

П. П. Морозкин – студент кафедры информационных систем

А. С. Степанов – научный руководитель

ГЕНЕРАЦИЯ CBCS-СТРУКТУРЫ SDL-МОДЕЛИ СЕТЕВОГО ПРОТОКОЛА

Тестирование SDL-моделей сетевых протоколов в настоящее время является актуальной задачей. Вследствие стремительно растущей сложности SDL-моделей, группой исследователей из ГУАП и исследовательского центра NOKIA был предложен, проанализирован и успешно внедрен метод тестирования SDL-моделей, основанный на совместном моделировании с использованием языков SDL и SystemC. Данный метод имеет недостаток, осложняющий процесс интеграции SystemC-компонентов с SDL-моделями. В статье исследуется подход, позволяющий устранить существующий недостаток за счет автоматизации процесса построения реализации SDL-моделей, пригодной для эффективной интеграции с тестовым окружением. Подход основан на автоматической генерации исходного SystemC-кода SDL-модели. Приводится описание построения SDL-парсера. Вводится понятие CBCS-структуры SDL-модели.

Введение. Модельно-ориентированный подход MDA [1] не получил [2] широкого распространения. Тем не менее, построение моделей, отражающих свойства реальных объектов, с помощью проблемно-ориентированных языков в настоящее время является перспективным подходом к проектированию систем различной природы. В частности применение языка SDL [3] для создания формальных спецификаций сетевых протоколов имеет более чем тридцатилетнюю историю. В рамках языка система представляется моделью, отражающей свойства реального объекта. Технология генерации исполняемой SDL-модели позволяет получить программную модель системы, предоставляющую основу для дальнейших симуляции и тестирования.

Разработка сетевого протокола начинается с определения требований к протоколу. Далее создается спецификация протокола на естественном языке. Следующий этап – формализация созданной спецификации, выполняемая путем построения формальных моделей с использованием языка SDL. По завершению этого этапа формальная SDL-спецификация сетевого протокола, представленная в виде SDL-моделей, тестируется на соответствие требованиям к протоколу. При положительных результатах тестирования дальнейшая разработка протокола обычно основывается на SDL-спецификации. При отрицательных результатах выполняется репроектирование спецификации протокола на естественном языке. Если такой процесс невозможен, выполняется пересмотр требований к протоколу. На рисунке 1 показаны начальные этапы разработки сетевого протокола.

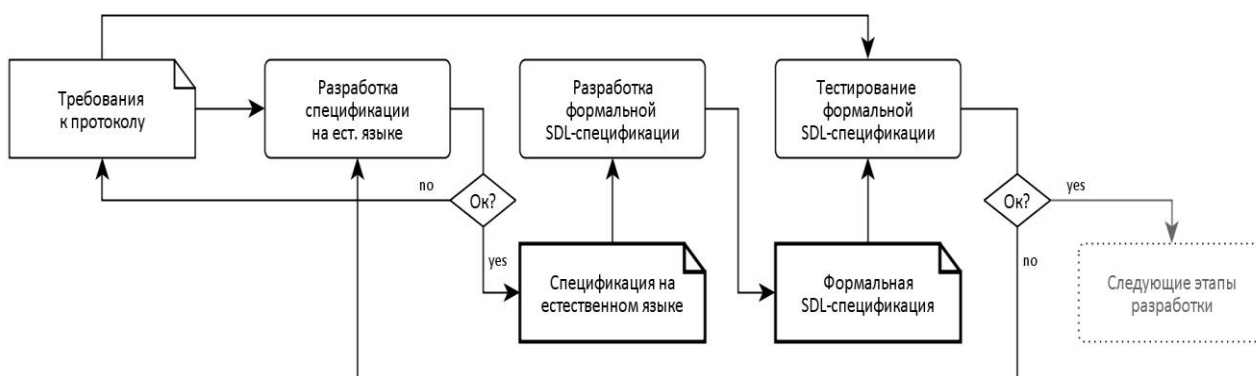


Рис. 1. Начальные этапы разработки сетевого протокола

При создании SDL-моделей зачастую разработчики в связи со специфичностью решаемых задач, а, следовательно, и моделей, сталкиваются с необходимостью выбора наиболее подходящих технологий для проведения тестирования. В силу того, что такие технологии, разработанные, как правило, в рамках крупных проектов отдельных компаний, являются закрытыми, то получение информации о дан-

Рис. 2. Концепция генерации исполняемой SDL-модели

При тестировании SDL-спецификации протокола согласно технологии совместного SystemC/SDL-моделирования SDL-модель содержит два узла, реализующих один протокол. Далее реализация SDL-модели интегрируется с тестовым SystemC-окружением, управляющим процессом тестирования, генерирующим тестовый поток данных и принимающим поток обработанных данных. Следует заметить, что модель и окружение представляются в виде единого программного исполнимого модуля. Такой модуль подвергается тестированию продолжительное время (до нескольких суток), после чего результаты тестирования, представленные в виде отчетов, хранящихся в лог-файлах, анализируются на предмет соответствия характеристик формальной SDL-спецификации требованиям к протоколу. Рисунок 3 иллюстрирует данную идею.

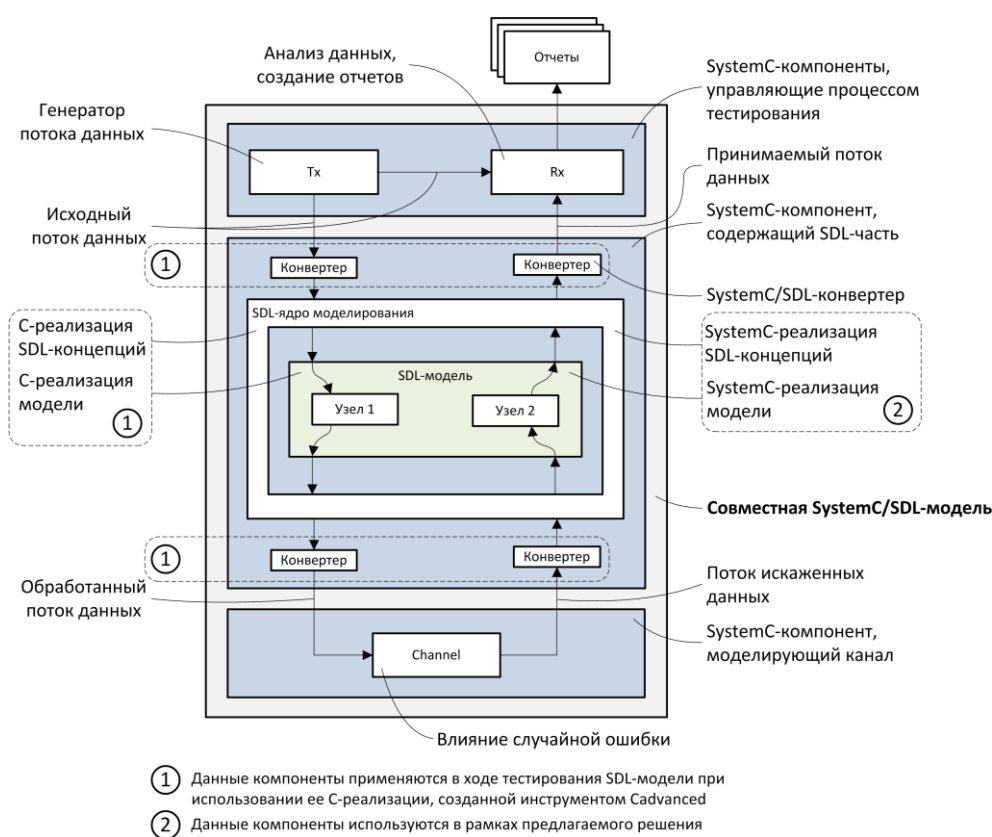


Рис. 3. Концепция тестирования SDL-модели при ее интеграции с SystemC-окружением

При передаче данных SystemC-компонент, моделирующий канал, случайно искажает данные по заданному алгоритму. Принимаемый поток данных анализируется совместно с исходным потоком, после чего создаются отчеты о поведении SDL-модели — ее восприятии, интерпретации и реакции на подверженный случайной ошибке поток данных.

Задача разбора PR-представления SDL-модели сводится к разработке инструмента, позволяющего выполнять анализ корректности модели. Благодаря профессору Теренсу Парру существует генератор парсеров ANTLR [6], позволяющий по заданной грамматике языка генерировать парсеры. Задача еще более сужается, так как ранее исследователь Мишель Шмитт разработал [7] аналогичный парсер, основанный на ANTLR. Особенностью парсера является возможность работы со сгенерированным в процессе анализа модели AST-деревом [8], отражающим синтаксис текстовой нотации модели. Для решения задач получения компонентной, поведенческой и коммуникационной составляющих модели предлагается на основании полученного AST-дерева строить CBCS-структуру (*Component, Behavior and Communication Structure*), отражающую иерархию сущностей модели, поведение процессов и их коммуникацию. На рисунке 4 показан пример SDL-модели и графического представления CBCS-структуры.

Компонент S является системой, В – блоками, Р – процессами и S1 – состояниями процессов. При поступлении сигнала попе, процесс P1, находящийся в блоке В1, используя маршрут P1_B1, отправляет процессу P2, находящемуся в блоке В2, использующему маршрут P2_B2, по каналу В1_B2 сигнал А.

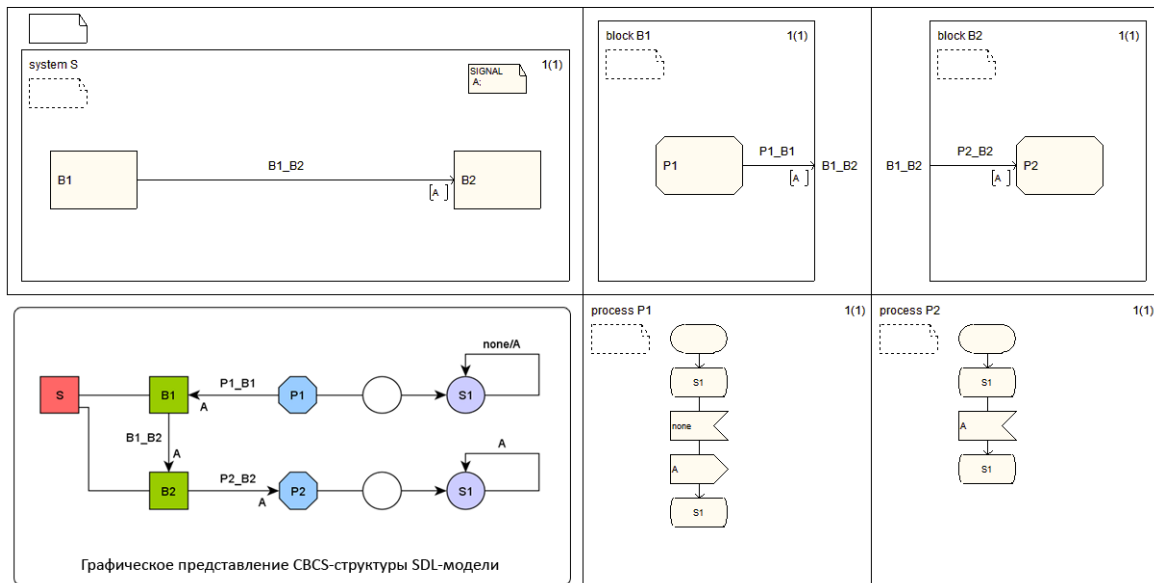


Рис. 4. Графическое представление SDL-модели и CBCS-структуры

По CBCS-структуре, построенной путем анализа элементов AST-дерева, генерируется исходный SystemC-код программной SDL-модели. Следует учесть, что для функционирования полученной модели требуется ядро моделирования, несущее концепции языка SDL. Следовательно, для реализации описываемого решения требуется разработать SDL-ядро моделирования, представленное на языке SystemC. Недостатком описанного решения является, пожалуй, объективная сложность реализации требующихся инструментальных средств.

Заключение. Цель данной статьи — предлагаемое решение задачи тестирования SDL-моделей согласно технологии совместного SystemC/SDL-моделирования. За счет автоматической генерации SystemC-реализации SDL-моделей применение решения позволит упростить построение совместных SystemC/SDL-моделей, сократить сроки тестирования SDL-моделей, а, следовательно, повысить скорость разработки продукта.

Библиографический список

1. Object Management Group, *Model Driven Architecture Guide*, Version 1.0.1, 2003.
2. Robert C. Martin, *The Clean Coder: A Code of Conduct for Professional Programmers*, Prentice Hall, 2011.
3. International Telecommunication Union, *Recommendation Z.100. Specification and Description Language (SDL)*, Geneva, 2002.
4. Black, D., & Donovan, J., *SystemC: From the Ground Up*, New-York: Springer Science+Buisness Media, Inc, 2004.
5. Sergey Balandin, Michel Gillet, Irina Lavrovskaya, Valentin Olenev, Alexey Rabin and Alexander Stepanov, *Co-Modeling of Embedded Networks Using SystemC and SDL*, IJERTCS journal.
6. ANTLR, <http://www.antlr.org>.
7. Michael Schmitt, *The Development of a Parser for SDL-2000*, Institute for Telematics, Germany.
8. Joel Jones, *Abstract Syntax Tree Implementation Idioms*.