XXX Юбилейная Неделя науки СПбГТУ. Материалы межвузовской научной конференции. Ч. VII: С. 123-124, 2002.

УДК 681.322.068:681.5.01

В.В. Сапунов (5 курс, каф. РВиКС), А.Э. Филиппов, асп.,
Ю.Г. Карпов, д.т.н., проф.

ФОРМАЛЬНЫЕ МЕТОДЫ ПОСТРОЕНИЯ СЛОЖНЫХ ПРОГРАММНЫХ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ СИСТЕМ

В настоящее время разработка большой программной системы сопряжена с рядом трудностей. Одной из главных задач является обеспечения корректной работы программной системы создаваемой системы.

В процессе разработки программной системы выдвигаются общие требования к будущей системе. Например, система управления автомобилем должна обеспечивать необходимый уровень безопасности: опасные конфигурации системы не должны возникать при нормальном функционировании. На основе сформулированных требований программная система проектируется путем декомпозиции, в настоящее время преимущественно объектно-ориентированной. Выдвигаются ряд проектных требований к компонентам системы. Предполагается, что выполнение проектных требований в совокупности должно обеспечить корректную работу системы. Тем не менее, при использовании неформальных средств описания требований к компонентам гарантировать корректную работу системы невозможно, поскольку невозможно строго проверить, действительно ли объединение требований для отдельных компонент приводит к корректной работе системы.

Формальные методы разработки программного обеспечения позволяют описывать требования к компонентам системы на высокоуровневом формальном языке требований. Образно говоря, формальное описание четко говорит, что должна делать компонента, но не описывает, как это достигается. Полученные формальные описания можно проверить на соответствие выдвинутым общим требованиям. Хорошо известными формальными методами являются Z, VDM.

Для проверки формального описания используют специализированные инструментальные средства, например система PVS. Часть данных средств работает на основе формального доказательства теорем, другие, по сути, реализуют метод перебора вариантов и представляют собой интерпретаторы высокоуровневого описания с возможностью обнаружения некорректного или подозрительного поведения.

Применение формальных методов разработки программного обеспечения позволяет проверить проект системы на принципиальную работоспособность до начала этапа реализации, и, следовательно, избежать дорогих ошибок проектирования.

Как видно, в процессе применения формальных методов необходимо подготовить формальное описание компонентов системы. В промышленном производстве программных продуктов в настоящее время распространена практика написания требований и иных специфицирующих документов на неформальном (английская проза) или полуформальном (например, Унифицированный Язык Моделирования, UML) языке. Важным вопросом представляется, не приведет ли внедрение новой технологии к удлинению сроков разработки программной системы, усложнению написания документации и необходимости привлечения дополнительных специалистов, в частности - из области математики и искусственного интеллекта - для создания и проверки формальной спецификации. В [1] отмечается, что многие из этих опасений не оправданы или могут быть устранены. В частности, приводятся примеры успешных разработок ответственных программных систем без выхода за временные и бюджетные рамки. Для упрощения проверки имеется возможность использования так называемых lightweighted (упрощенных) методов, когда проверка формальной спецификации выпол-

няется не путем доказательства теорем, а путем "исполнения" формальной спецификации, по сути, перебором вариантов. Предполагается, что такая технология будет более понятна программистам и, таким образом, облегчит внедрения нового подхода на уровне предприятия. Заметим, однако, что такая проверка не является полностью строгой, поскольку проверить все варианты выполнения в большинстве случаев не удается.

Отдельным интересным вопросом является возможность преобразования полуформальных описаний (например, на языке UML) в формальные спецификации с интеграции двух возможных подходов к разработке программного обеспечения.

Центральным вопросом, во многом решающим применимость формальных методов в настоящее время, является возможность адекватного описания современных, объектно-ориентированных программных систем. В настоящее время разработаны некоторые методы и языки объектно-ориентированного формального описания, например Z++, VDM++. Подробное исследование вопроса применимости формальных методов к описанию современных объектно-ориентированных концепций приведено в докладе.

ЛИТЕРАТУРА:

1.J.P. Bowen, M.G. Hinchey. Seven More Myths of Formal Methods. IEEE Software, Vol.12, n.4, July 1995.