

## МЕТОД ТРАНСЛЯЦИИ SDL-СПЕЦИФИКАЦИЙ С ПОМОЩЬЮ МОДИФИЦИРОВАННЫХ СЕТЕЙ ПЕТРИ ВЫСОКОГО УРОВНЯ

Анастасия Заболотная

**Аннотация:** Рассматриваются SDL-спецификации распределенных систем с динамическим порождением и удалением экземпляров процессов. Для них предложен метод трансляции в модифицированные цветные сети Петри - иерархические временные типизированные сети (ИЧТ-сети), в которых используется предложенная Мерлином концепция интервального времени. Естественный подход к верификации основан на использовании формальных моделей, таких, как конечные автоматы, сети Петри и их обобщения. При этом процесс анализа и верификации упрощается. Данная работа описывает SDL-системы с таймерами, которые позволяют адекватно представить значительный класс коммуникационных протоколов. Для них предложен метод трансляции в модифицированные цветные сети Петри - иерархические временные типизированные сети, в которых используются предложенная Мерлином концепция интервального времени. Алгоритм трансляции SDL-спецификаций в сетевые модели системы SDLE реализован методом трансляции в два этапа. Способ моделирования основывается на том, что в многоуровневом описании системы в SDL позиция каждого экземпляра процесса в общей иерархии системы остается неизменной, что позволяет описание системы транслировать в структуру сети, а экземпляры процесса моделировать с помощью фишек. В результате работы алгоритма создается такая сетевая модель, в которой в каждом месте будет содержать не более одной фишки, моделирующей некоторый экземпляр процесса. Таким образом, если во время функционирования системы может существовать  $n$  разных экземпляров любого процесса, то в каждом месте моделирующей его сети может содержаться не более  $n$  фишек, причем каждая фишка будет соответствовать своему экземпляру процесса. Это факт позволяет существенно повысить эффективность моделирования, так как существенно уменьшает перебор вариантов связывания переменных.

**Ключевые слова:** сети Петри, SDL, сетевая модель, коммуникационный протокол, экземпляр процесса.

---

### Введение

Верификация распределенных систем вообще и коммуникационных протоколов, в частности, актуальная проблема современного программирования. Для представления распределенных систем часто используется язык выполняемых спецификаций SDL [1, 2], принятый как стандарт ITU. Преимущество SDL в его выразительности, однако, именно она и затрудняет анализ и верификацию спецификаций таких систем. Один из подходов заключается в автоматическом переводе спецификаций распределенных систем в модели, для которых разработаны методы анализа. В качестве моделей выбраны модифицированные цветные сети Петри, названные иерархическими временными типизированным

сетями (ИЧТ-сетями) [3]. ИЧТ - сети расширяют безопасные цветные сети Петри посредством понятий времени (семантика Мерлина), приоритетов, а также специальных мест, изображающих очереди фишек.

### Общая схема трансляции

Транслятор функционирует следующим образом (рис. 1). Текст SDL - программы подается на вход лексического анализатора. Модуль анализатора обрабатывает текстовый файл, т.е. разбивает текст на лексемы. Синтаксический анализатор выделяет синтаксически - завершенные конструкции и вызывает процедуры внутреннего представления. Далее выполняется соединение каналов, создания списков сигналов, с которыми работают процессы и ряд других действий, которые в итоге и завершают построение внутреннего представления спецификации.

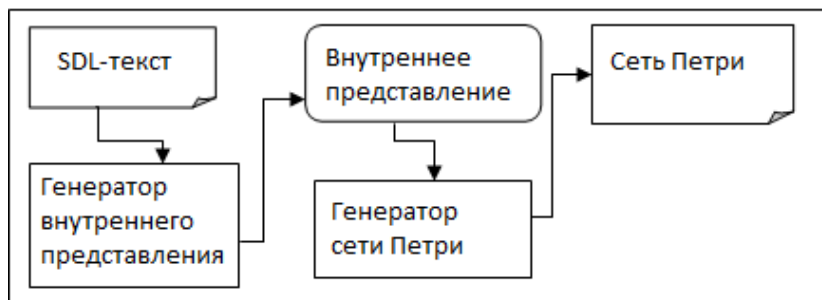


Рис. 1. Общая схема трансляции

Во время выполнения этого этапа также выполняется проверка входящего файла на наличие ошибок, а также на соответствие подмножеству языка SDL, которое может быть представлено с помощью сетей Петри. В случае отсутствия ошибок запускается модуль генерации сетевой модели и строится внутреннее представление ИЧТ-сети (рис. 2).

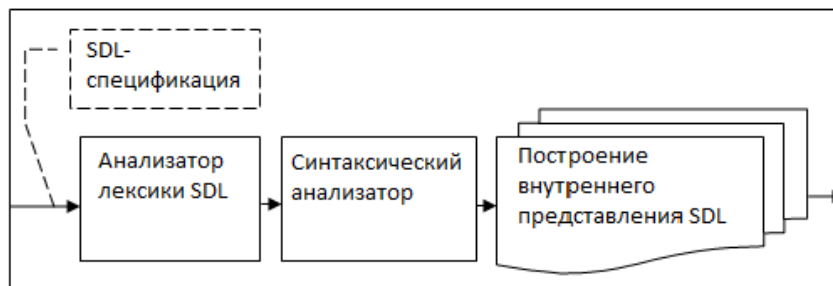


Рис. 2. Генерация внутреннего представления

### Трансляция статических конструкций

SDL (Specification and Description Language) - язык спецификаций и описания, разработанная Международным консультативным комитетом по телефонии и телеграфии (ССИТТ). Язык предназначен для описания структуры и функционирования систем реального времени, особенно, сетей связи. SDL построен на базе модели конечного автомата по объектно-ориентированной схеме (Таблица 1).

Язык SDL имеет как статические, так и динамические конструкции, описывающие порождения и уничтожения экземпляра процесса. Введем ограничения на SDL для того, чтобы модели были максимально простыми и удобными для анализа и верификации (Таблица 2). Поэтому на данном этапе

не будут рассматриваться модели с динамическими конструкциями: трансляция происходит только для систем, в которых не происходит динамического создания и удаления процессов (Таблица 3).

Таблица 1. Графические обозначения

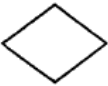
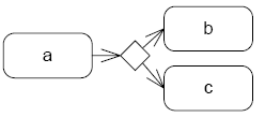
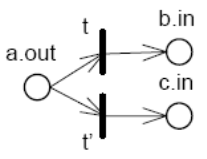
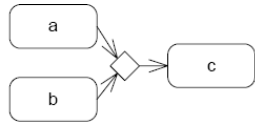
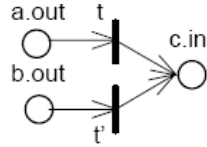
Place:	○
Private transition:	
Public transition:	▮
Arc:	→

Таблица 2. Трансляция статических конструкций

SDL name	Значение символов	SDL graphics	IPN pattern	Connection
Start (initial)	Старт процесса			in: N/A out: a
Stop (final)	Остановка			in: a out: N/A
StateNext, State	Состояние			in: a out: a
Input	Вход			in: a out: b
Output	Выход			in: a out: b
Call	Вызов процедуры			in: a out: c

Таблица 3. Трансляция расширенного подмножества статических конструкций

SDL name	Значение символов	SDL graphics	SDL specifications	IPN pattern	Connection
Control Flow	Поток управления				in: b out: a
Fork	Разветвление				in: b,c out: a
Join	Соединение				in: c out: a,b

Decision	Решение				in: b,c out: a
Merge	Слияние				in: c out: a,b

### Пример трансляции SDL-спецификации в сети Петри

Рассмотрим спецификацию системы S, текстовое описание приведено далее, а графическое показано на рис. 1:

```

system S;
signal s1, s2, s3(Integer), s4, s5, s6;
channel C1
    from B1 to env with s1, s2;
endchannel C1;
channel C2
    from B1 to B2 with s4;
    from B2 to B1 with s5, s6;
endchannel C2;
channel C3
    from env to B2 with s3;
    from B2 to env with s1;
endchannel C3;
block B1 referenced;
block B2 referenced;
endsystem S;

```

В ней описаны три канала - c1, c2, c3 и два блока - B1 и B2. По каналу c1, соединяющему внешнюю среду (environment) с блоком B1, от блока могут передаваться сигналы s1, s2 к внешней среде (environment). По двунаправленному каналу c3, соединяющему блок B2 с окружением, от блока может быть передано сигнал s1, а от внешней среды блок может получить сигнал s3. Между блоками B1 и B2 есть двунаправленный канал c2. По нему от блока B1 в блок B2 может передаваться сигнал s4, который имеет параметр целого сорта, а в обратном направлении - сигналы s5, s6. Блок B1 имеет подструктуру B1, описание которой приведено на рис.3. Каждому подблоку подструктуры B1 в сети соответствует модуль, имя которого совпадает с именем подблока. Блок B2 в системе не описано.

При создании сети соответствующей SDL-системе, используются видимые для всех блоков описания списки сигналов, а также каналов, соединяющих между собой блоки и окружающую среду, и часть информации из описания процессов. Строящаяся сеть содержит по одному модулю на каждый блок SDL-системы. Для большей наглядности в качестве имени модуля можно использовать имя соответствующего блока. На этом же этапе отображаются каналы. Каждый канал в SDL имеет ассоциированную с ним FIFO-

очередь, в которой хранятся сигналы, полученные через этот канал. Этим же свойством обладают и маршруты.

При трансляции SDL-системы, каналы, не поддающиеся разбивке в процессе дальнейшей детализации системы, а также маршруты, естественно представлять местами специального типа - очередями, а сигналы, которые хранятся в очередях, - фишками. Кроме того, каждая фишка должна содержать информацию о том, какой процесс ее отправил и которому процессу она предназначена.

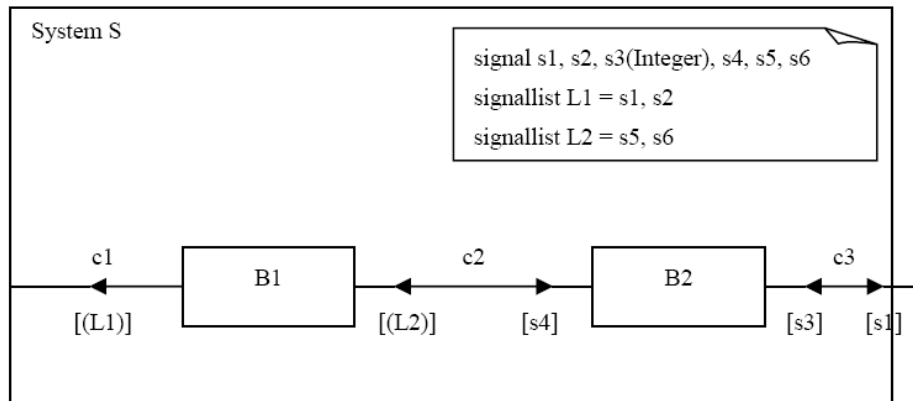


Рис. 3. Описание системы S

При начальной разметке все места, соответствующие каналам, пустые. Однонаправленный канал в сети представляется одним местом. Входной канал отображается местом, что является входным для модуля, представляющий блок, связанный с этим каналом, выходной канал - исходным местом. Двухнаправленный канал представляется двумя местами.

Описание системы - это последовательность диаграмм, где каждая следующая диаграмма осуществляет дальнейшую детализацию системы. Предусмотрены диаграммы подструктур блоков и каналов. Диаграмма подструктуры блока используется в тех случаях, когда рассматриваемый блок представляет сложный объект и состоит из подблоков и внутренних каналов. Каналы внутри системы, которые соединяют блоки между собой и с окружением, назовем внешними. В результате разбиения блоков возникает структура, аналогичная структуре системы, в которой рамка блока играет роль рамки системы.

На каждой странице, связанной с модулем, соответствующим некоторого блока, отображается внутренняя структура этого блока. Это отображение осуществляется таким же способом, что и отображение структуры системы на первой странице. Каждому подблоку на подстранице соответствует один модуль, каждому внутреннему каналу - одно или два места, в зависимости от того, какой канал задействован - одно- или двухнаправленный.

Рассмотрим блок B1, имеющий подструктуру B1, описание которой приведено на рис. 3. Каждому подблоку подструктуры B1 отвечает модуль, имя которого совпадает с именем подблока.

На рис. 4 показано, как отражаются в сети внутренние каналы подструктуры B1. Место c1.1 соответствует внутреннему каналу c1.1 и является исходным для модуля b1.2, место c1.2 соответствует внутреннему каналу c1.2 и является исходным для модуля b1.1, место c2.1 соответствует внутреннему каналу c2.1 и есть входным для модуля b1.2. Каналу c2.2 в сети соответствует два места: c2.21 - входное место и c2.22 - исходное для модуля b1.3, моделирующего подблок b1.3. Новые однонаправленные каналы cx, cy и cz в сети моделируются местами с такими же именами соответственно.

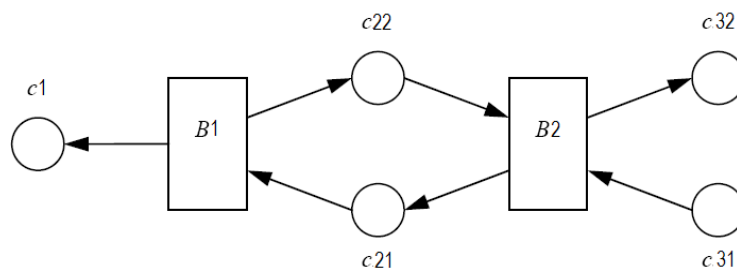


Рис. 4. Мережа для системы S

Каждый сигнал, поступивший по внешнему каналу, может передаваться только по одному внутреннему каналу. По правилам построения иерархической сети копия каждого из мест, представляющий внешний канал, появляется на связанной с модулем странице, которая, в свою очередь, содержит по одному модулю для каждого описания внутреннего подблока или процесса.

В сети, строятся, места, созданные на предыдущем этапе и соответствующие внешнему каналу, сливаются с местами, соответствующими внутренним каналам, подключенным к внешнему таким образом. Назовем места, созданные на предыдущем этапе и соответствующие внешнему каналу, «старыми местами», а места, созданные на этапе отражения подструктуры и соответствующие внутренним каналам - «новыми местами».

Таким образом, каждому «старому месту», что является входным для некоторого модуля, представляющий блок, соответствует свой набор «новых мест», которые являются входными местами для модулей, представляющих подблоки данного блока. Аналогично, каждому «старому месту», что является исходным местом некоторого модуля, соответствует свой набор «новых мест», которые являются выходными местами для модулей, представляющих подблоки данного блока.

В нашем примере для «старого места»  $c1$ , что является исходным местом для модуля  $B1$ , соответствующими «новыми местами» будут  $c1.1$  и  $c1.2$ , которые будут выходными местами для модулей  $b1.2$  и  $b1.1$  соответственно (рис. 5).

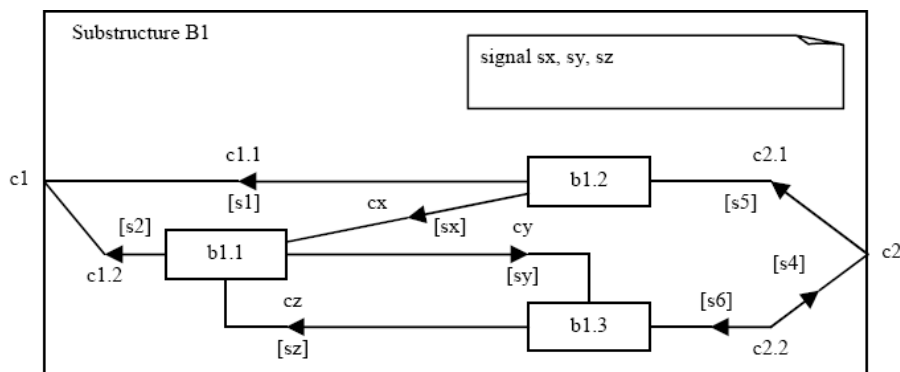


Рис. 5. Описание подструктуры B1

Каждое «старое место», что является входным местом и моделирует блок модуля, сливается с соответствующими ему «новыми местами», которые являются входными местами для модулей, моделирующих подблоки этого блока. Аналогичным образом происходит слияние «старого места» - исходного для модуля, моделирующий блок, с соответствующими ему «новыми местами» - выходными

для модулей, моделирующих подблоки этого блока. Заметим, что несколько мест, которые соответствуют внутренним каналам, могут быть слиты с одним местом, соответствующим внешнему каналу.

На рис.6 показана страница для подструктуры  $B1$ , в которой места, соответствующие внутренним каналам, уже слиты с местами, соответствующими внешним каналам. Место  $c1$  слито с местами  $c1.1$  и  $c1.2$ , место  $c21$  - с местами  $c2.1$  и  $c2.21$ , а место  $c22$  - с местом  $c2.22$ . В дальнейшем «слитым» местам будем приписывать те же имена, имеющие «старые места» на странице, моделирующий охватывает блок. Например, вместо имени  $c22$  &  $c2.22$  будет использоваться имя  $c22$ .

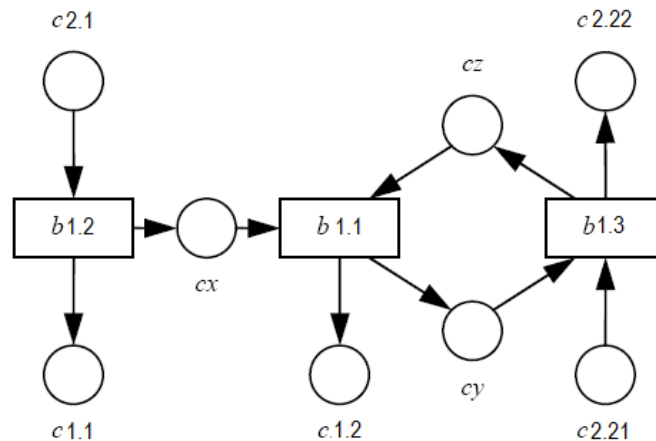


Рис. 6. Сетевое представление подструктуры  $B1$

Если описание любого блока дано за пределами описания системы (указана ссылка на отсутствие описания), то дальнейшего уточнения сети для этого блока не происходит. В результирующей сети (рис. 7) данному блоку будет соответствовать один модуль, который был порожден на предыдущем этапе построения. При моделировании подструктуры канала на странице, связанной с дополнительным модулем, отражается подструктура этого канала аналогично тому, как отражалась подструктура блока.

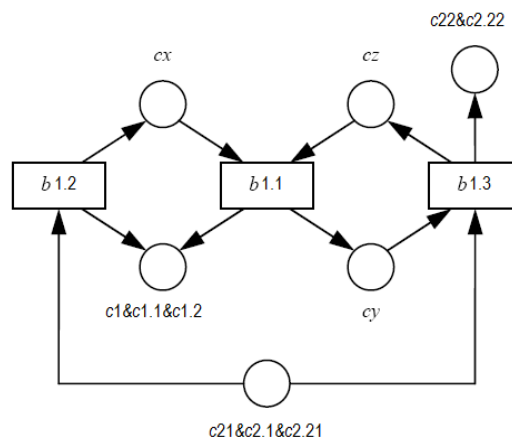


Рис. 7. Конечная сеть для подструктуры  $B1$

Каждый блок, не разделенный на подблоки, должен содержать хотя бы один процесс. Между собой и с блоком процессы соединяются маршрутами. Моделирование блока, состоящего из процессов, аналогично моделированию блока любого уровня иерархии, при этом каждому описанию процесса сопоставляется один модуль. Отличие заключается в моделировании точки присоединения маршрутов к каналу. Сигнал, поступающий в блок по входному каналу, может передаваться по нескольким подсоединенным к нему маршрутам.

В описании семантики SDL точно не определяется, в какие маршруты будет передаваться сигнал. Каждый раз множество маршрутов, которые могут передать этот сигнал, и множество экземпляров процесса, которые могут получить сигнал, выбирается произвольно.

---

### Заключение

Автоматическая генерация сетевых моделей коммуникационных протоколов существенно сокращает трудоемкость проведения экспериментов по их верификацией, а использование принципа иерархии - поуровневого создания сети, делает возможным построение сетевых моделей для систем реальной сложности. Моделирование протоколов с помощью сетей Петри позволяет распознавать семантические ошибки, которые трудно обнаружить стандартными методами тестирования.

С помощью описанного транслятора полученные сетевые модели для SDL-протоколов и проведены соответствующие эксперименты. В ходе экспериментов выявлены семантические ошибки в этих протоколах.

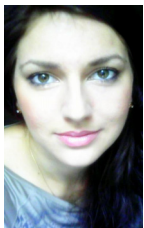
---

### Список литературы

- [ITU-T, 2000] ITU-T. Specification and Description Language (SDL). Recommendation Z.100, 2000.
- [Карабегов А.В., Тер-Микаэлян Т.М., 1993] Карабегов А.В., Тер-Микаэлян Т.М. Введение в язык SDL, Москва, 1993.
- [Jensen K., Christensen S., Wells L., 2007] Jensen K., Christensen S., Wells L. Colored Petri Nets and CPN Tools for Modeling and Validation of Concurrent Systems. Intern. J. on Software Tools for Technology Transfer, 2007.
- [Peterson J. L., 1981] Peterson J. L. Petri Net Theory and the Modeling of Systems. Englewood Cliffs, New Jersey: P, 1981.
- [Jensen K., 1997] Jensen K. Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Verlag, 1997.
- [Murata T., 1989] Murata T. Petri Nets: Properties, Analysis and Applications. Proc. of the IEEE, 1989.
- [Котов В.Е., 1984] Котов В.Е. Сети Петри. – М.: Наука, 1984.
- [Кривий С.Л., 2007] Кривий С.Л. Дискретна математика: Навч. посіб. для студ. вищ. навч. закл. – Київ, 2007.
- [Заболотна А.С., 2011] Заболотна А.С. Метод трансляції SDL – специфікацій за допомогою мереж Петрі високого рівня. Львів, Україна, 26-29 листопада 2011.
- [Churina T. G., 2000] Churina T. G. Colored Petri net approach to modeling of SDL specifications. Joint Bulletin of NCC & IIS. Ser.: Comput. Sci, 2000.

---

### Информация об авторе



**Анастасия Заболотная** - аспирантка II года обучения, Кафедра Информационных систем, Киевский национальный университет имени Тараса Шевченка, Киев, Украина; e-mail: ZabolotnaA@gmail.com

Области научного исследования: сети Петри, дискретная математика, телекоммуникационные системы.