*Модел. и анализ информ. систем.* Т. **21**, № **6** (2014) 94–106


УДК 519.681.3

# Анализ и верификация MSC-диаграмм распределённых систем с помощью раскрашенных сетей Петри [1]

Черненок С. А., Непомнящий В. А.

*Институт систем информатики имени А. П. Ершова СО РАН*
*630090 Россия, г. Новосибирск, проспект Лаврентьева, 6*

*e-mail: chernenoksergey@gmail.com, vnep@iis.nsk.su*

*получена 12 октября 2014*

**Ключевые слова:** спецификация, трансляция, верификация, распределённые системы, телекоммуникационные протоколы, MSC-диаграммы, раскрашенные сети Петри

Стандартный язык диаграмм последовательных сообщений MSC предназначен для описания сценариев взаимодействия объектов. Благодаря своей выразительности и простоте MSC-диаграммы широко применяются на практике на всех этапах проектирования и разработки программных систем. В частности, язык MSC используется для спецификации поведения в распределенных системах и коммуникационных протоколах. В работе рассматривается метод анализа и верификации диаграмм MSC и HMSC. Метод основывается на трансляции конструкций (H)MSC в раскрашенные сети Петри. Описываемые правила трансляции охватывают большинство конструкций стандарта, включая концепцию данных. Приводятся оценки размера сетей Петри, полученных в результате трансляции. Свойства построенных сетей анализируются и верифицируются с помощью известной системы CPN Tools и системы автоматической верификации на основе SPIN. Работоспособность данного метода продемонстрирована на примере.

## Введение

Диаграммы последовательностей сообщений (MSC-диаграммы) являются популярным языком сценариев, предназначенным для формализации и анализа системных требований на этапе проектирования программного обеспечения. Они обладают достаточной выразительной силой для описания взаимодействия системных компонент в распределенных системах и протоколах [9]. MSC и их расширение HMSC (High-level MSC) находят широкое применение на практике на всех стадиях проектирования систем: от ранних стадий анализа предметной области и сбора системных

[1] Работа частично поддержана грантом РФФИ 14-07-00401.

требований до тестирования. Поэтому развитие методов и средств анализа и верификации моделей распределенных систем, описанных на языке MSC, представляет безусловный практический интерес.

Из-за отсутствия формальной семантики для многих конструкций языка MSC необходим хорошо изученный промежуточный формализм для их представления. В данной работе в качестве такого формализма рассматриваются раскрашенные сети Петри (CPN) [10], которые широко используются на практике и имеют разработанные методы и инструментальные средства проверки свойств сетевых моделей.

Цель данной работы состоит в описании метода трансляции (H)MSC-диаграмм в CPN и применении метода для формального анализа и верификации свойств таких диаграмм. Построенная в результате трансляции сеть должна иметь оптимальный размер и формат, совместимый с входным форматом известной системы анализа CPN Tools [10].

Проблема анализа сценарных спецификаций систем исследуется рядом авторов.

Обзор особенностей языка MSC дан в [8]. В работе рассматривается подмножество языка, включающее в себя HMSC-диаграммы и диаграммы MSC с сообщениями. Авторы исследуют выразительную силу MSC-диаграмм, сравнивая их с взаимодействующими конечными автоматами. Приводится обзор проблем реализуемости и верификации программных моделей, специфицированных на этом языке. Выделены подклассы MSC, для которых эти проблемы разрешимы.

В работах [1,12] описывается практическое применение MSC-диаграмм в методах и средствах для анализа, верификации и тестирования спецификаций телекоммуникационных систем, использующих модели базовых протоколов.

Обзор литературы и сравнение известных подходов по методам трансляции MSC-диаграмм в сети Петри представлен в [13]. В статье не рассмотрены подходы, где используются HMSC-диаграммы для композиции сценариев, а также не рассмотрена проблема локального выбора в конструкциях с альтернативами и концепция данных.

Работа [11] посвящена трансляции MSC-диаграмм в ординарные сети Петри. Не рассмотрены алгоритмы трансляции структурных элементов MSC со слабой последовательной композицией событий, элемент «условие» в работе не несёт конкретной семантики.

В работах, описанных далее, вместо MSC-диаграмм используются UML Sequence Diagrams (SD-диаграммы), которые являются аналогом MSC-диаграмм для стандарта моделирования UML 2.0.

В работе [15] рассматривается трансляция диаграмм UML SD в CPN. Авторы описывают правила трансляции ограниченного набора элементов диаграммы, который включает в себя сообщения и несколько видов композиции. Также используются ограничения на элементы обмена сообщениями (т.е. рассматриваются только синхронные сообщения) и интерпретацию условий в диаграмме.

Работы [7,17] описывают моделирование диаграмм UML SD при помощи высокоуровневых сетей Петри. Приводятся правила трансляции для многих конструкций стандарта. Правила трансляции для конструкций композиции диаграмм (ссылочные диаграммы, композиция ссылок) не описаны. Отметим, что дополнительные ограничения на семантику конструкций SD-диаграмм в этих работах не позволяют

успешно специфицировать и анализировать распределённые системы с независимыми компонентами.

Отметим также, что во всех перечисленных работах не рассматриваются в полной мере конструкции диаграмм, использующие динамические данные.

Таким образом, основными преимуществами нашего подхода являются широкий охват конструкций стандарта MSC, динамическая интерпретация конструкций с данными и элементов типа «условие», рассмотрение HMSC-диаграмм и ссылочных выражений в качестве средств композиции MSC, а также конструкций локального и нелокального выбора. Важная особенность нашего подхода состоит в том, что его можно применять для анализа и верификации (H)MSC-спецификаций распределенных систем, поскольку никаких существенных ограничений на рассматриваемые конструкции диаграмм не накладывается.

Основные результаты данной статьи были представлены в [5].

# 1. MSC-диаграммы

Стандарт MSC включает несколько структурных уровней представления: базовые MSC-диаграммы, HMSC-диаграммы (композиция MSC-диаграмм), документ MSC (набор диаграмм HMSC и MSC).

MSC-диаграмма представляет собой сценарий работы системы, описывающий возможную трассу или набор трасс исполнения системы. Определим *стартовые события* MSC как события, которые могут быть выполнены первыми среди всех событий данной диаграммы. По аналогии со стартовыми событиями будем определять также и *финальные события* MSC. К ним относятся последние события в упорядоченной последовательности всех событий каждого процесса MSC. *Набором возможных трасс* MSC будем называть множество последовательностей исполнения событий диаграммы, в котором каждая последовательность начинается со стартового события. Концом каждой последовательности исполнения будет являться событие, после исполнения которого в диаграмме не останется динамически допустимых событий для исполнения. Если последовательность исполнения событий содержит все финальные события диаграммы, это означает, что все события каждого процесса в MSC-диаграмме были выполнены.

Каждая MSC отображает связь между набором системных компонент (сущностей или процессов) [9]. Все события процесса упорядочены вдоль его оси, независимо от других процессов. Взаимодействие между процессами осуществляется при помощи *сообщений*, которые определяют связь между событиями этих процессов. В стандарте MSC отправка и прием сообщения являются двумя разными асинхронными событиями. Диаграмма MSC устанавливает частичное упорядочивание на множестве содержащихся в ней событий.

Кроме событий отправки и приема сообщений, к *базовым элементам* MSC относятся локальные события, условия, события создания и завершения процессов, таймеры, шлюзы обмена сообщениями (message gates) и шлюзы упорядочения (order gates), конструкции удалённого вызова методов и обобщённого упорядочения (general ordering). Помимо базовых элементов стандарт содержит структурные конструкции, которые позволяют задавать разные способы композиции для набора событий одного или нескольких процессов. Так, при помощи *встроенных выражений* MSC мож-

но описывать параллельную, альтернативную или циклическую композицию набора событий. Ссылочные выражения, декомпозиция сущностей и HMSC-диаграммы предназначены для задания разных уровней абстракции описываемой системы и позволяют выполнять синтез и декомпозицию диаграмм. Заметим, что в стандарте MSC определено, что связь всех структурных конструкций внутри диаграмм осуществляется при помощи слабой последовательной композиции.

В обновлении стандарта MSC-2000 появилась возможность использования конструкций с выражениями на языке данных. Стандарт не накладывает ограничений на нотацию данных, поэтому язык данных может быть выбран любым. Объявления данных размещаются в документе MSC. Выражения на языке данных могут быть установлены только для определенных элементов диаграммы. К таким элементам относятся сообщения, локальные действия, условия и ссылочные выражения.

В данной работе для описания объявлений и выражений с данными используется синтаксис языка SDL-2010. Предполагается, что язык данных MSC допускает использование простых типов `Boolean`, `Integer` и `String`, а также составного типа `Enumeration`. Выражения на языке данных состоят из переменных, литералов, скобок, арифметических операторов и операторов присваивания и сравнения.

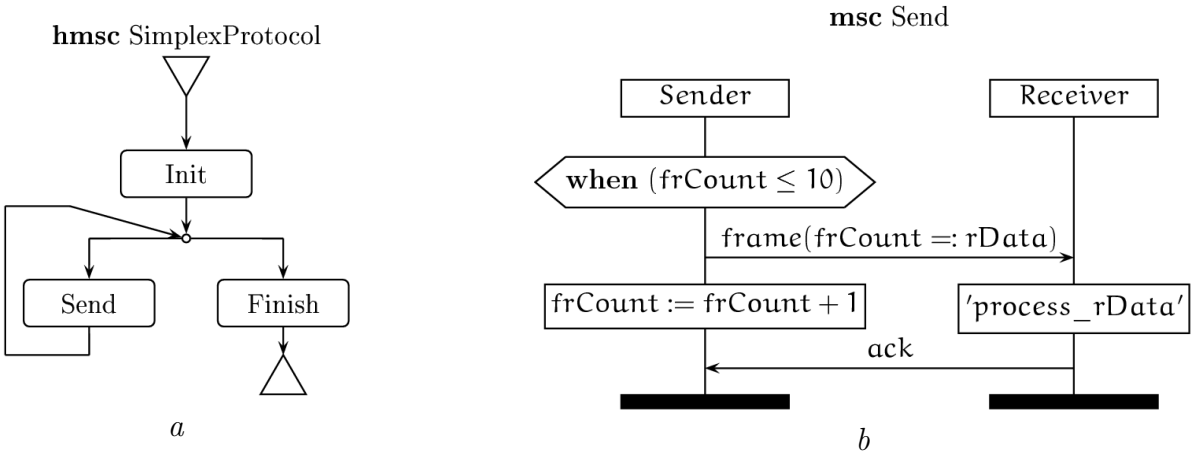


Рис. 1. *a* — HMSC *SimplexProtocol*; *b* — MSC *Send*

Отметим, что некоторые конструкции стандарта MSC не были отдельно рассмотрены в работе. Тем не менее, эти конструкции могут быть промоделированы с помощью уже рассмотренных. К таким конструкциям относятся конструкции удалённого вызова методов (могут быть промоделированы сообщениями с использованием обобщённого упорядочения), конструкции шлюзов упорядочения (моделируются с помощью конструкций обобщённого упорядочения и шлюзов обмена сообщениями), а также декомпозиция процессов MSC (см. главу 2.4.).

Пример диаграмм HMSC и MSC изображён на Рис. 1. Диаграммы описывают модифицированную версию неограниченного симплексного протокола [16]. HMSC *SimplexProtocol* содержит следующие ссылочные MSC: *Init*, *Send* и *Finish*. Диаграммы *Init* и *Finish* предназначены для описания фаз инициализации и завершения протокола. После фазы инициализации каждый процесс может исполнить события в одной из диаграмм *Send* или *Finish*. MSC *Send* (Рис. 1*b*) изображает обмен данными между процессами *Sender* и *Receiver*. *Sender* имеет переменную *frCount*, содержащую целочисленные данные для передачи. После отправки кадра с данными *Sender*

ожидает кадра подтверждения от *Receiver*. Выбор той или иной диаграммы в точке разветвления HMSC основывается на значениях охранных условий, помещённых в начало каждой MSC. Таким образом, пока начальное условие истинно (т.е. значение переменной *frCount* не превышает 10), сценарий *Send* может быть исполнен. В противном случае считается, что процесс *Sender* больше не имеет данных для передачи. В этом случае может быть исполнен только сценарий *Finish*.

# 2. Трансляция MSC-диаграмм в CPN

## 2.1. Общее описание алгоритма трансляции

Входными данными алгоритма трансляции являются диаграмма (H)MSC или документ MSC в текстовой нотации согласно стандарту. Результатом работы алгоритма является CPN в формате, совместимом с системой CPN Tools. Эта сеть будет иерархической в случае, если исходная спецификация задана при помощи HMSC-диаграммы, или если входная MSC содержала ссылочные выражения.

Алгоритм трансляции можно разбить на три основных этапа.

На 1 этапе по входной (H)MSC-диаграмме строится её внутреннее представление, которое будем называть *графом частичного порядка* (H)MSC. Граф генерируется следующим образом. Для каждого события в диаграмме создается узел в графе частичного порядка. Данный узел хранит информацию о множестве смежных узлов, идентификаторе процесса и структурной конструкции, к которой принадлежит данное событие, а также приписанные этому событию выражения на языке данных. Узлы в построенном графе соединяются ориентированными дугами на основании частичного порядка так же, как связаны события в исходной диаграмме.

На 2 этапе выполняется обработка графа частичного порядка. В процессе обработки происходят такие действия, как создание вспомогательных узлов графа (входные и выходные узлы для встроенных выражений MSC), развертка ссылок (построение графов частичного порядка для ссылочных MSC), поиск и обработка альтернатив с нелокальным выбором.

На 3 этапе обработанный граф частичного порядка транслируется в CPN. Каждому узлу графа соответствует переход в CPN. Каждой дуге, соединяющей два узла в графе частичного порядка, соответствует место и две ориентированных дуги, соединяющие два перехода в CPN. Ориентация построенных дуг сети совпадает с ориентацией дуг графа частичного порядка. Места сети, которые используются для передачи управления между событиями MSC, помечаются типом `UNIT`. Исполнению события в MSC соответствует срабатывание соответствующего перехода в результирующей CPN. Стартовым событиям MSC-диаграммы соответствуют переходы с входными местами `start` с начальной разметкой `1'()`. Финальным событиям MSC-диаграммы соответствуют переходы с выходными местами `end`, не содержащие исходящих дуг. Все типы данных и переменные, объявленные в документе MSC, преобразуются в соответствующие типы данных и переменные языка CPN ML.

Описанный алгоритм трансляции по входной (H)MSC строит CPN таким образом, что множество всех возможных трасс MSC будет совпадать с множеством всех возможных последовательностей событий (срабатываний переходов) сети, полученной в результате трансляции. Началом каждой последовательности срабатываний

переходов в результирующей CPN является переход, который соответствует стартовому событию исходной MSC. Концом последовательности срабатываний переходов является переход, соответствующий событию в MSC, после исполнения которого в диаграмме не останется динамически допустимых трасс исполнения событий.

Рассмотрим далее правила трансляции для основных элементов из стандарта MSC. Описание трансляции остальных конструкций приведено в [3].

## 2.2. Трансляция базовых элементов MSC

Каждое событие из базового набора элементов MSC моделируется одним переходом CPN. Далее рассмотрим трансляцию сообщений и предикатных условий.

Согласно стандарту MSC, событию приема сообщения предшествует событие его отправки. Каждое сообщение моделируется двумя переходами в CPN. Порядок следования переходов, соответствующих событиям отправки и приема сообщения, соблюдается при помощи места, которое связывает эти переходы и гарантирует правильную очередность между их срабатываниями.

Предикатные (охранные) условия используются для проверки в процессе исполнения диаграммы истинности предиката, заданного на языке данных MSC. Такие условия могут размещаться в начале встроенного выражения или MSC-диаграммы. Если проверочное условие ложно, то исполнение событий диаграммы, которые следуют после охранного условия, прекращается. В процессе трансляции каждое предикатное условие преобразуется в один переход сети со спусковой функцией, вычисляющей значение установленного предиката.

## 2.3. Трансляция элементов MSC с данными

Стандарт MSC выделяет статические и динамические данные. Статические данные используются в качестве параметров в ссылках и не могут изменяться внутри MSC, в которую они были переданы. Динамические данные могут изменяться процессом во время его функционирования. Далее будем рассматривать только динамические данные как более общий и распространённый случай использования данных в MSC.

Динамические данные могут содержаться в сообщениях, локальных действиях и условиях (см. Рис. 1*b*). Выражения в сообщениях и локальных действиях содержат операции присваивания. Выражения в условиях не содержат операций присваивания и представляют собой логический предикат.

Алгоритм трансляции событий с данными состоит в следующем. Для каждого события строится узел в графе частичного порядка, содержащий информацию о выражениях на языке данных и переменных. Далее построенный граф преобразуется в переход с входными и выходными совмещёнными местами (fusion places) для каждой переменной, участвующей в выражении. Совмещённое место является аналогом глобальной переменной в языках программирования. При трансляции условий каждая входная или выходная дуга соответствующего условию перехода, инцидентная месту переменной из проверочного предиката, помечается именем этой переменной, поскольку условия не изменяют значения переменных. При трансляции выражений в сообщениях и действиях все переменные, встречающиеся только в правой части присваиваний, транслируются по правилам для условий. Для переменных из левой

части присваиваний входящие в места этих переменных дуги помечаются конструкцией CPN ML `let-in-end` [10], с помощью которой происходит изменение значений переменных.

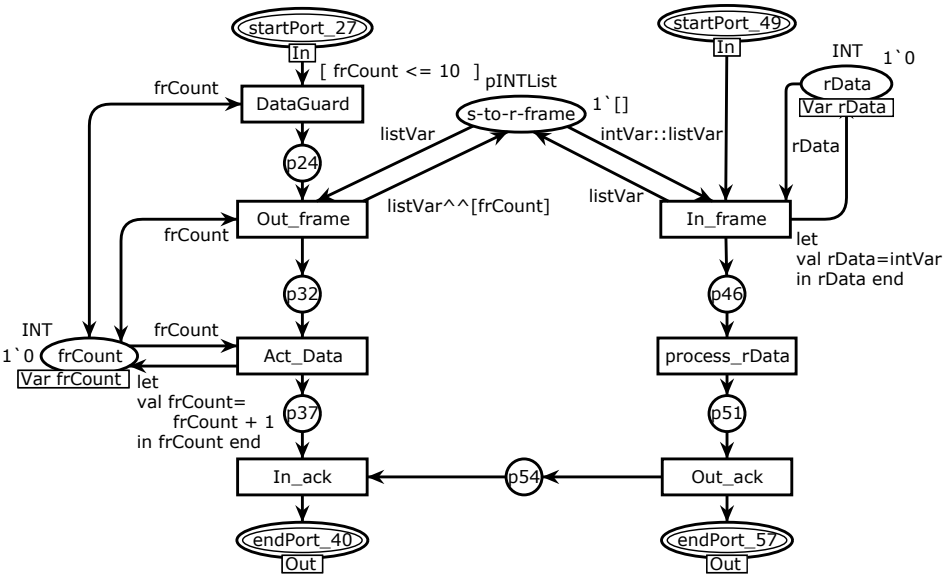


Рис. 2. CPN, полученная в результате трансляции MSC *Send* из рисунка 1*b*

Стандарт MSC подразумевает, что взаимодействующие процессы обмениваются сообщениями при помощи буфера. Модель буфера в MSC такова, что для каждого сообщения в диаграмме существует один FIFO-буфер. В результирующей CPN такие буферы моделируются при помощи мест списочного типа `list`. Этот тип используется для хранения очереди записей с типом `product`, где каждая запись содержит набор передаваемых данных.

Для каждого события отправки сообщения с данными в строящейся CPN создается переход с выходным местом-очередью, предназначенном для хранения переменных с отправленными данными. Это место является входным для перехода, соответствующего событию приема сообщения. Таким образом, исполнение события отправки сообщения с данными моделируется в CPN срабатыванием соответствующего перехода и помещением записи с передаваемыми данными в место-очередь. При приёме сообщения происходит извлечение первой фишки с данными из места-очереди и сохранение полученных данных в переменных процесса-получателя.

Пример сети, полученной в результате трансляции MSC-диаграммы *Send*, содержащей конструкции с выражениями на языке данных, приведён на Рис. 2.

## 2.4. Трансляция структурных элементов

**Встроенные выражения MSC.** Рассмотрим далее встроенное выражение альтернативного выбора (выражение `alt`).

Встроенное выражение `alt`, заданное для нескольких операндов (ветвей), позволяет выполнить только один из операндов, исключая выполнение других. Основной проблемой при трансляции выражений `alt` в (H)MSC является *нелокальный выбор*, при котором процессы могут независимо выбирать разные альтернативные

ветви исполнения в одной альтернативе [4, 14]. MSC-диаграммы могут оставаться синтаксически корректными, даже если будут содержать выражения `alt`, не обладающие свойством локального выбора. Отсутствие данного свойства может приводить к неоднозначной трактовке MSC-диаграммы, а также к незапланированному поведению и возможным тупикам в результирующей CPN.

Стандарт MSC [9] не определяет конкретной семантики для выражений `alt` с нелокальным выбором в случае, если (H)MSC-диаграммы с такими конструкциями рассматриваются в качестве формальных исполняемых моделей. В нашей работе проблема локального выбора разрешается за счёт доопределения семантики выражения `alt` следующим образом. Для каждой альтернативной ветви исполнения в результирующей CPN создается синхронизирующий переход, общий для всех процессов, участвующих в выборе. Срабатывание синхронизирующего перехода определяет выбор конкретной альтернативной ветви.

**Ссылочные выражения MSC.** Ссылки на MSC используются для описания вложенности MSC-диаграмм внутри HMSC или MSC. Ссылка может содержать как имя MSC-диаграммы, на которую она ссылается, так и ссылочное выражение, состоящее из комбинации имён диаграмм и операторов композиции.

Ссылочное выражение (H)MSC в процессе трансляции преобразуется в отдельную подсеть (модуль) CPN, что позволяет сохранить первоначальную иерархическую структуру входной диаграммы. Таким образом, ссылочному выражению соответствует один подстановочный переход (substitution transition) CPN. Подстановочный переход представляет собой абстракцию поведения системы, детализированную в отдельной подсети. После построения сети, моделирующей корневую (H)MSC-диаграмму, выполняется рекурсивное построение всех подсетей, соответствующих ссылочным MSC, по общему алгоритму из раздела 2.1.

**Диаграммы HMSC.** HMSC-диаграмма является выразительным способом задания отношений между различными MSC. Графически HMSC изображается в виде ориентированного графа, в котором каждый из узлов может быть начальным или конечным узлом, ссылочным выражением MSC или connect-узлом. HMSC может содержать только один начальный узел (см. Рис. 1*a*).

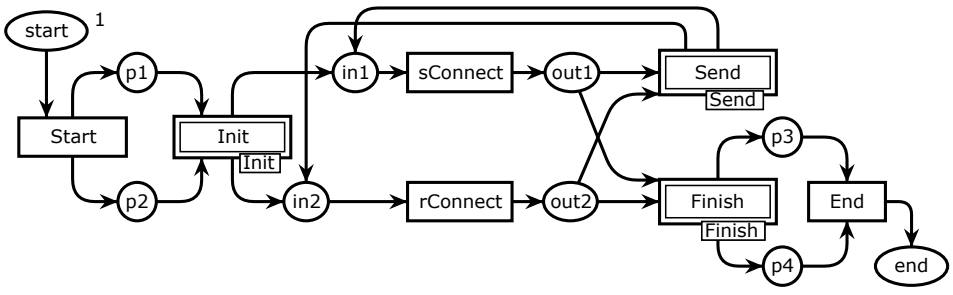


Рис. 3. CPN, полученная в результате трансляции HMSC *SimplexProtocol*

При описании алгоритма трансляции далее будем считать, что число процессов во всех ссылочных MSC-диаграммах внутри HMSC совпадает. Также считаем, что в HMSC отсутствуют connect-узлы (каждый connect-узел можно заменить ссылочной MSC). На 1 этапе трансляции по входной HMSC строится граф частичного порядка тем же способом, что и для MSC-диаграмм. Далее выполняется построение

графа частичного порядка для каждой ссылочной MSC. На заключительном этапе трансляции каждый узел графа, соответствующий ссылке на MSC, переводится в подстановочный переход по правилам для ссылочных выражений, описанным выше.

На Рис. 3 представлен пример сети, полученной в результате трансляции HMSC-диаграммы *SimplexProtocol*, изображённой на Рис. 1*a*.

**Декомпозиция процессов.** Для каждого процесса в MSC-диаграмме может быть определено его внутреннее поведение, детализированное в отдельном документе MSC. Детализированные процессы внутри MSC-диаграммы помечаются ключевым словом `decomposed`, за которым следует имя диаграммы, описывающей поведение данного процесса. Отметим, что один процесс MSC на более низком уровне абстракции может быть детализирован несколькими процессами MSC, каждый из которых также может быть детализирован.

Конструкция декомпозиции процессов может быть промоделирована при помощи встроенных выражений MSC с использованием алгоритма трансформации [9].

### 2.5. Оценка размера результирующей CPN

Пусть дана MSC-диаграмма с числом событий $N$, процессов $P$ и сообщений $M$. Введем следующие обозначения:

$S$ — число стартовых и финальных событий диаграммы;

$I_p$ и $I_l$ — число конструкций параллельного исполнения (`par`) и циклов (`loop`);

$VAR$ — число переменных, используемых в диаграмме.

Тогда верхняя оценка $T_{MSC}$ числа переходов сети, полученной в результате трансляции данной MSC, будет иметь вид: $T_{MSC} \leq N + 2P(I_p + I_l)$. Верхняя оценка $P_{MSC}$ числа мест результирующей CPN имеет вид: $P_{MSC} \leq N + M + S + VAR + 2P(I_p + I_l)$.

Таким образом, получаем линейную оценку числа мест и переходов результирующей CPN в случае, если в MSC не используются конструкции `par` и `loop`.

Рассмотрим MSC с $N$ событиями, содержащую детализированный процесс $i$. Пусть детализированное поведение процесса $i$ описывается в документе MSC с количеством событий $N_i$. Тогда верхняя оценка $T_d$ числа переходов результирующей CPN имеет вид: $T_d \leq T_{MSC} + T^i_{MSC} - R_i$, где $T_{MSC}$ – верхняя оценка числа переходов сети транслированной диаграммы, содержащей декомпозицию, $T^i_{MSC}$ – верхняя оценка числа переходов сети транслированного MSC документа, описывающего детализацию процесса $i$, $R_i$ – число ссылочных выражений, которые содержит процесс $i$. Аналогично определяется верхняя оценка $P_d$ числа мест: $P_d \leq P_{MSC} + P^i_{MSC} - PR_i$.

Рассмотрим HMSC-диаграмму с общим количеством начальных и конечных узлов $N_0$ и ссылочными узлами в количестве $N_r$. Тогда верхняя оценка числа переходов $T_{HMSC}$ и числа мест $P_{HMSC}$ сети Петри, полученной в результате трансляции HMSC, имеет вид: $T_{HMSC} \leq N_0 + N_r$, $P_{HMSC} \leq N_0 + PN_r$.

## 3. Верификация MSC-диаграмм

### 3.1. Метод верификации

Для верификации CPN с помощью системы SPIN был разработан метод [2], в котором используются следующие ограничения на входные сети. Допускаются базовые

типы `int`, `bool` и `unit`, а также перечисления, кортежи, записи и списки. Ёмкость мест (количество фишек в местах) и длина списков в верифицируемой CPN должна быть ограничена известной константой.

CPN, полученная в результате работы описанного выше алгоритма трансляции (H)MSC, удовлетворяет требованиям на допустимые типы данных.

Определим подклассы (H)MSC-диаграмм, для которых ограничения на ёмкость мест и длину списков CPN будут выполняться.

Отметим, что накопление фишек в местах и увеличение размера списков в сети может происходить при срабатывании переходов, соответствующих событиям отправки сообщений, помещённым в конструкцию цикла.

Диаграмма (H)MSC называется *регулярной* [9], если для каждого цикла внутри диаграммы выполнены следующие условия. Все сообщения, отправленные от процесса $i_1$ к процессу $i_2$, должны быть подтверждены обратным сообщением к процессу $i_1$. Если цикл содержит два локальных действия, принадлежащих разным процессам, то между этими процессами должен существовать обмен сообщениями.

В регулярной (H)MSC-диаграмме каждый процесс, отправив сообщение внутри цикла, не сможет приступить к следующей итерации отправки этого сообщения, пока не получит подтверждения от получателя на предыдущее сообщение. Это означает, что буфер каждого сообщения внутри цикла будет хранить не более одного элемента. Следовательно, в CPN, полученной в результате трансляции регулярной (H)MSC-диаграммы, ёмкость мест и длина списков будет ограничена числом 1.

Рассмотрим нерегулярную MSC-диаграмму, которая содержит только циклы с ограничением на число итераций, т.е. в диаграмме используются циклические конструкции `loop<n,m>` с заданным числом итераций, или циклы с ограничивающими число итераций охранными условиями. Тогда верхняя оценка ёмкости мест и длины списков в результирующей CPN будет равна наибольшему числу итераций среди всех циклов (для вложенных циклов число итераций внешнего цикла равно произведению числа итераций этого цикла и всех его подциклов).

В случае, если MSC-диаграмма не регулярна и содержит циклы с неограниченным числом итераций, ёмкость мест и длина списков в результирующей CPN может быть также неограниченной. Однако, если для каждого неограниченного цикла установить охранное условие, которое останется истинным только до определённого числа итераций, то требуемые ограничения будут удовлетворены. Верхняя оценка ёмкости мест и длины списков при трансляции такой диаграммы будет определяться по аналогии с предыдущим случаем.

Такие (H)MSC-диаграммы, которые не являются регулярными, но содержат только ограниченные циклы, будем называть *квазирегулярными*.

## 3.2. Верификация протокола PAR

Рассмотрим пример верификации свойств модели протокола PAR, описанного в [16]. Для моделирования таймера в протоколе используются специальные конструкции MSC, отвечающие за установку таймера, таймаут и сброс таймера. В результирующей CPN исполнение события таймера моделируется срабатыванием соответствующего этому событию перехода.

Была составлена HMSC-диаграмма, описывающая возможные сценарии работы

протокола, такие как инициализация протокола, потеря кадра в процессе передачи его от отправителя к получателю и обратно, успешная доставка и подтверждение кадра, завершение передачи кадров и проверка корректности переданных данных.

Описанная спецификация с помощью алгоритмов трансляции была преобразована в CPN. Чтобы уменьшить пространство состояний полученной CPN и применить верификатор на основе SPIN, исходная спецификация была преобразована в квазирегулярную форму [6]. Для этого было введено дополнительное ограничение на модель протокола: количество потерь кадров при передаче ограничено числом 10.

Для построенной сети были сформулированы следующие свойства:

1) отсутствие тупиков;
2) все отправленные данные должны быть получены;
3) полученная последовательность данных совпадает с отправленной последовательностью данных.

Свойство 1 означает, что результирующая CPN не должна иметь тупиковых разметок, за исключением разметки, соответствующей единственной фишке в конечной точке исполнения HMSC. Свойство 2 определяется при помощи LTL формулы, приведённой в [6]. Свойство 3 представляет собой постусловие, которое может быть сформулировано следующим образом: в конечном состоянии модели CPN, где нет доступных для исполнения переходов, место, соответствующее конечной точке HMSC, должно содержать единственную фишку со значением `1'()`.

Анализ свойств сетей проводился в CPN Tools (свойство 1) и системе автоматической верификации на основе SPIN [2] (свойства 2 и 3). В результате анализа перечисленных свойств получили, что данные свойства оказались выполненными для модели протокола PAR. Детали верификации HMSC-спецификации протокола PAR с исходными файлами примера можно найти в [6].

# 4. Заключение

Диаграммы (H)MSC являются простым и выразительным средством спецификации программных систем широкого профиля. Наличие структурных конструкций и языка данных в (H)MSC позволяет использовать их в качестве полных спецификаций. Поэтому актуальны проблемы анализа и верификации моделей систем, описанных на языке MSC. В настоящей работе описан метод трансляции (H)MSC-диаграмм распределенных систем в CPN. Алгоритмы трансляции охватывают большинство конструкций из основного стандарта (включая концепцию данных), кроме расширения с временем [9]. На основе представленных алгоритмов трансляции реализован транслятор, работа которого была протестирована на различных примерах с выполнением анализа и верификации некоторых свойств результирующих сетей с помощью системы CPN Tools и системы верификации CPN на основе SPIN.

В планы дальнейшей работы входит рассмотрение временных расширений стандарта MSC и исследование прочих диалектов языка MSC, подходящих для спецификации сложных распределенных систем. Также планируется применение транслятора на других практически значимых примерах (коммуникационные протоколы, Новосибирская олимпиадная система тестирования NSUts).

# Список литературы


1. *Анureев И.С., Баранов С.Н., Белоглазов Д.М., Бодин Е.В., Дробинцев П.Д., Колчин А.В., Котляров В.П., Летичевский А.А., Летичевский А.А. мл., Непомнящий В.А., Никифоров И.В., Потиенко С.В., Прийма Л.В., Тютин Б.В.* Средства поддержки интегрированной технологии для анализа и верификации спецификаций телекоммуникационных приложений // СПИИРАН. 2013. Вып. 3(26). С. 349–383 (English transl.: *Anureev I., Baranov S., Beloglazov D., Bodin E., Drobintsev P., Kolchin A., Kotlyarov V., Letichevsky A., Letichevsky A. Jr., Nepomniaschy V., Nikiforov I., Potienko S., Pryma L., Tyutin B.* Tools for Supporting Integrated Technology of Analysis and Verification of Specifications for Telecommunication Applications // SPIIRAS. Proc. 2013. V. 3. P. 349–383.)

2. *Стененко А.А., Непомнящий В.А.* Верификация раскрашенных сетей Петри методом проверки моделей / Институт систем информатики СО РАН. Препринт 172. Новосибирск, 2014. `http://www.iis.nsk.su/files/preprints/172.pdf` (English transl.: *Stenenko A.A., Nepomniaschy V.A.* Verification of Coloured Petri Nets by model checking method / Institute of Informatics Systems SB RAS. Preprint 172. Novosibirsk, 2014.)

3. *Черненок С.А., Непомнящий В.А.* Анализ MSC-диаграмм распределенных систем с помощью раскрашенных сетей Петри / Институт систем информатики СО РАН. Препринт 171. Новосибирск, 2013. `http://www.iis.nsk.su/files/preprints/171.pdf` (English transl.: *Chernenok S.A., Nepomniaschy V.A.* Analysis of Message Sequence Charts of Distributed Systems Using Coloured Petri Nets / Institute of Informatics Systems SB RAS. Preprint 171. Novosibirsk, 2013.)

4. *Abdallah R., Gotlieb A., Helouet L., Jard C.* Scenario Realizability with Constraint Optimization // FASE 2013, LNCS 7793. 2013. P. 194–209.

5. *Chernenok S.A., Nepomniaschy V.A.* Analysis and Verification of Message Sequence Charts of Distributed Systems Using Coloured Petri Nets // Proc. of 5th Workshop "PSSV: Theory and Applications". Moscow, 2014. P. 38–49.

6. *Chernenok S.A.* Analysis and Verification of Message Sequence Charts of Distributed Systems Using Coloured Petri Nets. Appendix. `http://bitbucket.org/chernenok/msc-verification`

7. *Eichner C., Fleischhack H., Meyer R., Schrimpf U., Stehno S.* Compositional Semantics for UML 2.0 Sequence Diagrams Using Petri Nets // SDL-Forum 2005, LNCS 3530. P. 133–148.

8. *Genest B., Muscholl A., Peled D.* Message Sequence Charts // Lectures on Concurrency and Petri Nets, LNCS 3098. 2003. P. 537–558.

9. ITU-T Recommendation Z.120 (02/2011): Message Sequence Chart (MSC). 2011.

10. *Jensen K., Kristensen L.M.* Coloured Petri Nets: Modeling and Validation of Concurrent Systems. Springer, 2009.

11. *Kryvyi S., Matvyeyeva L.* Algorithm of Translation of MSC-specified System into Petri Net // Fundamenta Informaticae, 2007. V. 79. № 3–4. P. 431–445.

12. *Letichevsky A., Kapitonova J., Letichevsky A. Jr., Volkov V., Baranov S., Weigert T.* Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications

// Computer Networks: The International Journal of Computer and Telecommunications Networking. 2005. V. 49. № 5. P. 661–675.

13. *Liang H., Dingel J., Diskin Z.* A Comparative Survey of Scenario-Based to State-Based Model Synthesis Approaches // SCESM 2006, ACM. NY, 2006. P. 5–12.

14. *Muccini H.* Detecting Implied Scenarios Analyzing Non-local Branching Choices // FASE 2013, LNCS 2621. P. 372–386.

15. *Ribeiro O.R., Fernandes J.M.* Some Rules to Transform Sequence Diagrams into Coloured Petri Nets // 7th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools (CPN 2006). Aarhus, Denmark, 2006. P. 237–256.

16. *Tanenbaum A.* Computers Networks (4th Edition). Prentice Hall PTR, 2002.

17. *Yang N., Yu H., Sun H., Qian Z.* Modeling UML sequence diagrams using extended Petri nets // Telecommunication Systems. Springer, 2012. V. 51. № 2–3. P. 147–158.

## Analysis and Verification of Message Sequence Charts of Distributed Systems with the Help of Coloured Petri Nets

Chernenok S. A., Nepomniaschy V. A.

*A.P. Ershov Institute of Informatics Systems,*
*Acad. Lavrentjev pr., 6, Novosibirsk 630090, Russia*

**Keywords:** specification, translation, verification, distributed systems, communication protocols, MSC diagrams, Coloured Petri Nets

The standard language of message sequence charts MSC is intended to describe scenarios of object interaction. Due to their expressiveness and simplicity MSC diagrams are widely used in practice at all stages of system design and development. In particular, the MSC language is used for describing communication behavior in distributed systems and communication protocols. In this paper the method for analysis and verification of MSC and HMSC diagrams is considered. The method is based on the translation of (H)MSC into coloured Petri nets. The translation algorithms cover most standard elements of the MSC including data concepts. Size estimates of the CPN which is the result of the translation are given. Properties of the resulting CPN are analyzed and verified by using the known system CPN Tools and the CPN verifier based on the known tool SPIN. The translation method has been demonstrated by the example.

**Сведения об авторах:**
**Черненок Сергей Анатольевич**,
Институт систем информатики имени А. П. Ершова СО РАН,
аспирант
**Непомнящий Валерий Александрович**,
Институт систем информатики имени А. П. Ершова СО РАН,
кандидат физ.-мат. наук, зав. лабораторией