложенного учеником

Впоследствии ученик может нарисовать решение к данной задаче и проверить его на правильность, как показано на рис 3.

Заключение

В рамках данной работы была реализована система проверки заданий по визуальному моделированию в среде QReal для диаграммы классов UML. Это позволяет обучать студентов основам визуального моделирования и автоматически получать результаты об их успехах и неудачах.

Литература

1. Lano K., Yassipour-Tehrani S., Alfraihi H. Experiences of Teaching Model-based Development, 2015.
2. Терехов А.Н., Брыксин Т.А., Литвинов Ю. В., Смирнов К.К., Никандров Г.А., Иванов В.Ю., Такун Е.И. Архитектура среды визуального моделирования QReal. Системное программирование. Т. 4. СПб.: Изд-во СПбГУ. 2000. С. 165–169.
3. Храмышкина Ю.С., Литвинов Ю.В. Поддержка диаграммы классов языка UML 2.5 в среде QReal // Материалы научно-практической конференции студентов, аспирантов и молодых учёных "Современные технологии в теории и практике программирования". СПб.: Изд-во Политехн. ун-та, 2016. С. 86-87.
4. Gupta M., Pande A. Design patterns mining using subgraph isomorphism: Relational view //International Journal of Software Engineering and Its Applications (IJSEIA). – 2011. – Т. 270.
5. Al-Khiaty M. A. R., Ahmed M. UML Class Diagrams: Similarity Aspects and Matching //Lecture Notes on Software Engineering. – 2016. – Т. 4. – №. 1. – С. 41.
6. Ballis D., Baruzzo A., Comini M. A rule-based method to match Software Patterns against UML Models //Electronic Notes in Theoretical Computer Science. – 2008. – Т. 219. – С. 51-66.