АВТОМАТИЧЕСКАЯ СИСТЕМА АНАЛИЗА И ВЕРИФИКАЦИИ ТЕЛЕКОММУНИКАЦИОННОЙ СИСТЕМЫ, ОПИСАННОЙ НА ЯЗЫКЕ MSC, С ПОМОЩЬЮ ФОРМАЛИЗМА СЕТЕЙ ПЕТРИ

Матвеева Л. Е.

Институт кибернетики им. В.М.Глушкова НАН Украины,
03680 Киев, просп. Академика Глушкова, 40, МСП Киев-187,
факс: (380)(44)266-74-18,
телефон: (380)(44)266-00-58, (380)(44)490-54-23,
email: luda@iss.org.ua

АНОТАЦІЯ

За останні 20 років формальні методи почали широко використовуватись для специфікації, аналізу, верифікації та відповідного тестування програмних та технічних систем і, зокрема, телекомунікаційних протоколів [1]. В даній роботі пропонується автоматизований технологічний процес формальної специфікації та верифікації телекомунікаційної системи. Формальна модель системи будується у вигляді ординарної мережі Петрі. Аналіз системи виконується за допомогою методів лінійної алгебри.

ABSTRACT

Last 20 years formal methods are being used widely to specify formally, analyze, verify and test software and hardware systems, particularly, telecommunication protocols [1]. In this paper automated system is presented which specify formally and verify the telecommunication system. The automated system applies the formal modeling technique of Petri nets and is based on linear algebra methods of analysis in order to research some properties of telephone system.

Введение. О формальных методах.

Формальные Методы есть набор методов и инструментальных средств, основанных на математическом моделировании и формальной логике, которые используются для формальной спецификации и верификации требований и архитектуры технических и программных систем. Формальные методы могут включать также автоматизированные доказательства ключевых свойств проектируемой системы. Возрастающая критичность и сложность программных приложений и технических систем приводят к растущему интересу к формальным методам. Формальные методы дополняют индуктивные методы, такие как тестирование, и помогают повышению уровня качества продукта выше того уровня качества, которое можно обеспечить лишь тестированием. Ниже перечислены некоторые из преимуществ эффективного применения формальных методов:

• Формальные методы помогают обнаруживать дефекты. Доказательством этого служат факты, когда с помощью формальных методов обнаруживались дефекты в системах программного обеспечения, уже прошедших обширное тестирование и доведенных до достаточно высокого уровня качества [5]. Индуктивная природа тестирования приводит к тому, что в сложных системах всегда будут существовать такие сценарии, которые не смогут быть протестированы по практическим соображениям.

• Формальные спецификации позволяют обнаруживать дефекты в требованиях и архитектуре ранее, чем они были бы обнаружены с помощью других средств, и значительно уменьшать степень влияния ошибок на понимание и реализацию правильных требований и архитектур проектов.

• Формализованные утверждения могут быть подвергнуты формальному анализу. Риска делать выводы о поведении системы, основываясь на конечном числе тестов, можно зачастую избежать путём использования методов математического доказательства. Эти методы позволяют большому (потенциально бесконечному) классу тестовых примеров быть полностью покрытым конечным доказательством, а также автоматически проверять формализованные утверждения.

• Использование формальных методов в некоторых случаях позволяет гарантировать отсутствие определённых дефектов.

Во многих работах формальные методы включают стандартизованные языки спецификаций такие, например, как язык язык LOTOS (Temporal Ordering Specification)[2], MSC (Message Sequence Charts) [3] и SDL (Specification and Description Language)[4] и другие широко используемые методы спецификации такие, как конечные автоматы, сети Петри [6,7] и темпоральные логики [8]. Реальная система подвергается формализации следующим образом. Сначала выделяются для рассмотрения некоторые её свойства. Затем посредством формальной спецификации они представляются в виде формальной модели, которая затем верифицируется. С

помощью различных инструментальных средств верификации производится доказательство того, имеет ли модель требуемые зафиксированные свойства реальной системы.

В процессе разработки и проектирования программного или технического обеспечения ЭВМ особенно актуальной является проблема автоматизации этих процессов с целью получения качественного продукта. Часто инженеры-разработчики и тестировщики-верификаторы используют в своей деятельности различные языковые средства, что в конечном итоге может привести к двусмысленному (неоднозначному) пониманию одних и тех же мест проекта, неточностям или даже противоречиям в исходных требованиях на разработку и, наконец, к неполноте описаний. Инженеры-разработчики, как правило, используют языки (VHDL, MSC, SDL, UML и т.п.), которые предназначены для решения задач проектирования, а тестировщики-верификаторы – языки (математической логики, теории автоматов, алгебраические, сетевые и т.п.), с помощью которых решаются задачи верификации и тестирования. Выходом из такой ситуации является разработка автоматизированных интерфейсов между языками инженеров-проектировщиков и тестировщиков-верификаторов.

1. Постановка задачи.

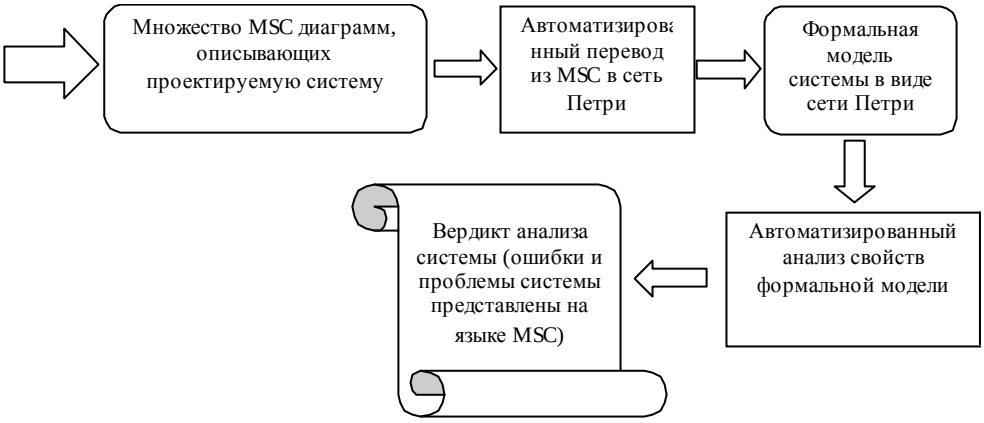

Рисунок 1

Представим постановку задачи, решение которой описано в данной работе: требуется построить технологический процесс (см. рисунок 1), который позволяет анализировать программную или техническую систему, описанную на языке MSC. При этом главные свойства данного технологического процесса следующие:

- Процесс полностью автоматический
- Входной язык и язык выходных вердиктов являются рабочими языкам инженеров-разработчиков и, следовательно, не требуют специальной математической подготовки.

В данной работе используется алгоритм перевода набора MSC-диаграмм в сети Петри (СП), созданный автором данной работы в коллективе соавторов [9]. Данный алгоритм работает с некоторым базовым подмножеством языка MSC. С помощью полученной в результате перевода СП автоматически проверяются свойства проектируемой системы с использованием методов линейной алгебры.

Рассматривается пример из области телекоммуникации — базовая телефонная модель POTS (Plain Old Telephony Service/System). Модель данной аппаратной системы представлена в виде ординарной СП с одноцветными фишками [7]. Формальный анализ свойств модели проводится посредством решения систем линейных однородных и неоднородных диофантовых уравнений (СЛОДУ, СЛНДУ) над множеством натуральных чисел методом TSS[10].

2. Базовые определения.

Приведём необходимые понятия и определения.

Определение 1: **сетью** называется тройка $N=(P,T,F)$, где P и T — некоторые непустые множества, элементы которых называются соответственно местами и переходами, $F\subseteq(P\times T)\bigcup(T\times P)$ — бинарное отношение инцидентности между местами и переходами. На основе отношения инцидентности F вводится **функция инцидентности** $\overline{F}:(P\times T)\bigcup(T\times P)\to \mathrm{N}$, где N — множество натуральных чисел.

Пусть $x\in P\bigcup T$ – произвольный элемент сети. Для элемента x определим множества его входных и выходных элементов: ${}^{\bullet}x=\{y\mid yFx\}$ и $x^{\bullet}=\{y\mid xFy\}$.

Для сети $N=(P,T,F)$ требуется выполнение следующих условий:

C1/ $P\bigcap T=0$,

C2/ $(F\neq 0)\wedge\left[(\forall x\in P\bigcup T)(\exists y\in P\bigcup T): xFy\vee yFx\right]$,

С3/ $(\forall p_1, p_2 \in P)\left({}^\bullet p_1 = {}^\bullet p_2 \wedge p_1^\bullet = p_2^\bullet \Rightarrow p_1 = p_2\right)$.

Определение 2: **разметкой сети** $N = (P, T, F)$ называется функция $\mu : P \to \mathrm{N}$. Если $\mu(p) = k \in \mathrm{N}$, то говорят, что в месте $p \in P$ имеется k фишек или говорят, что функция помещает в место p сети N k фишек.

Определение 3: **сеть Петри (СП)** это пара (N, μ_0), где N — некоторая сеть, а μ_0 — некоторая начальная разметка сети N.

Переход $t \in T$ допустим на разметке μ СП $(N, \mu_0) \Leftrightarrow \forall p \in {}^\bullet t : \mu(p) \geq \overline{F}(p, t)$. **Срабатывание перехода** t при разметке сети μ порождает разметку сети μ' по следующему правилу: $\forall p \in P : \mu'(p) = \mu(p) - \overline{F}(p, t) + \overline{F}(t, p)$. Разметка μ **достижима** из начальной разметки μ_0, если существует последовательность разметок $\mu_0, \mu_1, ..., \mu$ и последовательность переходов $\sigma = t_1, t_2, ..., t_n$ такие, что: $\mu_0 \xrightarrow{t_0} \mu_1 \xrightarrow{t_2} ... \xrightarrow{t_n} \mu$.

Место p СП называется ограниченным, если существует число n такое, что для $\forall \mu : \mu(p) \leq n$. СП называется **ограниченной** сетью, если любое её место ограниченно.

Место p СП называется безопасным, если $\forall \mu : \mu(p) \leq 1$. СП называется **безопасной**, если все её места безопасны.

СП называется **ординарной**, если кратность всех её дуг равна 1.

СП называется **чистой**, если для всякого перехода СП ${}^\bullet t \cap t^\bullet = \varnothing$.

Граф разметок СП — ориентированный граф, множество вершин которого образовано множеством достижимых в СП разметок. Из вершины μ в вершину μ' ведёт дуга, помеченная символом t ,если и только если $\mu \xrightarrow{t} \mu'$, где μ и μ' — достижимые разметки, а t — переход данной СП.

Циклом в СП называется последовательность $x_0, x_1, ...x_k$, где x_i является или местом, или переходом, и выполняются следующие условия $x_{i+1} \in x_i^\bullet$; и $x_0 = x_k$.

Непустое подмножество мест Q в ординарной СП называется **тупиком (ловушкой)**, если каждый переход, выходное (входное) место которого принадлежит Q, содержит в Q и своё входное (выходное) место.

Матрица инцидентности СП, уравнение состояния и инварианты СП. Для СП и каждой разметки μ, достижимой из начальной разметки μ_0, можно сформулировать следующую задачу, которую называют **уравнением состояния**: $Ax = \mu - \mu_0$, $x \geq 0$, где x — вектор Парика (или вектор-счётчик) для последовательности переходов σ, а A — **матрица инцидентности** СП с целочисленными коэффициентами, которые определяются с помощью уравнения: $a_{ij} = \overline{F}(t_j, p_i) - \overline{F}(p_i, t_j)$, где $\overline{F}$ — функция инцидентности. Справедливость этого уравнения следует из правила сохранения фишек, так как коэффициенты a_{ij} матрицы инцидентности СП представляют собой число фишек, которые перемещаются, изменяются и добавляются в место p_i при срабатывании перехода t_j в этой СП.

В [11] была доказана теорема о том, что если некоторая заданная разметка μ СП достижима из заданной начальной μ_0, то существует решение соответствующего этой сети уравнения состояния, представляющее вектор Парика. Если уравнение состояния не имеет решения, то разметка сети μ не достижима из μ_0. Иными словами, существование решения уравнения состояния является лишь необходимым условием достижимости разметки μ.

Определение 4: положительные, целочисленные решения однородной системы (СЛОДУ) $Ax = 0$, производной от уравнения состояния, называются **T-инвариантами** СП. Положительные, целочисленные решения однородной системы (СЛОДУ) $A^T y = 0$ называются **S-инвариантами** СП, где A^T — транспонированная матрица инцидентности.

3. Алгоритм перевода набора MSC диаграмм в сеть Петри

MSC — технология моделирования с возможностью графического интерфейса, которая была стандартизована ITU (International Telecommunication Union) [3]. Обычно, она применяется в области телекоммуникационных приложений, так как такие приложения имеют свойства распределённых реактивных систем с режимом работы в реальном времени, часто в комбинации с SDL[4]. И именно эти свойства таких систем делают MSC, со своей возможностью описания сценариев, чрезвычайно подходящим как для спецификации, так и для тестирования. То есть, MSC может применяться на протяжении всего цикла проектирования системы, начиная со спецификации системы, и вплоть до написания тестов на готовый

продукт. MSC описывает поток сообщений между инстанциями, которые представляют собой асинхронно взаимодействующие объекты или сущности системы типа блоков, сервисов или процессов системы. Одна диаграмма MSC описывает некоторый фрагмент поведения системы или же сценарий взаимодействия между инстанциями. MSC имеет два вида синтаксического представления: текстовое и графическое, между которыми согласно стандарту MSC'2000 [3] установлено взаимооднозначное соответствие. Базовыми элементами языка являются те, которые необходимы для определения потока сообщений, например: инстанции, передачи сообщений и условия. Посылки и получения сообщений упорядочены вдоль инстанций таким образом, что событие посылки сообщения происходит строго раньше, чем событие его получения. Существует ещё одно правило в стандарте MSC'2000, задающее порядок событий коммуникации на инстанциях: то, что графически находится выше – происходит раньше того, что расположено графически ниже. MSC-диаграмма определяет частичный порядок событий, которые она описывает. Отношение частичного порядка есть транзитивное, антисимметричное и рефлексивное отношение.

Первые результаты по определению семантики MSC, основанной на сетях Петри, были описаны в [13]. В настоящее время работа по решению задачи перевода диаграмм MSC в сети Петри продолжается, результаты представлены, например, в [14], [15], [16]. Авторы разрабатывают семантику для стандартов MSC’96 и MSC’2000, основанную на сетях Петри, а также работают над преобразованием диаграмм в сети, однако автоматический перевод в этих работах не рассматривается.

Описание алгоритма. Приведем формализованное описание алгоритма перевода.

ВХОД: набор MSC–диаграмм в базовом подмножестве языка MSC’2000 [3], описывающий систему.

ВЫХОД: сеть Петри (СП), адекватная исходным MSC-диаграммам.

МЕТОД: Перевод каждой MSC диаграммы из набора в СП производится в два этапа. В процессе перевода происходит синтез общей СП из составных СП, соответствующих исходным MSC-диаграммам.

Начало

Этап 1. Построение по MSC-диаграмме графа частичного порядка (partial-order graph) событий системы с учётом статических требований (static requirements) стандарта языка MSC’2000 [3], регламентирующих порядок событий коммуникации.

Этап 2. Перевод графа частичного порядка, полученного на этапе 1, в сеть Петри.

Конец

Шаги синтеза осуществляются в соответствии с представленным ниже правилом, при условии выполнения следующих требований.

Требования:

1. Рассматривается лишь подмножество языка MSC’2000, состоящее из инстанций, передачи сообщений и условий установочного типа, в текстовом представлении. Кроме того, принимается, что все диаграммы, поступающие на вход, являются синтаксически правильными и удовлетворяют статическим требованиям стандарта MSC’2000. Например, для каждого события выдачи сообщения диаграмма должна иметь парное событие получения сообщения.

2. Требуется именование инстанций полное и точное.

3. Требуется именование условий полное и точное, значения же условий никак не влияют на синтез.

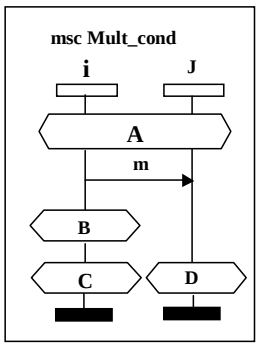

Рисунок 2.

4. Считается, что каждая инстанция каждой диаграммы должна обязательно иметь начальное и конечное условия, как локальные для неё, так и, возможно, разделяемые несколькими инстанциями. Данное требование не относится к случаю создания и завершения инстанции в рамках данной диаграммы. Условие называется *начальным*, если в диаграмме нет событий и условий ему предшествующих, и *конечным*, если в диаграмме после данного условия нет никаких событий и условий. Возможны не только единичные, но и множественные начальные и конечные условия, которые являются конъюнкцией условий, включающих начальные условия, и, соответственно, конъюнкцией условий, включающих конечные условия. Например, на рисунке 2 условия B и C являются множественным конечным условием, D — единичным конечным условием или просто конечным условием, A является начальным условием.

5. Следующее правило важно для «склеивания» по *перекрывающимся* (overlapped) и *множественным* (multiple) условиям: a) перекрывающиеся и множественные условия выполняются одновременно, и их комбинация равняется конъюнкции условий, b) порядок перечисления в диаграмме не важен, в силу

одновременности выполнения условий, с) эти условия могут склеиваться независимо друг от друга. Рассмотрим пример на рисунке 3. Каждая из MSC диаграмм X1, X2, X3 является возможным продолжением msc X. Условия B и C являются перекрывающимися в msc X и msc X3.

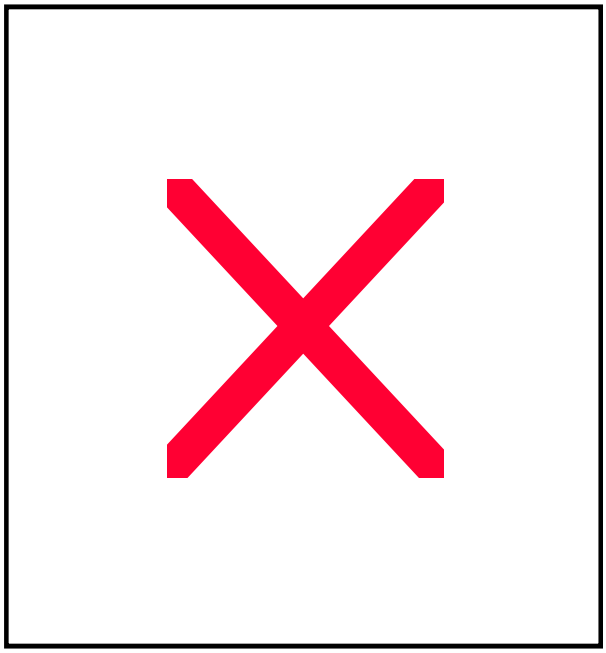

Рисунок 3

6. Считается, что для каждой диаграммы из набора, описывающего систему, выполняется следующее: каждые начальное и конечное условия диаграммы должны покрывать минимум одну инстанцию, на которой происходит какое-либо событие (в данном случае, выдача или приём сообщения).

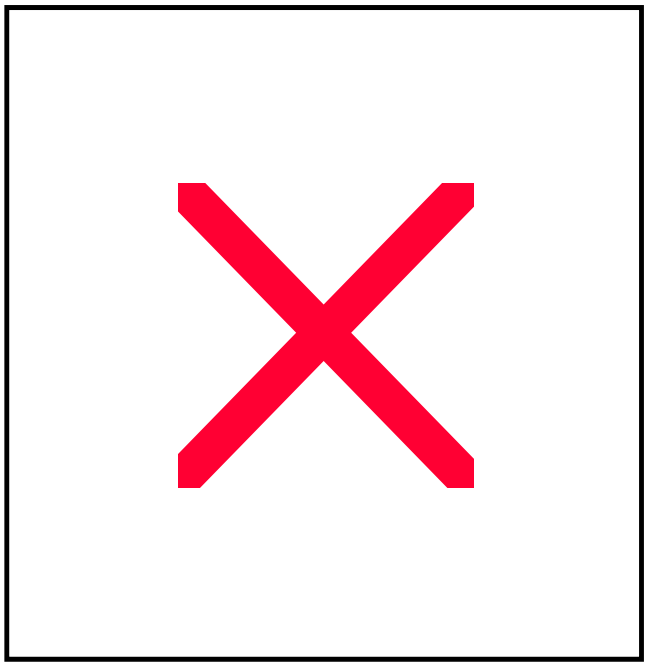

Рисунок 4

7. «Склеивание» принудительное и осуществляется согласно следующему правилу.

Правило:

Если у первой диаграммы существует конечное условие, которое соответствует (имеет то же имя) начальному условию второй диаграммы, причём данные условия охватывают одно и то же множество инстанций обоих диаграмм, то первую и вторую диаграммы можно «склеить» по данному условию, не зависимо от того, является ли это условие глобальным или локальным для своих диаграмм. Например, на рисунке 4 msc Z является синтезом или композицией msc X и msc Y. Штриховая линия показывает места «склеивания». Это правило предполагает также возможность не только «склеивать» по данному условию две диаграммы в одну, но также «склеивать» вместе более, чем две диаграммы одновременно (множественное «склеивание»). Альтернативы (выбор варианта возможной трассы системы) рассматриваем в смысле оператора недетерминированной альтернативной композиции в противовес оператору с отложенным выбором. В

результате такого склеивания получается MSC диаграмма, описывающая множество альтернативных возможных трасс системы.

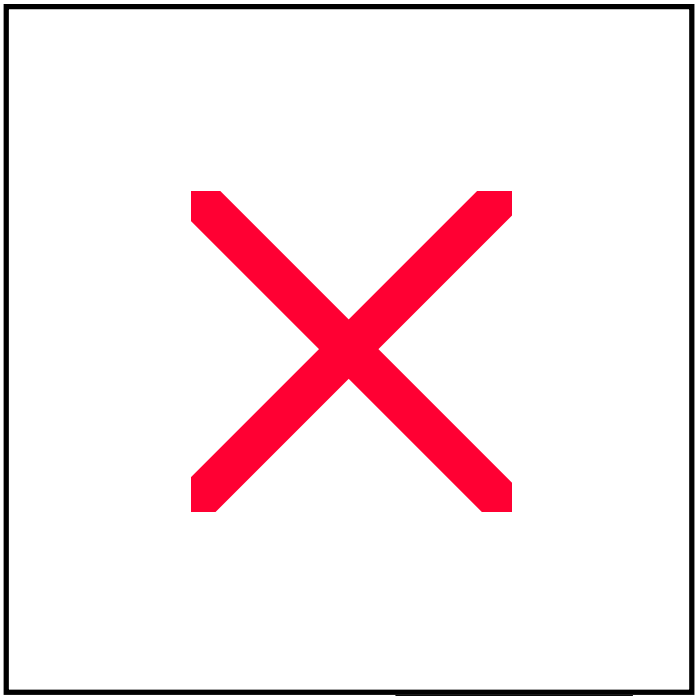

Рисунок 5

Рассмотрим, например, рисунок 5. Существует два возможных продолжения диаграммы msc X: это msc Y и msc Z. Причём, выбор происходит, когда встречается условие C, а не откладывается до того момента, когда события представленные диаграммой msc Z (msc Result 1) и диаграммой msc Y (msc Result 2) начнут друг от друга отличаться, а именно, до событий передачи сообщений m и k.

Все приведенные выше требования и правила не противоречат стандарту MSC'2000, а лишь уточняют его. Доказательство корректности алгоритма приводится в [9] и базируется на использовании алгебры процессов ACP [19].

Ключевую роль в синтезе из полученных в процессе перевода СП одной общей СП играет **таблица** начальных и конечных условий исходных диаграмм, так как вся необходимая информация для композиции находится в ней. Она формируется на этапе 1 в процессе последовательной обработки MSC диаграмм. Таким образом, склеивание происходит последовательно по мере перевода частей системы, с использованием таблицы начальных и конечных условий. В процессе такого перевода формируется ещё одна **таблица,** для поддержания постоянной связи между описанием системы в языке MSC и её представлением в виде сети Петри. Эта таблица необходима для того, чтобы результаты последующего анализа и верификации системы, выполняемые с помощью сетей Петри, представить инженеру-разработчику в том же формате MSC диаграмм.

4. Основные методы анализа СП

Известны следующие основные методы анализа СП: анализ с помощью **графа разметок** СП [7], методы **линейной алгебры** [11,12], техника **развёртки** СП в сеть процесса с последующим применением методов линейной алгебры [17], техника **редуцирования** СП [11].

Первый метод применяется ко всем типам СП. Автоматическая верификация с помощью графа разметок есть задача model checking, которая подразумевает исчерпывающий поиск по всему пространству состояний модели. Но при таком методе анализа возникает проблема взрыва пространства состояний (state explosion problem), которая является ограничивающим фактором в применении любых методов автоматической верификации к системам большого размера.

Второй метод анализа является эффективными, но во многих случаях применим только к специальным подклассам СП или к специальным модельным ситуациям. Одним из основных алгебраических методов анализа поведенческих свойств СП есть анализ её инвариантов. Инварианты СП описывают статические и некоторые динамические свойства СП. К статическим (структурным) свойствам СП относятся следующие свойства: структурная живость, управляемость, ограниченность и структурная ограниченность, консервативность и частичная консервативность, повторяемость и частичная повторяемость, непротиворечивость и частичная непротиворечивость. К динамическим свойствам СП относят наличие или отсутствие дедлока (deadlock), свойство взаимного исключения (mutual exclusion), свойство справедливости (fairness) и т.п. Для верификации динамических свойств следующих подклассов СП: СП без циклов и СП с циклами только в виде ловушек или только в виде тупиков, может использоваться техника уравнения состояния. Для остальных СП, не принадлежащих данным подклассам, имеет силу тот факт, что согласно

теореме об уравнении состояния [11] существование решения является лишь необходимым условием достижимости.

Технику развёртки выделяем в отдельную группу, так как этот метод позволяет, во-первых, обрабатывать более широкий подкласс СП, чем в случае второго метода анализа, во-вторых, анализировать более широкий набор свойств систем. Понятие развёртки СП было введено как средство описания параллельной семантики сетей, и его целью являлось избежание проблемы взрыва пространства состояний. Алгоритм развёртки СП сводится к построению некоторой конечной начальной части сети именуемый префиксом, который включает все достижимые состояния исходной СП и заканчивается указанием на бесконечное повторение её периодического фрагмента. Префикс затем используется для верификации моделируемой системы. Таким образом, такая развёртка СП позволяет, во-первых, применять методы линейной алгебры, преодолев ограничения, связанные с подклассами СП и со спецификой постановки задачи анализа модели, и, во-вторых, решить проблему взрыва пространства состояний.

Анализ, связанный с редукцией СП до более простых сетей, состоит в нахождении множеств преобразований сети, которые сохраняют определённые свойства. Смысл этих редукций очевиден: для того чтобы упростить анализ большой системы (упрощаются граф разметок и матрица инцидентности), её модель в виде СП может быть редуцирована, а именно, по определённым правилам может быть уменьшено её количество мест и переходов. При этом будут сохранены некоторые определённые свойства, которые анализируются в модели. Следующие шесть преобразований СП, подробно описанные в [11], а именно: слияние серии мест, слияние серии переходов, слияние параллельных мест, слияние параллельных переходов, удаление мест-петель, удаление переходов-петель, сохраняют следующие свойства СП: структурные ограниченность и живость, безопасность.

5. Пример перевода и анализа базовой модели POTS

Рассмотрим полный набор MSC-диаграмм (рисунок 6), описывающий работу реальной телефонной системы, состоящей из базовых сервисов и традиционно именуемой Plain Old Telephony Service или Plain Old Telephony System, POTS. Отметим лишь, что MSC-диаграммы №3, 4 и 8 на рисунке 6 должны повториться симметрично относительно m-й/n-й инстанций. Базовая функциональность такой системы включает в себя: возможность снять телефонную трубку, набрать номер телефона, а также связь с телефоном, соответствующим номеру (то есть разговор двух абонентов). В модели также представлены состояния занятости абонента и отсутствия абонента (никто не снимает трубку). Относительно объектов MSC-диаграмм известно, что каждый телефон может находиться в одном из двух состояний — свободен (free) и занят (busy), а также, что телефонная станция имеет k каналов связи для K абонентов ($k < K$), где K — число абонентов, обслуживаемых данной телефонной станцией. Телефонная станция представлена на языке MSC двумя инстанциями: Channel (означает ресурс канала связи) и mem (означает запоминающее устройство телефонной станции).

Применим к данному набору диаграмм алгоритм автоматического перевода (см. раздел 3) и, редуцировав «линейные участки» СП (см. раздел 4), получим СП, представленную на рисунке 7. Данная СП для POTS учитывает все возможные ситуации: соединение и разговор двух абонентов, занятость абонента и отсутствие абонента (никто не снимает трубку). В этой СП m free(n free) означает, что m(n)-ый телефон свободен; m busy (n busy) — m(n)-ый телефон занят; начальная разметка M_0 = (1,0,0,0,0,0,0,1,0,0,k), где k — число каналов для связи. Полученная сеть симметрична. То есть, СП, моделирующая вызов n-м абонентом m-го абонента, выглядит аналогично и имеет с данной СП общие места первое, второе, восьмое, девятое и одиннадцатое. Склеивая их и сохраняя все переходы в обеих сетях, получим полную СП для m-го и n-го абонентов. Общим местом для СП, моделирующей попарные соединения всех абонентов сети, является одиннадцатое место. Таким образом, получено представление системы в виде формальной модели. Целью такого представления являются анализ и верификация реальной системы. Причём для анализа свойств системы достаточно проанализировать СП, моделирующую вызов n-м абонентом m-го абонента. Требуется проверить, выполняет ли система те функции, для которых она предназначена и могут ли в ней возникнуть ошибки и аварийные ситуации. В данной СП (рисунок 7) переходы и места интерпретируются соответственно условиям и событиям передачи сообщений данных MSC-диаграмм (рисунок 6) следующим образом.

Описание переходов данной СП:

t1 = offhook(m), t2 = dial_n, t3 = onhook(m), t4 = busy, t5 = onhook(m), t6 = ring(m,n), t7 = offhook(n), t8 = onhook(n), t9 = onhook(m), t10 = onhook(m), t11 = offhook(n), t12 = onhook(n);

где offhook(m) означает, что m-й абонент поднимает трубку телефона, onhook(n)/onhook(m) означает, что n-й/ m-й абонент кладёт трубку телефона, ring(m,n) означает звонок m-го абонента n-му.

Описание мест данной СП:

P1 = m free, P2 = m busy, P3 = dial state, P4 = NW_dial, P5 = busy state, P6 = ringing state,
P7 = connected, P8 = n free, P9 = n busy, P10 = dial state, P11 = NW free,

Анализ инвариантов для приведенной СП позволяет установить следующие свойства этой модели.

Ограниченность. Данная СП структурно ограничена и, следовательно, ограничена, как следует из множества её S-инвариантов: s^1 = (0,1,1,0,0,0,0,0,0,0,0), s^2 = (0,0,0,0,0,1,0,1,0,0,0), s^3 = (1,0,0,1,1,1,1,0,0,0,0),

s^4 = (0,1,0,1,1,0,1,0,1,1,0), s^5 = (1,0,1,0,0,0,0,0,0,0,1,), s^6 = (0,0,0,0,0,0,0,1,1,1,1), s^7 = (1,0,0,1,1,0,1,0,1,1,1).

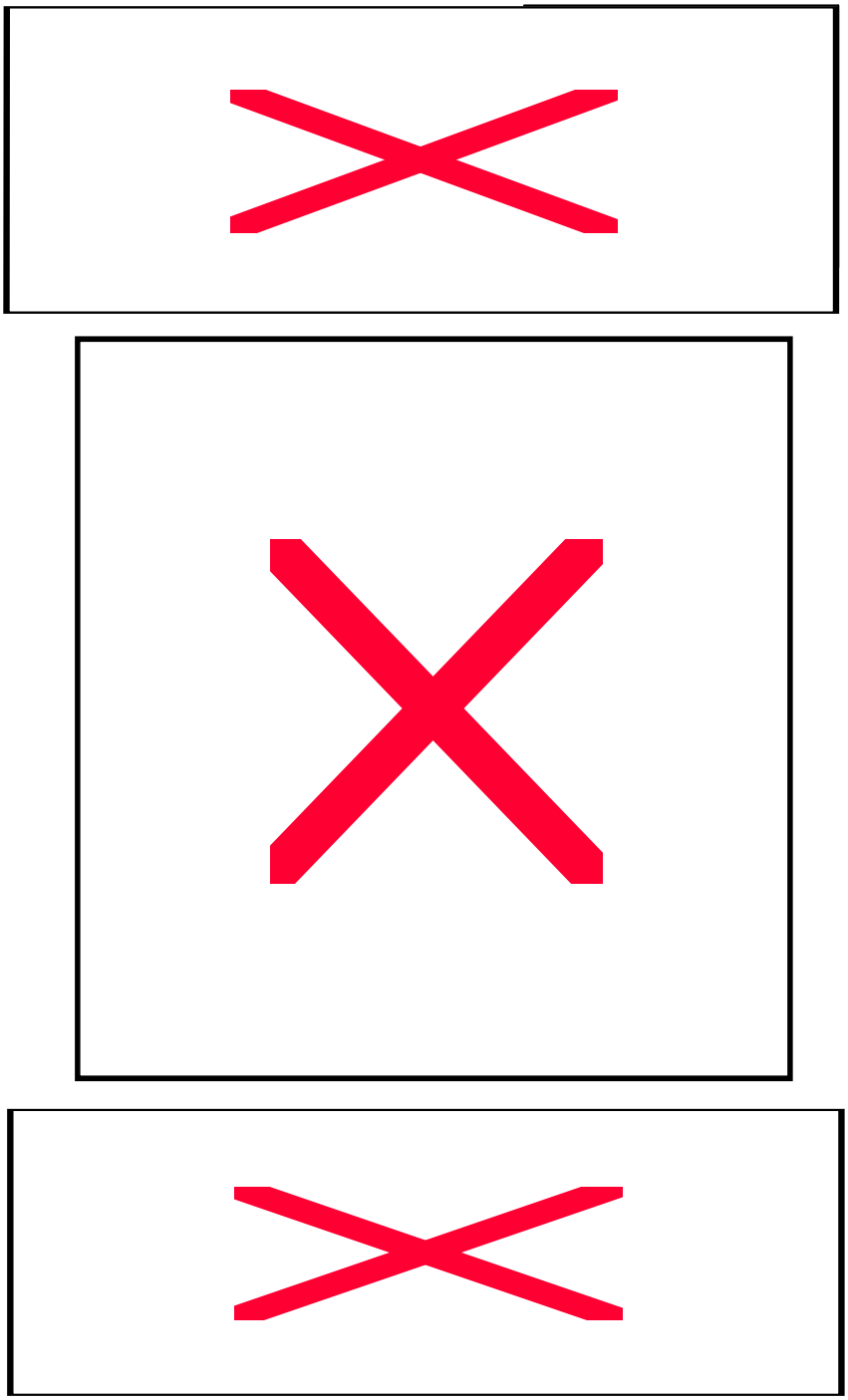

Рисунок 6. MSC-диаграммы, представляющие все возможные протоколы POTS

Действительно, все места СП покрываются ненулевыми координатами. Более того, для места 11 получаем такую верхнюю оценку числа фишек при начальной разметке $M_0 = (1,0,0,0,0,0,0,1,0,0,k)$:

$$M(11) \leq \min [M_0 s_5^T / s_5 (11), M_0 s_6^T / s_6 (11), M_0 s_7^T / s_7 (11)] = \min [k+1/1, k+1/1, k+1/1] = k+1.$$

Это свойство означает, что в телефонной сети может одновременно происходить не более k+1 телефонных разговоров.

Повторяемость. СП обладает свойством повторяемости, поскольку все переходы покрываются ненулевыми координатами из множества *T*-инвариантов:

$t_1 = (1,0,1,0,0,0,0,0,0,0,0,0)$, $t_2 = (0,0,0,0,0,0,0,0,0,0,1,1)$, $t_3 = (1,1,0,1,1,0,0,0,0,0,0,0)$,

$t_4 = (1,1,0,0,0,1,0,0,0,1,0,0)$, $t_5 = (0,1,0,0,0,1,1,1,0,0,0,0)$, $t_6 = (1,1,0,0,0,1,1,0,1,0,0,1)$.

Физическая интерпретация этого свойства означает, что в телефонной сети повторное соединение любых двух абонентов возможно практически бесконечное число раз.

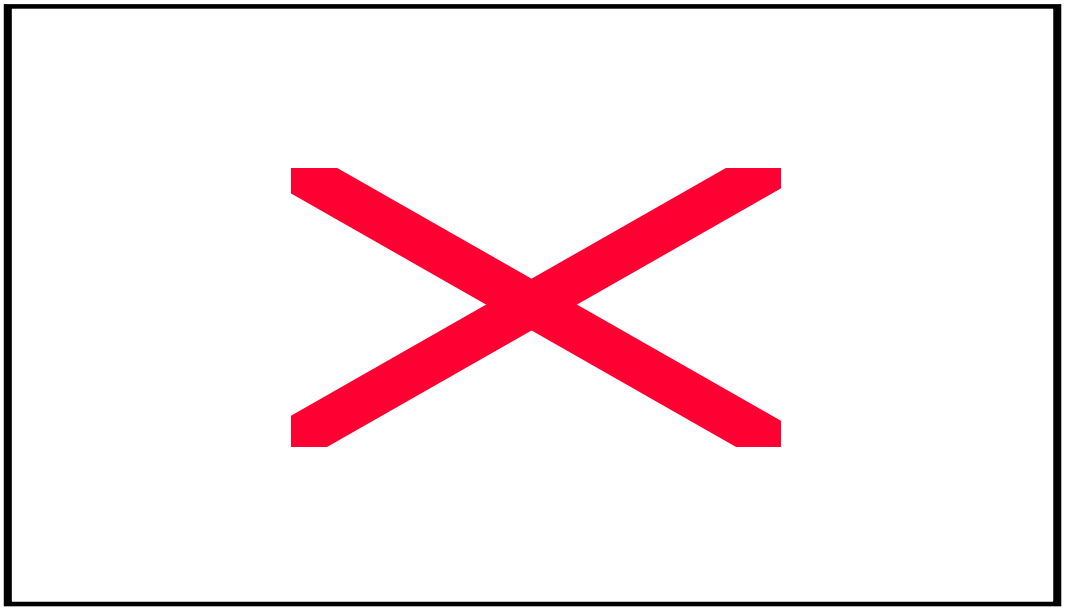

Рисунок 7. СП, построенная по набору MSC-диаграммам после редукции и моделирующая работу базовой системы POTS

Непротиворечивость. СП обладает свойством непротиворечивости, поскольку любая разметка M является достижимой из самой себя. Это значит, что телефонная сеть всегда приходит в исходное состояние.

Взаимное исключение в данной СП состоит в том, что никакой телефон не может быть одновременно в состоянии занят и свободен. Это свойство имеет место, т.к. разметки M = (1,1,0,0,0,0,0,1,0,0,k-1), M = (0,1,0,0,0,0,0,1,1,0,k-1) не являются достижимыми в этой СП.

Установленные в результате анализа свойства фиксируются как базовые свойства базовой модели POTS. Данная телефонная сеть обладает следующими свойствами: количество телефонов, подключенных в сеть, и каналов связи определяется конфигурацией сети и неизменно в любой момент времени в рамках данной конфигурации; имеется ограничение на использование ресурсов — одновременно в сети не может происходить более k+1 попарных соединений абонентов; телефонная сеть должна возвращаться в исходное состояние после разговора двух абонентов; возможно бесконечное повторение связывания двух абонентов между собой независимо от текущей конфигурации сети; отсутствует возможность связывания абонента с самим собой; невозможно попадание телефонной сети в тупиковое состояние.

Таким образом, в результате проведенного анализа можно сделать вывод, что построенная формальная модель имеет требуемые зафиксированные свойства данной реальной системы, и данный набор MSC-диаграмм корректно описывает модель POTS. Напомним, что в процессе перевода из MSC в СП формируется таблица, которая служит для поддержания постоянной связи между описанием системы в языке MSC и её представлением в виде сети Петри. С помощью этой таблицы найденные проблемы и ошибки системы могут представляться инженеру-разработчику в формате MSC диаграмм.

Заключение. Дальнейшие исследования.

Формальная спецификация и верификация программных и аппаратных систем является одним из путей повышения надёжности проектируемых систем, а также ускорения процесса их проектирования. Использование формальных методов в процессе проектирования не гарантирует корректность априорно, так как они верифицируют формальную модель, а не реальную систему. Однако они могут значительно углубить понимание разрабатываемой системы, выявляя и демонстрируя противоречия, двусмысленности и неполноту, которые иначе можно было бы не обнаружить.

Подводя итог, отметим, что главными свойствами данного технологического процесса есть следующие: во-первых, процесс полностью автоматизированный, во-вторых, входной язык и язык выходных вердиктов являются рабочими языкам инженеров-разработчиков и, следовательно, не требуют специальной математической подготовки. Среди особенностей алгоритма перевода, входящего в данный процесс, хотелось бы выделить работу с условиями языка MSC, т.к. из литературы следует, что работа с условиями при таком переводе является наиболее трудным участком [18].

Предложенный в работе технологический процесс воспринимает на вход лишь подмножество языка MSC, поэтому естественным продолжением начатых исследований является расширение этого подмножества до максимально возможных размеров или до всего языка MSC. Очень важным является получение необходимого опыта для последующего построения автоматизированных переводов с языков SDL, UML и т.п.

Конечной целью данных исследований является разработка единой технологической линии частично или полностью автоматизированной, позволяющей одновременно осуществлять процессы проектирования, анализа, верификации и тестирования различных свойств проектируемой системы.

Література

1. *Wing J.M.* A specifier's introduction to formal methods. IEEE Computer, September, 1990. - PP. 8— 24
2. *Bolongesi T,. Brinksma E*. Introduction to ISO specification language LOTOS Computer Networks and ISDN Systems, vol. 14. - 1987. - PP. 25—59
3. ITU-TS Recommendation Z.120: Message Sequence Chart (MSC). ITU-TS, Geneva, 2000.
4. *Saracco R., Tilanus A.J*. CCITT SDL: Overview of the language and its applications Computer Networks and ISDN Systems, vol. 14. - 1987. - PP. 65—74
5. *Miller S.P., Srivas M.* Formal Verification of the AAMP5 Microprocessor – A Case Study in the Industrial Use of Formal Methods, in Proceedings of the 1995 Workshop on Industrial-Strength Formal Specification Techniques (WIFT'95), IEEE Computer Society, Orlando, Florida, USA, April 5-8, 1995.
6. *Jensen K.* EACTS Colored Petri Nets. vol. 1, Springer-Verlag, 1992 EACTS Temporal Logic of Programs. Springer-Verlag, 1987
7. *В.Е.Котов* Сети Петри. М.:Наука, 1984, 157 стр.
8. *Kröger K. EACTS Temporal Logic of Programs. Springer-Verlag, 1987*
9. *С.Л.Крывый, Л.Е.Матвеева, М.В.Лопатина*. Automatic Transformation of MSC Diagrams into Petri Nets, in Proceedings of SCI'2003, Orlando, USA, 29-31 July 2003, Vol. 5, pp. 140-146
10. *Крывый С. Л.* О некоторых методах решения и критериях совместности систем линейных диофантовых уравнений в области натуральных чисел. Кибернетика и сист. анализ.-1999.-№4.-С.12-36
11. *Murata T.* Petri Nets: Properties, Analysis and Applications. In "Proceedings of the IEEE", 1989, vol.77, № 4, P. 541-580.
12. *Esparza J., Melzer S.* Verification of Safety Properties Using Integer Programming: Beyond the State Equation. Formal Methods in System Design, 2000, №16, P. 159-189.
13. *J. Grabowski, P. Graubmann, E. Rudolph.* Towards a Petri Net Based Semantics Definition for Message Sequence Charts, In O.Fergemand and A.Sarma, editors, SDL'93 Using Objects, Proceedings of the 6th SDL Forum, Darmstadt, 1993. Elsevier Science Publishers B.V.
14. *O. Kluge*. Time in Message Sequence Charts Specifications and How to Derive Stochastic Petri Nets. In: Proceedings of the Third International Workshop on Communication Based Systems (CBS3), Berlin, March 2000.
15. *S. Heymer*. A Semantic for MSC based on Petri-Net Components, SIIM Technical Report A-00-12, Informatik-Berichte Humboldt-Universität zu Berlin, June 2000.
16. *O. Kluge, J. Padberg, H. Ehrig*. Modeling Train Control Systems: From Message Sequence Charts to Petri Nets, Technische Universität Berlin,http://cs.tu-berlin.de/SPP/index.html, 2001
17. *K.L.McMillan*. Using Unfolding to Avoid the State Explosion Problem in the Verification of Asynchronous Circuits. Proc. 4th Workshop on Computer Aided Verification, LNCS 663, pp.164-174, 1992.
18. *S. Mauw, M.A. Reniers*. Thoughts on the meaning of conditions. Experts meeting SG10, St.Petersburg TD9016, ITU-TS,1995.
19. *J.A. Bergstra, J.W. Klop*. Process Algebra for Synchronous Communication, Inf.&Control 60,pp.109-137,1984.