

Методы трёхмерной визуализации архитектуры объектно-ориентированного ПО

Е. В. Чепурнова, А. В. Игнатенко

Московский государственный университет им. М.В. Ломоносова, Москва, Россия
echepurnova@graphics.cs.msu.ru, ignatenko@graphics.cs.msu.ru

В современном мире программное обеспечение играет всё большую роль в жизни общества. В связи с этим у разработчиков ПО уходит немало усилий на поддержание и развитие программных продуктов. Визуализация программного кода предназначена для упрощения программистского труда при взаимодействии с кодом. Самый известный пример визуализации – UML-моделирование. UML-диаграммы представляют из себя двумерные графы и являются общепринятым стандартом.

Сейчас интенсивно развиваются методы трёхмерной визуализации ПО. Они обладают рядом преимуществ: трёхмерное пространство ближе человеку по своей сути; такие понятия, как объём, перспектива, ещё одна степень свободы, можно выгодно использовать при визуализации; трёхмерные представления легко переносимы на отображение в виртуальной реальности.

В результате проведённого исследования существующих методов были сделаны следующие выводы:

- для ознакомления с программным кодом удобны средства, визуализирующие статическую информацию (классы, пакеты, т.п.) [1];
- для более глубокого изучения кода, необходима визуализация ещё и динамической информации (например, графы вызовов функций) [2];
- для оценки качества архитектуры ПО могут понадобиться очень разнообразные средства, например для некоторых объектно-ориентированных систем были бы удобны средства, позволяющие выявить известные образцы проектирования;
- для моделирования ПО существуют трёхмерные аналоги UML-диаграмм [3].

Авторами была разработана система GML Doc3D автоматической трёхмерной визуализации объектно-ориентированного кода (C++, C# и Java). GML Doc3D создавалась как средство для изучения незнакомого кода, она визуализирует статическую информацию и обладает несколькими режимами визуализации.

В режиме визуализации всех пространств имён (или пакетов) (рис. 1) была использована метафора трёхмерного города: светло-серыми площадками обозначаются пакеты, высотными домами – классы. Кроме того учитываются некоторые метрики: высотой дома обозначается количество методов данного класса, шириной – количество его атрибутов.

Таким образом при первом взгляде на данное представление программист легко обнаруживает наиболее крупные дома, которым, очень вероятно, соответствуют либо классы, содержащие важные данные, либо классы, которые часто используются.

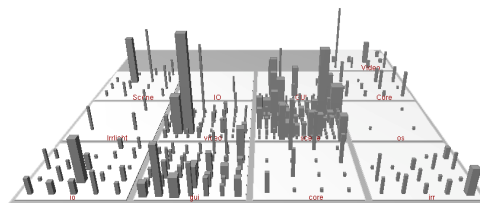


Рис.1. Визуализация пакетов и классов в виде города

Были предложены новые способы визуализации наследования классов и различных связей между классами (рис. 2, 3). В режиме отображения наследования от каждого базового класса рисуется конус, на основании которого располагаются производные классы. В режиме отображения классовых зависимостей интересующий пользователя класс помещается в центр воображаемой сферы, на сфере располагаются все классы, связанные с ним, тип связи обозначается цветом.

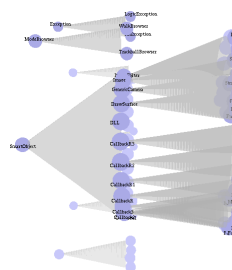


Рис.2. Визуализация наследования классов

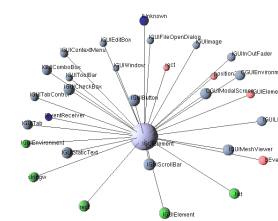


Рис.3. Визуализация различных связей между классами

Система обладает дружелюбным интерфейсом и интуитивно-понятной навигацией. Поддерживается связь трёхмерной документации с текстовой: допускается открытие HTML-документации для любого класса или пакета.

Было проведено субъективное тестирование системы, в котором участвовала группа программистов. Тестирование показало, что все разработанные представления интуитивно-понятны, система удобна для изучения библиотек средних размеров (около 500 классов). Как недостаток стоит отметить отсутствие алгоритма адаптации для крупных систем.

- [1] Michael Balzer, Andreas Noack, Oliver Deussen, Claus Lewerentz. *Software Landscapes: Visualizing the Structure of Large Software Systems*. In Proceedings of VisSym, 2004.
- [2] Johannes Bohnet, Jürgen Döllner. *Visual Exploration of Function Call Graphs for Feature Location in Complex Software Systems*. In Proceedings of the 2006 ACM symposium on Software, 2006.
- [3] Joseph (Yossi) Gil, Stuart Kent. *Three Dimensional Software Modelling*. In Proceedings of the 20th international conference on Software engineering. 1998.