Российская академия наук
Сибирское отделение
Институт систем информатики
им. А.П.Ершова

На правах рукописи

Чурина Татьяна Геннадьевна

МОДЕЛИРОВАНИЕ И ВАЛИДАЦИЯ КОММУНИКАЦИОННЫХ ПРОТОКОЛОВ, ПРЕДСТАВЛЕННЫХ НА ЯЗЫКАХ ESTELLE И SDL, С ПОМОЩЬЮ СЕТЕЙ ПЕТРИ ВЫСОКОГО УРОВНЯ

05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов, систем и сетей

Автореферат
диссертации на соискание ученой степени
кандидата физико-математических наук

Новосибирск, 2000

Работа выполнена в Институте систем информатики
Сибирского отделения Российской академии наук

Научный руководитель:	кандидат физико-математических наук Непомнящий В.А.
Официальные оппоненты:	доктор технических наук Бандман О.Л. кандидат физико-математических наук Скопин И.Н.
Ведущая организация:	Ярославский государственный университет (г. Ярославль)

Защита состоится 15 декабря 2000 года в 14 час. 30 мин. на заседании диссертационного совета К0003.93.01 в Институте систем информатики Сибирского отделения РАН по адресу:

630090, г.Новосибирск, пр. Лаврентьева, 6.

С диссертацией можно ознакомиться в читальном зале библиотеки ВЦ СО РАН (пр. Лаврентьева, 6).

Автореферат разослан “14” ноября 2000 г.

Ученый секретарь
специализированного совета
К0003.93.01
к.ф.-м.н. М.А.Бульонков

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность. В последние годы анализ, валидация и верификация коммуникационных протоколов становятся все более актуальными проблемами. На практике используются два основных подхода к проблеме верификации коммуникационных протоколов. Первый состоит в применении так называемых техник формального описания (FDT), к которым относятся, главным образом, языки выполнимых спецификаций Estelle, являющиеся стандартами ISO, а также SDL, принятый в качестве стандарта ITU. Преимущество языков Estelle и SDL в их выразительной силе, однако именно оно затрудняет анализ и верификацию протоколов связи, способы анализа выполнимых спецификаций остаются предметом исследования. Второй подход базируется на использовании таких моделей, как конечные автоматы, сети Петри и их обобщения, и состоит в моделировании коммуникационных протоколов и верификации полученных моделей. Хотя, по сравнению с FDT, модели более удобны для анализа и верификации, однако при использовании данного подхода моделирование распределенных систем, как правило, выполняется отдельно для каждого примера, что приводит к необходимости верификации процесса моделирования, а это, в свою очередь, является сложной проблемой для реальных распределенных систем.

Автоматический перевод FDT-спецификаций в формальные модели, для которых существуют эффективные методы анализа и автоматические средства верификации, объедининяет преимущества обоих подходов. Известны примеры трансляции FDT-спецификаций в конечно-автоматные модели, сети Петри, алгебры процессов и темпоральные логики действий.

Для Estelle-спецификаций в работе Ж.Л.Ричье, Й.Сифакиса и др. предложен метод автоматического построения конечно-автоматных моделей посредством исчерпывающей симуляции, позволяющий верифицировать некоторые свойства коммуникационных протоколов. В работах В.Димитрова и Р.Лая представлены методы трансляции Estelle-спецификаций в сети Петри, причем используются как ординарные, так и сети Петри высокого уровня (так называемые нумерические). При этом если в работе Димитрова рассматривается ограниченное подмножество Estelle-спецификаций, то в работах Лая — широкое подмножество, включающее динамические конструкции, но без задержек и приоритетов. Однако для адекватного представления протоколов важно рассматривать Estelle-спецификации с задержками и приоритетами. Кроме того, реализация данного метода не описана, а упомянута в качестве те-

мы исследования.

По сравнению с Estelle язык SDL вызывает большой интерес и широко применяется на практике. Опубликован ряд работ по трансляции SDL-спецификаций в различные сетевые модели. Развитие методов трансляции SDL-спецификаций осуществлялось по двум направлениям. При первом используются сети Петри высокого уровня, такие как PrT(predicate-transition)-сети (работы Е.Кеттунена и Н.Хусберга) и M-сети (работы Б.Гралмана). Моделирование с использованием PrT-сетей не является полным моделированием всей спецификации, а оно осуществляется только для верхних уровней спецификации, в которых отражаются поток управления и связи между объектами спецификации.

При втором направлении (работы Й.Фишера и Ф.Баузе) используются новые классы сетей Петри высокого уровня — SDL-сети, ориентированные на язык. Однако их применение требует разработки специальных методов анализа, неизбежно трудоемких в силу сложности сетей. Как в работах Й.Фишера, так и в работах Ф.Баузе описаны сетевые модели, в которых каждому экземпляру процесса соответствует подсеть, и предложены методы построения графа достижимости. В последней работе представлена техника анализа эффективности SDL-сетей, но для применения этих методов требуются дальнейшие исследования, поскольку графы достижимости весьма громоздки и обычные способы их обработки неэффективны.

В России также велись исследования по верификации коммуникационных протоколов. Отметим в частности, работы О.Л.Бандман — по спецификации поведения сетевых протоколов, Н.А.Анисимова — по ручному моделированию с использованием сетей Петри, А.Петренко, Н.В.Евтушенко, Ю.Г.Карпова — по тестированию коммуникационных протоколов с помощью конечно-автоматных моделей, а В.А.Соколова — с помощью сетей Петри.

Среди различных сетей Петри высокого уровня можно выделить раскрашенные сети Петри (РСП), принятые в качестве международного стандарта. Для них разработаны и реализованы практические методы анализа. Кроме того, существует доступная система Design/CPN, активно используемая в практических исследованиях. В книге Йенсена поставлена проблема автоматического построения сетевых моделей SDL-спецификаций, развития средств их верификации, а также проведения экспериментов по обнаружению семантических ошибок распределенных систем с помощью этих средств. Значительный интерес представляет аналогичная проблема и для Estelle-спецификаций.

Цель диссертации состоит в разработке эффективных методов и средств моделирования и валидации коммуникационных протоколов. Достижение цели связывается с решением следующих задач. Первая задача — это разработка алгоритма перевода Estelle-спецификаций без динамических конструкций в эффективную сетевую модель. Вторая — моделирование языка SDL88 посредством раскрашенных сетей Петри. Третья задача — реализация разработанных методов и проведение экспериментов, подтверждающих, что алгоритмы перевода эффективны и могут быть применены для исследования протоколов связи.

Методы исследования базируются на применении аппарата сетей Петри, коммуникационных протоколов и стандартных языков выполнимых спецификаций.

Научная новизна состоит в следующем.

— Разработан алгоритм перевода Estelle-спецификаций без динамических конструкций в эффективную сетевую модель — ИВТ-сеть. ИВТ-сети, т. е. иерархические временные типизированные сети — вариант раскрашенных сетей. В ИВТ-сетях используется предложенная Мерлином концепция времени, близкая к семантике языка Estelle. Впервые при моделировании коммуникационных протоколов были использованы ИВТ-сети и проведено моделирование Estelle-спецификаций с временными задержками и приоритетами, что позволило проводить исследования с представительным классом коммуникационных протоколов.

— Разработан алгоритм перевода спецификаций языка SDL88 без динамических конструкций в раскрашенные сети Петри. Впервые проведено моделирование спецификаций с операторами посылки сигналов, использующих как идентификаторы экземпляров процессов, так и маршрутизацию сигналов.

— Разработаны способ моделирования SDL-спецификаций с динамическими конструкциями и алгоритмы перевода динамических конструкций в раскрашенные сети Петри. При моделировании SDL-спецификаций с динамическими конструкциями ИВТ-сетей недостаточно, поскольку экземпляры процессов мы отображаем фишками, а существование нескольких экземпляров одного процесса предполагает наличие в местах нескольких фишек. Поэтому моделирование SDL-спецификаций осуществляется посредством раскрашенных сетей Петри. Полученные сетевые модели можно исследовать в системе Design/CPN. Впервые проведено моделирование, охватывающее практически полный язык SDL88, посредством раскрашенных сетей Петри, что позволяет решить проблему, поставленную Йенсеном.

Практическая ценность данных исследований заключается в реа-

лизации трансляторов с языков Estelle и SDL в ИВТ-сети и раскрашенные сети соответственно, а также в проведении экспериментов по валидации различных версий коммуникационных протоколов скользящего окна, i-протокола, Inres. Автоматическое построение сетевых моделей существенно сокращает трудоемкость экспериментов, а использование принципа иерархии — поуровнего создания сети — делает возможным построение сетевых моделей для систем реальной сложности. Моделирование протоколов посредством сетей Петри позволяет распознавать семантические ошибки, которые трудно обнаружить стандартными методами тестирования.

Апробация работы проведена на следующих международных научных конференциях.

3rd International Conference on Parallel Computing Technologies, St.Petersburg, Russia, 1995. (Lect. Notes Comput. Sci., Vol. 964).

IFIP 15th International Symposium on Protocol Specification, Testing and Verification, Warsaw, Poland, 1995.

Third International Workshop on Concurrency, Specification and Programming, Warsaw, Poland, October 1997.

International Conference on Parallel Computing in Electrical Engineering, Bialystok, Poland, 1998.

1st International Workshop on the Formal Description Technique Estelle, Evry, France, 1998.

Четвертый Сибирский Конгресс по Прикладной и Индустриальной Математике (ИНПРИМ-2000), Новосибирск, Россия, 2000.

Кроме того, полученные результаты обсуждались на семинарах лаборатории теоретического программирования ИСИ СО РАН и кафедры систем информатики НГУ. Работа поддерживалась следующими грантами: РФФИ 93-01-986, 1993—1995; Международного Научного Фонда и Российского правительства, JCP 100, 1994; ИНТАС 1010-CT93-0042, 1993—1994; ИНТАС-РФФИ N 95-0378, 1997—1999; Президиума СОРАН Поддержки международных проектов, 1997.

Публикации. По теме диссертации опубликовано 14 научных работ.

Структура работы. Диссертация состоит из введения, трех глав, заключения, списка литературы из 67 наименований и приложения. Основное содержание составляет 125 страниц, объем приложения 15 страниц. Работа включает 43 иллюстрации и 2 таблицы.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается актуальность направления исследования диссертационной работы, формулируются цели, характеризуется

научная новизна и практическая ценность работы, приводится список конференций и семинаров, на которых проведена апробация данной работы.

В **первой главе** представлено моделирование Estelle-спецификаций без динамических конструкций. В **разделе 1.1** описываются базовые понятия языка Estelle, основанного на модели расширенного конечного автомата. Спецификация на языке Estelle описывает иерархически структурированную систему недетерминированных компонентов, взаимодействующих с помощью сообщений по двунаправленным каналам между портами (*точками взаимодействия*). Каждый компонент есть экземпляр модуля.

Раздел 1.2 посвящен описанию ИВТ-сетей. В ИВТ-сетях допускается тип массив и запись. Кроме того, сети могут содержать места-очереди, способные хранить неограниченное число фишек. ИВТ-сети являются квазибезопасными, т. е. в них все места, за исключением мест-очередей, могут содержать не более одной фишки. Иерархическая сеть — это композиция множества неиерархических сетей, называемых *страницами*. Страницы могут содержать вершины специального типа — *модули*, которые соединяются с местами на странице по тому же принципу, что и переходы. Модуль представляет подсеть, располагающуюся на отдельной странице, которая в свою очередь может содержать модули. Такая страница называется *подстраницей*, на которой располагается модуль. Подстраница содержит копии всех мест, с которыми связан модуль.

Некоторые термины, такие как модуль и переход, используются и в языке Estelle, и в сетях, поэтому будем использовать приставки E- и N- для обозначения объектов Estelle и сети соответственно.

В **разделе 1.3** к рассмотрению предложен алгоритм для одноуровневых спецификаций, т. е. состоящих из системных модулей, не имеющих наследников. Сеть, моделирующая одноуровневую спецификацию, является иерархической и строится с помощью поэтапного уточнения. Сначала создается страница, которая соответствует основной структуре системы и содержит по одному N-модулю для каждого системного E-модуля. При генерации сетевой модели описания типов переходят, возможно с синтаксическими изменениями, в декларации сети. Каналу, соединяющему две точки взаимодействия, в общем случае сопоставляется два места-очереди, каждое из которых моделирует поток сообщений между E-модулями в одном напрвлениии. Соединение N-модулей и мест происходит таким образом, чтобы сообщения, отправленные одним E-модулем, попадали в очередь, ассоциированную с точкой взаимодей-

ствия другого. Формальным параметрам и экспортируемым переменным в сети также сопоставляются места.

При построении сети, моделирующей тело модуля, каждому E-переходу сопоставляется один N-модуль, а переменной — одно место. На странице также создается место *State*, тип которого определяется множеством локальных состояний E-модуля. Место *State* является входным и выходным для каждого N-модуля и обеспечивает выполнение только одного E-перехода. Если E-переход использует какой-нибудь ресурс (переменную, параметр или точку взаимодействия), то место, соответствующее этому ресурсу, соединяется с N-модулем, представляющим E-переход. Действие E-перехода представляется сетью, которая располагается на подстранице, связанной с N-модулем, сопоставленным E-переходу. Блок перехода разбивается на последовательность подблоков. Каждому подблоку сопоставляется фрагмент сети. Фрагменты соединяются друг с другом с помощью служебных мест в том же порядке, что и подблоки. Подблок, являющийся последовательностью операторов присваивания, и, возможно, операторов посылки сообщений, представляется в сети одним N-переходом, причем, ни один из операторов присваивания не использует переменной, ранее измененной в этом же подблоке. При этом операторы, образующие подблок, преобразуются в выражения на дугах. Оператор вызова процедуры или функции в сети E-перехода представляется N-модулем. Сеть, моделирующая соответствующую процедуру или функцию, размещается на странице, связанной с этим N-модулем. При моделировании условных операторов и циклов используются библиотечные фрагменты. Цепочку фрагментов, представляющих блок E-перехода, ограничивают два служебных перехода — *start* и *end*. Место *State* является входным для перехода *start* и выходным для перехода *end*, обеспечивая тем самым атомарность выполнения E-перехода. Фрагмент сети, соответствующий задержке перехода, помещается между фрагментом, моделирующим условие возможности этого перехода, и переходом *start*, так как отсчет задержки происходит, когда условие возможности E-перехода проверено и выполняется. Время задержки E-перехода отсчитывается специальным переходом с интервалом срабатывания, указанным в задержке; все прочие N-переходы имеют интервал срабатывания $[0, 0]$.

В **разделе 1.4** рассматривается алгоритм генерации сетевых моделей для иерархических спецификаций. Для спецификации строится сеть, иерархия страниц которой повторяет общую иерархию системы. Первая страница представляет уровень системных модулей. Если какой-нибудь E-модуль имеет наследников, то подстраница, связанная

с соответствующим ему N-модулем, содержит по одному N-модулю для каждого экземпляра модуля-наследника. Затем для каждого из полученных N-модулей строится дерево страниц, повторяющее иерархию подсистемы с корнем в соответствующем системном E-модуле. Трансляция модулей осуществляется по одной схеме, которая не зависит от положения модуля в иерархии Estelle-спецификации. Для каждой подсистемы создается дополнительная конструкция, которая связывает все модули в подсистеме и реализует правило выбора E-перехода для выполнения на следующем такте. Родительский модуль управляет вычислениями наследников с помощью контрольных мест. Все модули-наследники класса *activity* делят одно контрольное место, что делает возможным выполнение перехода только одного из E-модулей. Каждый модуль-наследник класса *process* имеет собственное контрольное место. Фишки появляются во всех местах одновременно, что соответствует параллельному выполнению.

Раздел 1.5 этой главы содержит описание результирующей сетевой модели и оценки ее размера. Алгоритм генерирует квазибезопасные сети. В такой модели нет необходимости в переборе вариантов связывания переменных при срабатывании переходов, что повышает эффективность моделирования. Если спецификация не содержит процедур и функций, то оценка размера полученной сети становится линейной.

Во **второй главе** представлено моделирование спецификаций языка SDL88 посредством раскрашенных сетей Петри. В **разделе 2.1** дается описание языка SDL88, который построен на базе модели расширенного конечного автомата и описывает систему, состоящую из блоков, соединенных между собой и с окружающей средой каналами. Функционирование системы описывается в терминах процессов, взаимодействующих с помощью сигналов, передаваемых по каналам, а внутри блоков — по маршрутам, соединяющим между собой процессы.

Раздел 2.2 посвящен описанию раскрашенных сетей, введенных Йенсеном. В отличие от ИВТ-сети в сетевой модели Йенсена имеется понятие *глобальных часов*, посредством которых представляется текущее время. Некоторые множества цветов получают дополнительный признак. Фишка, принимающая значения из такого множества, дополнительно несет значение, называемое *временным штампом*. Оно определяет момент времени, раньше которого фишка не может использоваться при срабатывании какого-либо перехода. Временной штамп новой фишки вычисляется как текущее время в модели плюс задержка, величина которой определяется *временной пометкой*. Временные пометки связаны с переходами и дугами.

Раздел 2.3 содержит описание моделирования спецификаций, не содержащих динамических конструкций. Построение сети, моделирующей такую спецификацию, аналогично построению ИВТ-сети для иерархической Estelle-спецификации. Отличия и трудности возникают при моделировании маршрутизации сигналов, операторов, связанных с установкой таймеров, и конструкций, связанных со специфической обработкой очереди входных сигналов. Кроме того, очереди сигналов в РСП представляются списками, поэтому, в отличие от ИВТ-сети, очереди соответствует одна фишка.

Выходные сигналы в SDL всегда несут персональный идентификатор экземпляра процесса (ПИД) отправителя и либо ПИД экземпляра-получателя, либо специальное значение, указывающее, что сообщение предназначается любому экземпляру, либо строку, состоящую из идентификаторов каналов и маршрутов, использующихся при передаче сигналов. Поэтому множество цветов, приписываемое месту-каналу, строится таким образом, что позволяет формировать сообщения трех форматов. При моделировании состоящего из подблоков блока в строящейся сети копия места, соответствующего каналу, присоединенному к блоку ("старому" каналу), сливается с местами, соответствующими каналам внутри блока, присоединенным к "старому". При этом копия места, моделирующего входной поток сигналов к блоку, сливается с местами, моделирующими входные потоки сигналов к подблокам этого блока. Аналогично сливаются места, моделирующие выходные потоки сигналов. При моделировании блока, состоящего из процессов, возникают отличия в отображении точки присоединения маршрутов к каналу. Сигнал, поступающий к блоку по входному каналу, может передаваться по нескольким подсоединенным к нему маршрутам. Для оптимизации сети предложенный алгоритм поддерживает три способа распределения сигналов, а именно: передачу сигнала из входного канала только в один присоединенный маршрут; в каждый присоединенный маршрут, по которому этот сигнал может передаваться; всем экземплярам процессов, которые связаны с этим каналом.

При построении сети для экземпляра процесса множество состояний процесса, все сорта, определения сигналов и списков сигналов, описанные в декларативной части процесса, преобразуются в множества цветов. Порт экземпляра процесса представляется местом *queue*, которое соединяется с копиями мест, соответствующих входным маршрутам процесса, посредством служебных переходов *link*. Множество состояний процесса определяет множество цветов служебного места *State*, роль этого места точно такая же, что и в ИВТ-сетях. Кроме того, в сети, пред-

ставляющей экземпляр процесса, создается служебный переход *delete*, моделирующий удаление первого сигнала из порта экземпляра процесса в том случае, если экземпляр, находясь в определенном состоянии, не воспринимает этот сигнал.

Если SDL-спецификация содержит процессы, использующие таймерные конструкции, то на первой странице сети создается место *now*, которое содержит одну фишку целого значения, несущую временной штамп, а также переход *add*, имеющий временную пометку @ + 1. Место *now* — входное и выходное для этого перехода. В процессе функционирования моделирующей сети временной штамп фишки в месте *now* определяет текущий момент в системе. Сеть, представляющая SDL-переход с оператором установки таймера, содержит копию места *now* и место, моделирующее таймер. Текущее время в модели не изменяется до тех пор, пока остаются переходы, имеющие возможность сработать. Изменение значения глобальных часов происходит тогда, когда в сети остаются только переходы, которым не позволяют сработать временные штампы фишек. Для того чтобы переход *add* имел равные возможности по отношению к другим переходам, на каждой входной к месту *State* дуге устанавливается временная пометка из некоторого интервала, который определяется исходя из предположений о длительности выполнения перехода в системе. В данном разделе подробно описаны фрагменты сети, моделирующие такие конструкции, как установка и сброс таймера, средство сохранения сигнала *SAVE* и сеть для перехода с разрешающим условием, функционирование которых нарушает дисциплину FIFO-очереди и вызывает трудности в сетевом моделировании.

Раздел 2.4 является логическим продолжением предыдущего и предлагает способ перевода в раскрашенные сети спецификаций с динамическими конструкциями. Здесь описаны основные моменты, которые возникают при моделировании создания и уничтожения экземпляров процессов во время функционирования системы. Моделирование основано на том, что многоуровневое описание системы в SDL имеет статический шаблон. Число экземпляров процесса может измениться в процессе функционирования системы, но позиция каждого экземпляра в общей иерархии системы остается неизменной. Статический шаблон моделируется структурой сети, а экземпляры процессов — фишками. Структура результирующей сети в целом аналогична той, что описана в разделе 2.3. Основное отличие состоит в том, что в сети N-модуль сопоставляется не экземпляру процесса, а его описанию, а фишки, принадлежащие конкретному экземпляру процесса, помещаются в места сети при создании экземпляра процесса и удаляются, когда он входит

в состояние $STOP$. Поскольку фишки, принадлежащие экземплярам одного и того же процесса, располагаются в одних и тех же местах, возникает необходимость в их дополнительной идентификации. Для этого фишки снабжаются уникальным признаком — ПИД. Для генерации ПИД используется специальное место PId, которое располагается на самой верхней странице сети и содержит одну фишку из множества цветов $integer$, значение которой определяет личный идентификатор экземпляра процесса при его создании во время функционирования сети. Копия места PId присутствует на каждой странице, где происходит моделирование порождения нового экземпляра процесса, и является входным и выходным местом для N-переходов, осуществляющих это моделирование.

Особенности моделирования порождения одного экземпляра процесса другим отражаются в сети в том, что для N-модуля, соответствующего порождаемому процессу, создается дополнительное служебное место cr_id (где id имя порождаемого процесса). Оно является для этого N-модуля входным и выходным для N-модуля, представляющего родителя. При срабатывании N-перехода, моделирующего запрос на создание экземпляра процесса, в служебное место cr_id добавится элемент, значение первого поля которого есть ПИД создаваемого экземпляра, значение второго поля — ПИД "экземпляра-родителя". Кроме того, на странице, соответвующей описанию порождаемого процесса, строится служебный переход $create$. Все служебные места (кроме соединительных), места-параметры, места-переменные и места-таймеры являются для него выходными, а входным — служебное место cr_id. При срабатывании перехода $create$ в каждое выходное место добавится по одной фишке, помеченной ПИД создаваемого экземпляра. На странице, соответствующей описанию процесса, на каждой входной дуге любого N-перехода в первой позиции кортежа указывается одна и та же переменная, исключение составляют дуги, соединяющие места PId и now с этим переходом. Так как в РСП все вхождения переменной в спусковую функцию перехода и выражения на связанных с ним дугах замещаются одним и тем же значением, в связывание автоматически входят фишки, принадлежащие одному и тому же экземпляру процесса.

Раздел 2.5 содержит описание результирующей сетевой модели и оценки ее размера. Если в спецификации не содержится процедур, то оценка становится линейной. При отображении спецификаций с динамическими конструкциями создается сетевая модель, в которой в каждом месте содержится не более одной фишки, моделирующей конкретный экземпляр процесса. Это факт позволяет повысить эффективность

моделирования, так как существенно уменьшает перебор вариантов связывания переменных при срабатывании переходов. Следует отметить, что формальной семантики языка SDL не существует. Обычно системы, моделирующие SDL, определяют семантику используемых конструкций и вносят определенные ограничения. Поэтому, можно считать, что предложенный алгоритм задает сетевую семантику языка SDL88.

В **третьей главе** в **разделе 3.1** описана разработанная в рамках нацеленного на валидацию коммуникационных протоколов проекта система ESPV, представляющая собой интегрированный программный комплекс для проектирования, анализа и симуляции сетевых моделей распределенных систем. Основными компонентами комплекса являются трансляторы с языков Estelle и SDL88, конвертор из внутреннего представления ИВТ-сети во входной формат системы Design/CPN, графический редактор и симулятор. Трансляторы с языков Estelle и SDL осуществляют автоматический перевод спецификации соответствующего языка во внутреннее представление ИВТ-сети. При трансляции SDL-спецификаций внутреннее представление может иметь специальные пометки на дугах, переходах и местах. Конвертор осуществляет автоматический перевод ИВТ-сети со специальными пометками в текстовое представление, являющееся входным для системы Design/CPN. Таким образом, система ESPV позволяет использовать средства анализа системы Design/CPN. В многооконном графическом редакторе ИВТ-сеть представляется деревом страниц, средствами редактора осуществляется построение и изменение сетевой модели и контроль структурной корректности иерархических сетевых моделей. Симулятор интегрирован с редактором и визуализирует функционирование ИВТ-сетей, а также позволяет протоколировать сеанс симуляции.

В **разделе 3.2** описана реализация алгоритмов перевода в сетевые модели. На первом проходе трансляции строится внутреннее представление программы в виде атрибутированного дерева разбора, а на втором — как для Estelle-, так и для SDL-спецификаций — генерируется внутреннее представление ИВТ-сети в системе ESPV. Процесс построения сетей производится по уровням, соответствующим этапам в описании алгоритмов трансляции, представленных в гл. 1 и 2. Все уровни сети, кроме нижних, используют модули системы ESPV. На завершающем шаге — оптимизации сети — удаляются все пустые переходы, т. е. имеющие пустые тела и не имеющие выражений на выходных дугах, и сливаются в один два последовательно выполняющихся перехода, обладающие определенным набором свойств. После создания внутреннего представления сети осуществляется размещение ее элементов на

плоскости, т. е. каждому элементу приписываются координаты на соответствующих страницах системы ESPV. При этом фрагменты сети, реализующие стандартные конструкции, размещаются типовым образом. При трансляции SDL-спецификаций сеть переводится в текстовое представление системы Design/CPN, которая не предоставляет средств для автоматического построения сети, но имеет возможность загружать сетевые диаграммы в текстовом представлении ("interchange"формате), базирующемся на языке SGML (Standard Generalized Markup Language).

В **разделе 3.3** рассматривается методика валидации сетевых моделей в системе ESPV. На первых этапах симуляция проводится с использованием упрощающих условий, далее изучаются исключительные ситуации при ненадежной среде, при этом используются условия прерывания симуляции. Кроме того, воздействие внешнего окружения моделируется входной последовательностью событий, а контроль потока сообщений осуществляется посредством встроенных в сетевую модель мест-очередей.

В **разделе 3.4** описаны эксперименты с протоколами скользящего окна, такими как протокол Стеннинга и i-протокол, представленными на языке Estelle, а также с протоколом Inres, представленным на SDL. Протоколы скользящего окна состоят из двух компонентов: передатчика и приемника, взаимодействующих между собой через ненадежную среду, в которой возможны потеря, дублирование и переупорядочение сообщений. Протокол должен обеспечить передачу сообщений от пользователя-передатчика пользователю-приемнику, причем к последнему они должны поступить в том же самом порядке, в каком были отправлены пользователем-передатчиком. Протокол Стеннинга в качестве сообщений отправляет номера последовательности по модулю N и помещает номер-сообщение в буфер, называемый окном передатчика. Во время эксперимента с протоколом с помощью пошаговой симуляции была воспроизведена специфическая последовательность событий, которую описал Каивола и которая приводит к рассогласованию между перепосылкой сообщений передатчиком и отправлением подтверждений приемником. В итоге на уровне протокола наблюдается непрерывная активность, тогда как на уровне пользователя всякая деятельность останавливается: со стороны передатчика новые сообщения не генерируются и со стороны приемника передачи новых сообщений на верхний уровень не происходит.

i-протокол является частью пакета GNU UUPC от Free Software Fondation и используется для передачи данных по линиям связи. По своей сути i-протокол — вариант протокола скользящего окна, отли-

чительная особенность i-протокола — минимизация числа перепосылок по сравнению с другими вариантами протокола скользящего окна. Основой для написания Estelle-спецификации i-протокола послужила спецификация на языке Promela. В ходе эксперимента в среде i-протокола была воспроизведена последовательность событий, приводящая к ошибке Каиволы в протоколе Стеннинга. Было выяснено, что несмотря на то, что любое подтверждение воспринимается передатчиком корректно (в отличие от протокола Стеннинга), воспроизведение этой последовательности событий приводит к ситуации $live - lock$ в i-протоколе. Более того, на основании проделанных экспериментов можно утверждать, что такая ситуация возникает в i-протоколе в результате любой последовательности событий, приводящей к состоянию, когда окна передачи полностью заполнены и ни одно подтверждение не получено.

Протокол Inres описывает взаимодействия между двумя протокольными объектами. Связь между ними происходит через ненадежную среду передачи, которая может терять сообщения, но не может их копировать или переупорядочивать. Каждый сеанс связи между протокольными объектами разбит на две фазы: фазу установления связи и фазу передачи данных. При написании системы Inres для организации моделирования действий пользователя дополнительно были реализованы процессы-пользователи. Кроме того, система была расширена для работы с многими пользователями. Для этой цели был реализован процесс, который во время функционирования по истечении установленного в таймере времени, создает экземпляры процессов пользователей и протокольных объектов. Эксперимент с протоколом Inres проводился в системе Design/CPN. Во время эксперимента была обнаружена семантическая ошибка, состоящая в дублировании данных, передаваемых между протокольными объектами. Такая ситуация возникает в том случае, если установленного в таймере времени в передающем протокольном объекте недостаточно для того, чтобы протокольные объекты успели обменяться сообщениями. При предположениях о времени, необходимом объектам для обмена данных, в протоколе были изменены временные характеристики, в результате чего поведение сети стало соответствовать ожидаемому.

В **приложении** приведены Promela- и Estelle-спецификации i-протокола.

Основные выводы и результаты

В рамках диссертации были получены следующие результаты.

— Разработан алгоритм перевода Estelle-спецификаций без динамических конструкций, но с задержками и приоритетами в ИВТ-сети.

— Разработан алгоритм перевода спецификаций языка SDL88 без динамических конструкций в раскрашенные сети Петри. Проведено моделирование спецификаций с операторами посылки сигналов, использующих как идентификаторы экземпляров процессов, так и маршрутизацию сигналов.

— Разработаны способ моделирования SDL-спецификаций с динамическими конструкциями и алгоритмы перевода динамических конструкций в раскрашенные сети Петри. Получены верхние оценки размера результирующей сети, подтверждающие эффективность алгоритма.

— Реализованы разработанные алгоритмы перевода Estelle- и SDL-спецификаций в соответствующие сетевые модели и проведены эксперименты по поиску семантических ошибок для таких коммуникационных протоколов, как протокол Стеннинга, *i*-протокол, Inres. В ходе экспериментов были обнаружены новые эффекты поведения протоколов.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Чурина Т.Г. Трансляция SDL-спецификаций в раскрашенные сети Петри// IV сибирский конгресс по прикладной и индустриальной математике (ИНПРИМ-2000):Тез.докл. — Новосибирск: Ин-т математики СО РАН, 2000. — С. 128.

2. Непомнящий В.А., Алексеев Г.И., Быстров А.В., Мыльников С.П., Е.В. Окунишникова Е.В., П.А. Чубарев П.А., Т.Г. Чурина Т.Г. Верификация коммуникационных протоколов, представленных на языках Estelle и SDL// Там же. — С. 123.

3. Чурина Т.Г. Моделирование динамических конструкций языка SDL посредством раскрашенных сетей Петри. — Новосибирск, 1999. — 35 с.— (Препр./АН РАН. Сиб. отд-ние. ИСИ; N 71)

4. Алексеев Г.И., Быстров А.В., Куртов С.А., Мыльников С.П., Непомнящий В.А., Окунишникова Е.В., Чубарев П.А., Чурина Т.Г. Использование сетей Петри для верификации распределенных систем, представленных на языке Estelle// Известия РАН. Сер. Теория и системы управления. — 1999. — N. 5. — С. 105–116.

5. Чурина Т.Г. Способ построения раскрашенных сетей Петри, моделирующих SDL-системы. — Новосибирск, 1998. — 56 с.— (Препр./АН РАН. Сиб. отд-ние. ИСИ; N 56)

6. Непомнящий В.А., Алексеев Г.И., Быстров А.В., Куртов С.А., Мыльников С.П. , Окунишникова Е.В., Чубарев П.А., Чурина Т.Г. Верификация Estelle-спецификаций распределенных систем посредством раскрашенных сетей Петри. — Новосибирск: ИСИ СОРАН, 1998. — 140 с.

7. Churina T.G., Okunishnikova E.V. Modelling Estelle specifications using Coloured Petri nets.// Joint Bulletin of the Novosibirsk Computing Center and the Institute of Informatics Systems. Ser. Computer Science. — 1998. — N 8. — P. 19–39.

8. Nepomniaschy V.A., Alekseev G.I., Bystrov A.V., Churina T.G., Mylnikov S.P.,Okunishnikova E.V. Towards Verification of Estelle-specified Communication Protocols: Coloured Petri Net Approach //Proc. Int. Conf. on Parallel Computing in Electrical Engineering. — Bialystok, Poland, 1998, P. 141–147.

9. Nepomniaschy V.A., Alekseev G.I., Bystrov A.V. , Churina T.G., Mylnikov S.P., Okunishnikova E.V. EPV — Petri Net Based Estelle Protocol Verifier //Proc. 1st Internati. Workshop on the Formal Description Technique Estelle. — Evry, France, 1998. — P. 101–108.

10. Churina T. G., Okunishnikova E. V. Coloured Petri nets approach to the validation of Estelle specifications // Proc. of Workshop on Concurrency, Specification and Programming. — Warsaw, Poland, 1997. — P. 25—36.

11. Nepomniaschy V. A., Alekseev G. I., Bystrov A. V., Churina T. G., Mylnikov S. P., Okunishnikova E. V. Petri net modelling of Estelle-specified communication protocols // Proc. 3rd Int. Conf. Parallel Computing Technologies. — Berlin a. o.: Springer-Verlag, 1995. — P. 94—108. — (Lect. Notes Comput. Sci., Vol. 964).

12. Окунишникова Е. В., Чурина Т. Г. Способ построения раскрашенных сетей Петри, моделирующих Estelle-спецификации // Проблемы спецификации и верификации параллельных систем. — ИСИ СО-РАН, Новосибирск, 1995. — С. 95—123.

13. Nepomniaschy V. A., Churina T. G., Okunishnikova E. V. Translation of Estelle protocol specification in coloured Petri nets. Extended Abstract // Proc. IFIP 15th Intern. Sympos. on Protocol Specification, Testing and Verification. — Warsaw, Poland, 1995. — P. 447—450.

14. Alekseev G. I., Bystrov A. V., Churina T. G., Mylnikov S. P. Petri-net based environment for the specification, analysis and simulation of concurrent systems // Specification, verification and net models of concurrent systems. — IIS, Novosibirsk, 1994. — P. 116—127.

Личный вклад автора

Все результаты, касающиеся моделирования и валидации SDL-спецификаций, получены автором самостоятельно. В работах по моделированию Estelle-спецификаций, выполненных в соавторстве, Т. Г. Чурина внесла следующий вклад: в работах [7 – 14] разработано отображение в ИВТ-сети структур данных, условий возможности E-перехода, входных переходов, сложных переходов, а также ряда стандартных кон-

струкций языка Estelle; разработаны основные принципы и алгоритм моделирования иерархической спецификации и шага выполнения; в работах [2, 4, 6] разработаны и описаны реализация алгоритма генерации сетевой модели по внутреннему представлению Estelle-спецификации, оптимизация сетевой модели и эксперименты с протоколом Стеннинга и i-протоколом.

Подписано в печать 19.05.2000 Объем 1,1 уч.-изд.л.
Формат бумаги 60×84 1/16 Тираж 100 экз.

НФ ООО ИПО "Эмари" РИЦ, 630090, г. Новосибирск, пр. Акад. Лаврентьева, 6