

Поиск некорректной начальной маркировки иерархической сети Петри на основе диаграмм состояний UML

12, декабрь 2014

Пашенкова А. В., Рудаков И. В.

УДК: 519.6

Россия, МГТУ им. Н.Э. Баумана

anna_vp@inbox.ru

Введение

В современной индустрии программного обеспечения (ПО) широко используется объектно-ориентированный подход к анализу и проектированию ПО с применением языка UML. Некорректное представление сложных алгоритмов и механизмов синхронизации на UML-диаграммах приводит к взаимным блокировкам потоков и другим проблемам при функционировании ПО. Поэтому особый интерес для разработчиков ПО представляют методы преобразования и анализа правильности UML-диаграмм.

В языке UML и CASE средствах на его основе нет собственных средств для обоснования правильности диаграмм, поэтому большое внимание уделяется методам преобразования UML-диаграмм в сеть Петри [1]. В рассмотренных работах [2, 3, 4] не учитывается специфика разработки программ. Предлагаются различные расширения сетей Петри, но при этом недостаточно внимания уделено иерархическим сетям Петри, в то время как одним из известных методов борьбы со сложностью систем является представление их с помощью иерархических структур. Анализ сети Петри производится с использованием пакетов моделирования, не включающих автоматическое преобразование UML-диаграмм в сеть Петри. Исходные UML-диаграммы преобразуются в сеть Петри вручную. Начальное состояние сети Петри заполняется на основании уже известных наборов входных данных, не учитывается устойчивость диаграмм к различным входным данным. Все это затрудняет использование предложенных методов. Таким образом, разработка метода поиска некорректных маркировок иерархической сети Петри, построенной на основе диаграмм состояний UML, является актуальной и научно значимой задачей.

1. Проверка корректности UML-диаграмм с применением сетей Петри

На этапе проектирования возникает необходимость проведения исследования функционирования и свойств проектируемого программного обеспечения, результаты которого

могут повлиять на архитектуру многопоточных приложений, механизм взаимодействия потоков при использовании разделяемых ресурсов.

Некорректные или несогласованные требования и ограничения к различным UML диаграммам могут привести к возникновению взаимных блокировок ресурсов и других проблем функционирования многопоточного ПО. Использование сетей Петри в качестве формального подхода к анализу позволяет устранить ошибки и несогласованность требований к разрабатываемому ПО.

Разработку ПО на основе использования UML диаграмм и сетей Петри можно разделить на несколько составных частей: разработка набора UML диаграмм, преобразование полученных диаграмм в сети Петри, анализ сетей Петри и внесение изменений в UML диаграммы в соответствии с результатом анализа.

Моделирование проектируемой системы на основе аппарата сетей Петри благодаря развитым инструментам анализа и мониторинга позволяет разработчикам эффективно обнаруживать ошибки уже до начала стадии реализации. Для этого исходная поведенческая диаграмма (как правило, это диаграмма состояний или диаграмма деятельности) согласно набору правил преобразуется в сеть Петри, позволяющую осуществить имитационное моделирование. Если при работе сети возникли ошибки (например, тупиковые ветви), значит, при проектировании архитектуры тоже были допущены ошибки, и результаты моделирования сети Петри позволяют их локализовать. В условиях все возрастающей сложности разрабатываемого программного обеспечения подобный подход призван значительно повысить его качество.

Наличие и отсутствие в модели определенного свойства могут носить для системы как положительный, так и отрицательный характер. Далее рассмотрены свойства классической сети Петри применительно к моделированию и анализу программного обеспечения [1].

1. Отсутствие тупиковых или «мертвых» маркировок. Маркировка, при которой ни один переход в сети не разрешен, называется тупиковой. В тупиковой маркировке сеть прекращает свое функционирование. Если множество достижимости не содержит ни одной тупиковой маркировки, то сеть функционирует бесконечно долго. Ошибки проектирования, приводящие к тупиковой маркировке при моделировании сетью Петри, могут никак не проявлять себя при тестировании программного обеспечения и даже в условиях реальной эксплуатации до наступления определенных условий (возникновения причинно-следственных связей). Кроме того, подобные ошибки трудно выявить при анализе кода или модели, представленной в виде совокупности UML-диаграмм.
2. Свойство ограниченность сети Петри. При разработке систем необходимо моделировать запросы, распределения и освобождения ресурсов и, следовательно, необходимо соответствие количества маркеров определенному числу. При разработке реальных систем очень важно свойство ограниченности, так как в реальных процес-

сах используемые ресурсы всегда ограничены. Если позиция является неограниченной, то при функционировании сети Петри в ней происходит неограниченное накопление маркеров, что является неприемлемым для модели реальной системы. При моделировании реального многопоточного программного обеспечения с ограниченными ресурсами должны быть обязательно проверены границы мест, моделирующих семафоры, критические секции и стеки.

3. Свойство живости сети Петри. Свойство живости характеризует функциональность всех переходов в сети. То есть за время работы все переходы срабатывают хотя бы один раз. Если анализ выявит, что существуют <<мертвые>> переходы, то необходимо перестроение модели и исключение не функционирующих переходов. В данной работе осуществляется проверка построенной сети Петри на наличие тупиковых маркировок.

2. Функциональная диаграмма метода

На Рис. 1 представлена диаграмма метода поиска некорректных начальных маркировок иерархической сети Петри, построенной на основе диаграммы состояний или диаграммы деятельности UML.

На вход метода подается диаграмма состояний или диаграмма деятельности в формате XML. С использованием правил преобразования [5] на основе диаграммы строится иерархическая сеть Петри.

Для снижения временной и емкостной сложности алгоритма анализа сети Петри может быть проведена редукция [5] построенной по диаграмме UML сети Петри.

Сеть Петри, построенная для диаграммы деятельности UML, моделирует поведение алгоритма с помощью потока маркеров, который движется по сети узлов и ребер согласно определенным правилам.

Диаграммы состояний и деятельности дают неполную информацию для преобразования их в сеть Петри. На основе диаграмм UML можно получить пустую сеть Петри с нужной структурой. Начальное состояние (маркировку) сети из диаграммы получить невозможно. Поэтому следующим шагом метода является вычисление множества начальных маркировок сети Петри. Впоследствии сеть Петри анализируется для каждого начального состояния (маркировки сети) из набора.

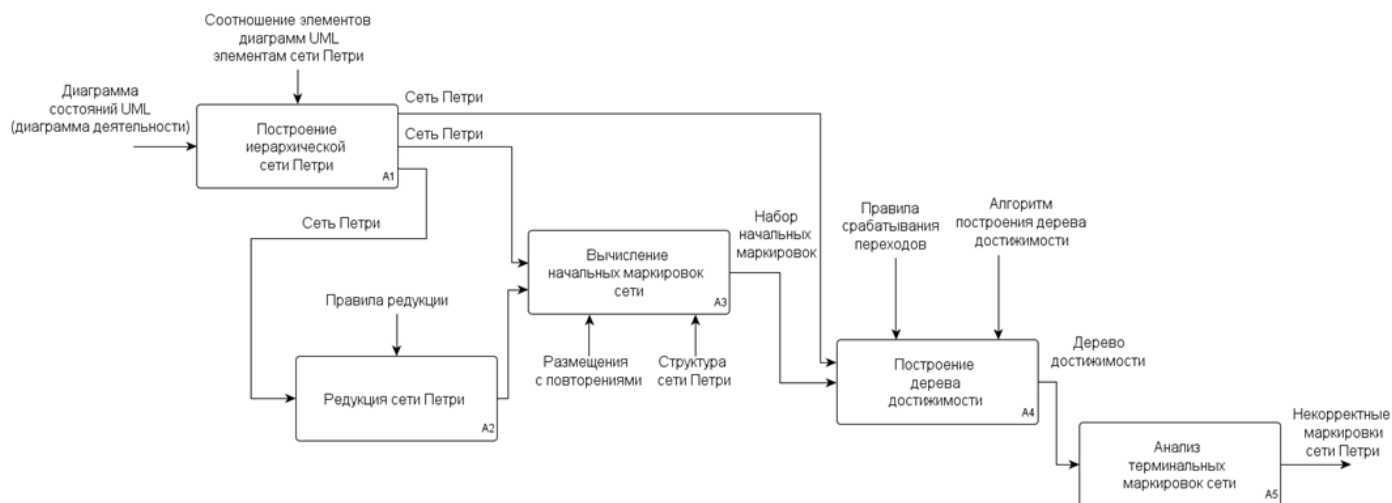


Рис. 1. Функциональная диаграмма метода

3. Поиск некорректной маркировки сети Петри

Под некорректной маркировкой сети Петри в работе понимается такое начальное состояние сети Петри, которое при определенной последовательности срабатываний переходов может привести к возникновению тупиковой ситуации.

Существует несколько подходов к анализу сетей Петри. Одним из известных методов, направленных на анализ свойств сети, является дерево достижимости. Дерево достижимости – это дерево, вершинам которого соответствуют возможные маркировки, а дугам – переходы сети. Корневая вершина представляет первоначальную маркировку. Из каждой вершины исходят дуги, соответствующие разрешенным переходам. Всякий путь в дереве, начинающийся в корне, соответствует допустимой последовательности переходов. Сеть Петри может иметь бесконечное дерево достижимости. Для получения дерева, которое можно считать полезным инструментом анализа, необходимо найти средства ограничения его до конечного размера.

Необходимо заметить, что если какое-то представление бесконечного множества конечно, то бесконечное множество маркировок должно отображаться в такое представление. В общем случае это приведет к потере информации и, возможно, к тому, что некоторые свойства сетей Петри определить будет нельзя, но все зависит от того, как представление было получено.

Чтобы сделать бесконечное дерево конечным, необходимо найти те средства, которые ограничивают введение новых маркировок (называемых граничными вершинами) на каждом шаге.

Особенностью алгоритма построения конечного дерева достижимости является специальная классификация маркировок. Каждая вершина дерева классифицируется как граничная, терминальная, дублирующая или внутренняя вершина. Маркировки, в которых нет разрешенных переходов, являются терминальными вершинами дерева достижимости. Другой класс маркировок -- это маркировки, ранее встречавшиеся в дереве. Такие дубли-

рующие маркировки называются дублирующими вершинами. Никакие последующие маркировки рассматривать не нужно, так как все они будут порождены из места первого появления дублирующей маркировки в дереве.

Для сведения дерева достижимости к конечному представлению используется еще одно средство. Для позиций, которые увеличивают число фишек некоторой последовательностью запусков переходов, можно создать произвольно большое число фишек, просто повторяя данную последовательность столько раз, сколько это нужно. Бесконечное число маркировок, получающихся из циклов такого типа, представляется с помощью специального символа w , который обозначает «бесконечность». Таким образом, в маркировке число фишек может быть либо неотрицательным целым, либо w .

Первым шагом алгоритма определяется начальная маркировка корнем дерева, т.е. граничной вершиной. Все последующие шаги направлены на классификацию граничных вершин дерева. До тех пор, пока имеются граничные вершины, они обрабатываются алгоритмом. В противном случае алгоритм останавливается. Схема алгоритма представлена на Рис. 2.

Пусть x – граничная вершина, которую необходимо обработать.

1. Если в дереве имеется другая вершина y , не являющаяся граничной, и с ней связана та же маркировка, $\mu[x] = \mu[y]$, то вершина x – дублирующая.
2. Если для маркировки $\mu[x]$ ни один из переходов не разрешен (то есть $\delta(\mu[x], t_j)$ не определено для всех $t_j \in T$), то x – терминальная вершина.
3. Для всякого перехода $t_j \in T$, разрешенного в $\mu[x]$ (то есть $\delta(\mu[x], t_j)$ определено), создать новую вершину z дерева достижимости. Маркировка $\mu[z]$, связанная с этой вершиной, определяется для каждой позиции p_i следующим образом:
 - Если $\mu[x]_i = w$, то $\mu[z]_i = w$.
 - Если на пути от корневой вершины к x существует вершина y с $\mu[y] < \delta(\mu[x], t_j)$ и $\mu[y]_i < \delta(\mu[x], t_j)_i$, то $\mu[z]_i = w$.
 - В противном случае $\mu[z]_i = \delta(\mu[x], t_j)_i$.

Дуга, помеченная t_j , направлена от вершины x к вершине z . Вершина x переопределяется как внутренняя, вершина z становится граничной. Когда все вершины дерева – терминальные, дублирующие или внутренние, алгоритм останавливается (см. Рис. 1).

Тупиковая ситуация в терминах сетей Петри подразумевает наличие тупиковой маркировки, при которой ни один из переходов не является разрешенным. Фактическое тупиковое состояние сети Петри не всегда означает тупиковую ситуацию в программе.

По завершении моделирования работы сети, представляющей некоторую процедуру алгоритма, сеть переходит в состояние, когда ни один из переходов не может сработать. В терминах сети Петри это означает тупиковую ситуацию. Необходимо различать корректное завершение работы сети от тупиковых ситуаций. Для этого в позицию сети Петри включается флаг, который определяет, завершающая эта позиция или нет.

Таким образом, если терминальная вершина дерева достижимости содержит фишки только в завершенных позициях, такая маркировка не является тупиковой, а начальная маркировка сети является корректной. В противном случае маркировка сети Петри означает тупиковую ситуацию, а начальная маркировка сети является некорректной.

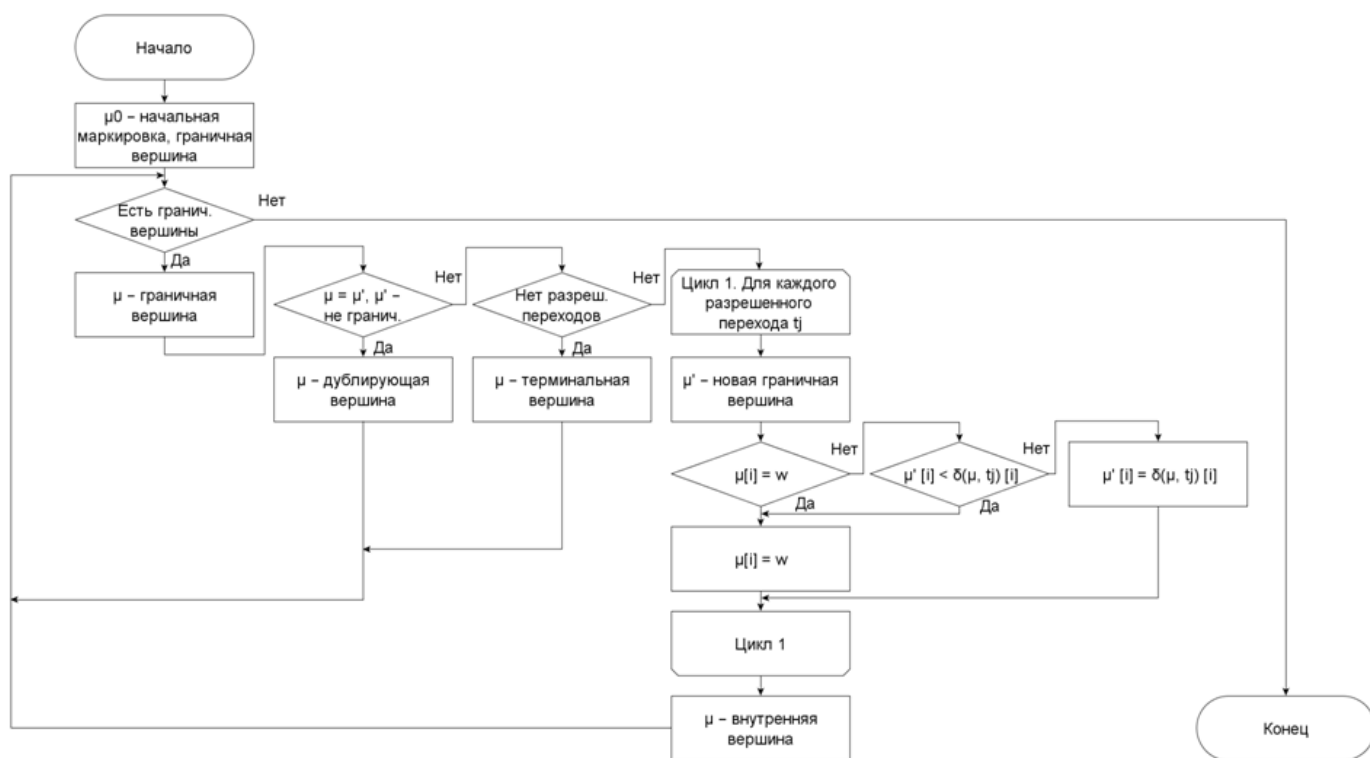


Рис. 2. Алгоритм построения дерева достижимости

4. Построение множества начальных маркировок сети

Диаграммы состояний и деятельности дают неполную информацию для преобразования их в сеть Петри. На основе диаграмм UML можно получить пустую сеть Петри с нужной структурой. Начальное состояние (маркировку) сети из диаграммы получить не-

возможно. Поэтому следующим шагом метода является вычисление множества начальных маркировок сети Петри.

Одним из вариантов заполнения начальной маркировки сети может быть полный перебор. Сложность такого метода зависит от количества всех возможных начальных состояний сети. Если пространство решений очень велико, то полный перебор может не дать результатов в течение долгого времени (недель, месяцев).

Другим методом заполнения начальной маркировки сети может быть применение эвристических алгоритмов. Эвристические алгоритмы широко применяются для решения задач высокой вычислительной сложности (задачи, принадлежащие классу NP), то есть вместо полного перебора вариантов, занимающего существенное время, а иногда технически невозможного, применяется значительно более быстрый, но недостаточно обоснованный теоретически, алгоритм.

Для вычисления пространства начальных состояний сети Петри можно применить итеративное моделирование. Идея итеративного моделирования заключается в том, что на каждом шаге итерации предлагается вариант модели, который подвергается обсуждению, рецензированию и редактированию.

Выполнением сети Петри управляют количество и распределение фишек в сети. Фишки находятся в позициях и управляют выполнением переходов сети. Сеть Петри выполняется посредством запусков переходов. Переход запускается удалением фишек из его входных позиций и образованием новых фишек, помещаемых в его выходные позиции.

Переход может запускаться только в том случае, когда он разрешен. Переход называется разрешенным, если каждая из его входных позиций имеет число фишек по крайней мере равное числу дуг из позиции в переход. Таким образом, для запуска перехода необходимо заполнить необходимым количеством фишек только те позиции, которые непосредственно связаны выходными дугами с переходом.

Суть итеративного моделирования и правила срабатывания сети Петри позволяют сформулировать алгоритм для нахождения пространства возможных начальных состояний сети. Под некорректным начальным состоянием сети понимается в данном случае такая маркировка, которая может привести к тупиковым ситуациям.

Для каждого перехода определен свой набор позиций, которые заполняются минимально необходимым количеством фишек. Количество фишек в каждой позиции равно числу выходных дуг в данный переход. Схема алгоритма представлена на Рис. 3.

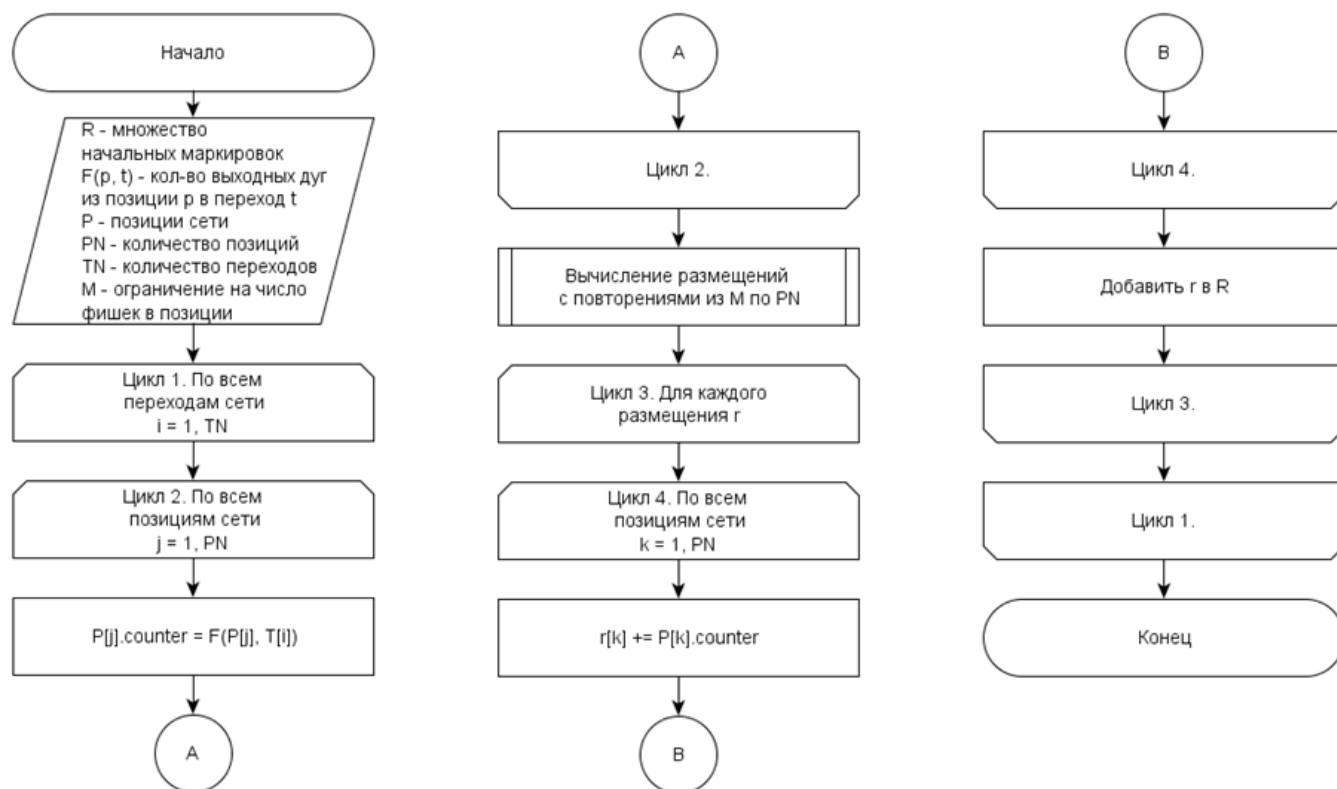


Рис. 3. Схема алгоритма вычисления множества начальных состояний сети Петри

5. Исследование алгоритма сеанса связи центра дистанционного управления

В данной главе исследуется работа метода на алгоритме сеанса связи центра дистанционного управления и контроля. Система, представляющая центр дистанционного управления и контроля, предназначена для контроля работы интеллектуальных электронных устройств и передачи им команд управления. Оборудование может быть подключено напрямую к любому порту компьютера или по сети с помощью устройства передачи данных. Диаграмма деятельности алгоритма сеанса связи взята из диссертации С.В. Коротикова [1] и представлена на

Рис. 5.

Соотношение размера сети Петри и размера UML-диаграммы отражено в Таблица 1.

Таблица 1. Соотношение размера сети Петри и размера UML-диаграммы

Количество узлов диаграммы	Количество узлов сети Петри	Количество узлов редуцированной сети Петри
13	14	8

Зависимость времени анализа сети Петри от размера пространства состояний сети представлена на Рис. 4.

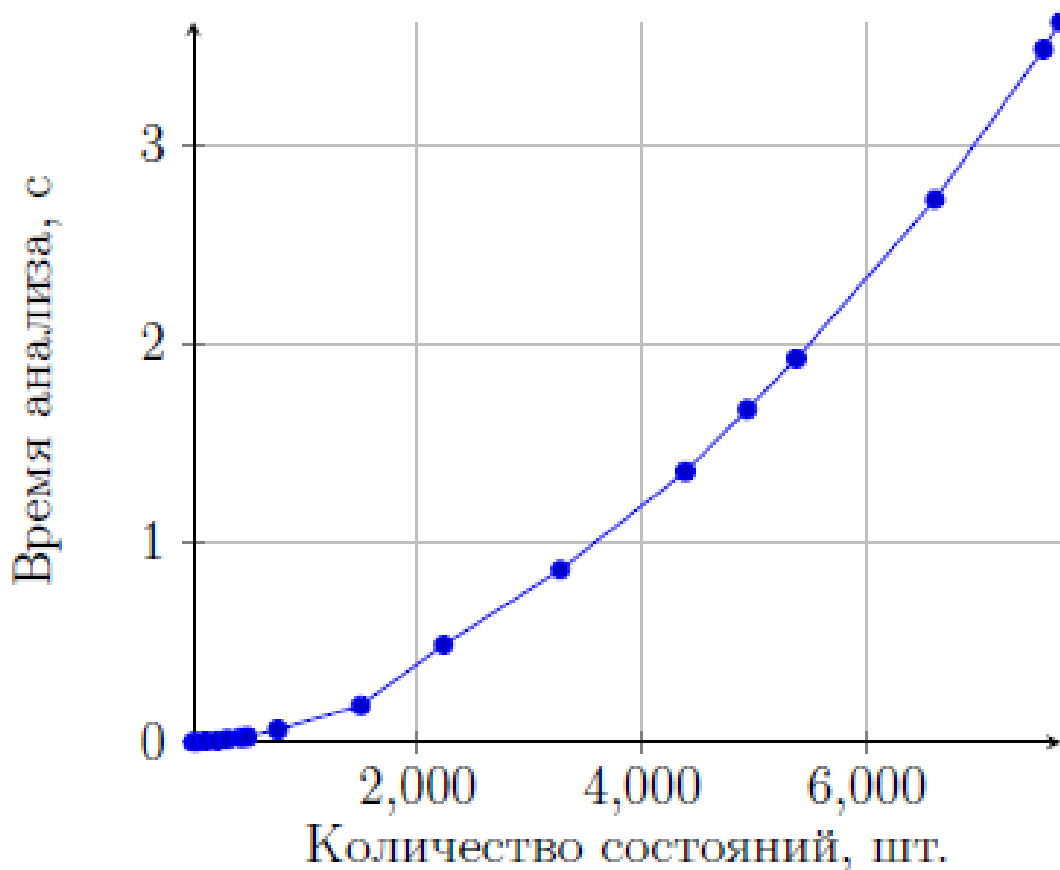


Рис. 4. График зависимости времени анализа от размера пространства состояний сети

В результате анализа построенной сети Петри были найдены начальные состояния, которые при определенной последовательности срабатываний переходов, могут привести к тупиковой ситуации.

Таким образом, исходная диаграмма является корректной при определенных начальных условиях. В исходной диаграмме тупиковая ситуация возникает из-за переполнения канала передачи данных. Для устранения проблемы можно отредактировать диаграмму так, чтобы канал передачи данных освобождался после запроса настроек (Рис. 6). Модифицированная таким образом сеть Петри не содержит некорректных начальных маркировок.

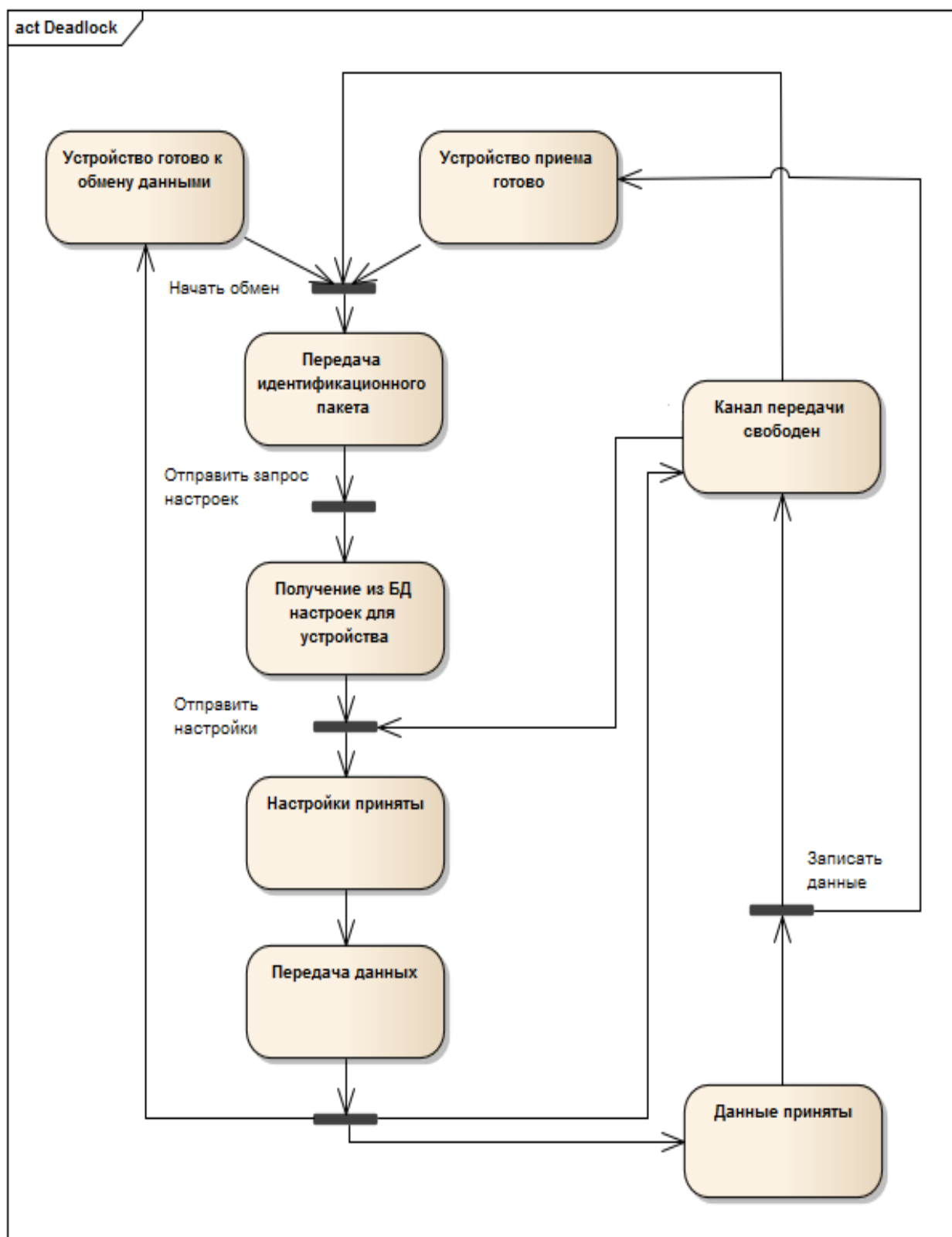


Рис. 5. Диаграмма деятельности сеанса связи центра дистанционного управления

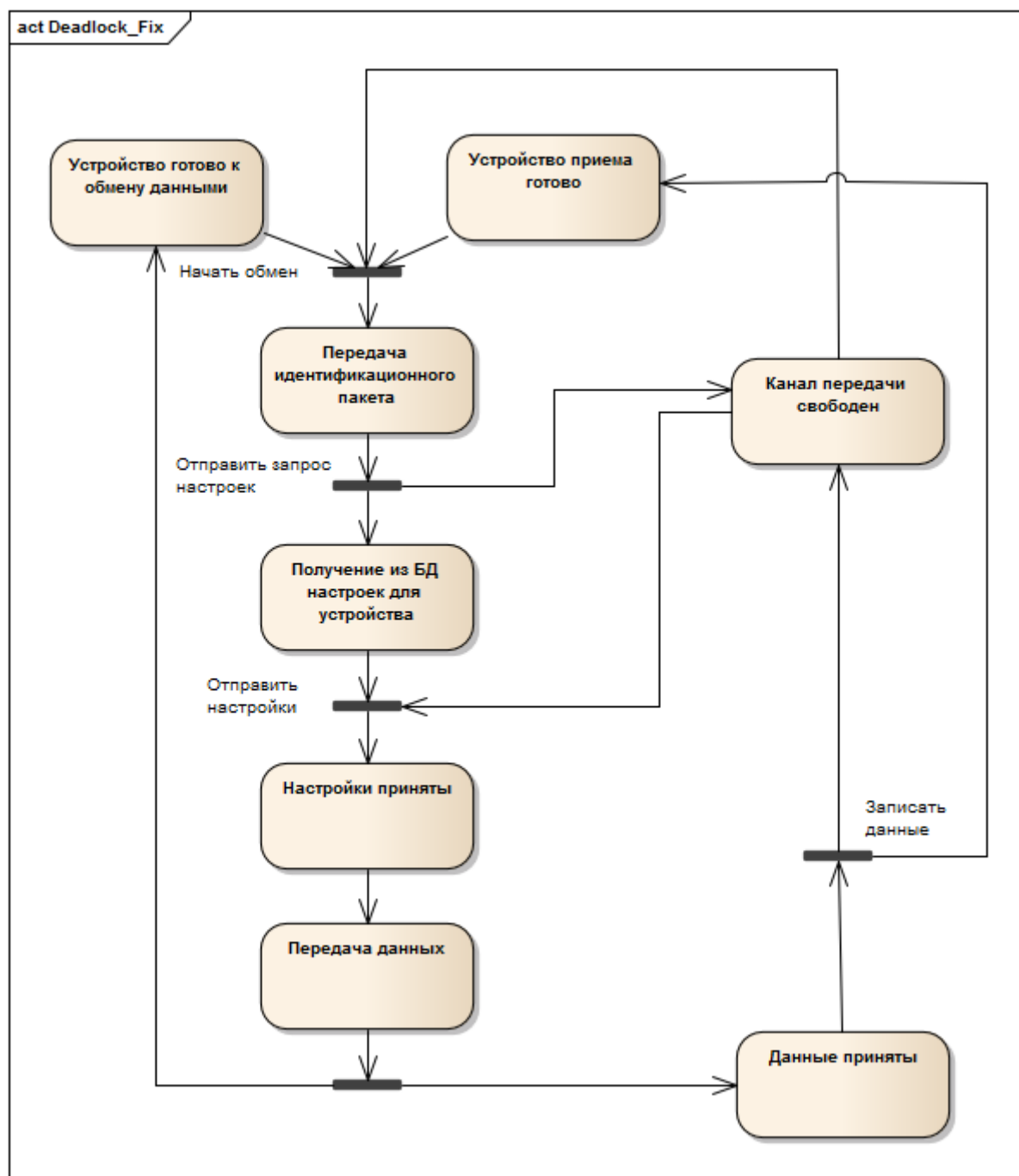


Рис. 6. Модифицированная диаграмма деятельности сеанса связи центра дистанционного управления

Заключение

В рамках данной работы были поставлены следующие задачи исследования разработанного метода:

1. Определение размера сети Петри от размера исходной диаграммы UML.
2. Определение зависимости размера изучаемого объекта от входных данных (начальной маркировки сети).
3. Определение зависимости времени работы метода от размера изучаемого объекта.

В ходе исследования работы метода на примере алгоритма сеанса связи центра дистанционного управления получены следующие результаты:

1. Количество узлов построенной сети Петри превосходит количество узлов диаграммы UML и зависит непосредственно от самой диаграммы.
2. Размер пространства состояний сети Петри экспоненциально зависит от количества фишек в начальной маркировке сети.
3. Время работы метода также экспоненциально зависит от размера пространства состояний сети. Предварительная редукция сети позволяет снизить временную и емкостную сложность алгоритмов анализа.

В результате проделанной работы был разработан метод нахождения некорректных начальных маркировок иерархической сети Петри, применение которого повышает надежность ПО путем обнаружения и дальнейшего устранения возможных тупиковых ситуаций. Создано программное обеспечение, реализующее разработанный метод, и проведено исследование разработанного метода.

Список литературы

1. Коротиков С. В. Применение сетей Петри в разработке программного обеспечения центров дистанционного контроля и управления: дис. канд. техн. наук. — Новосибирск: НГТУ, 2007.
2. Hu Z., Shatz S.M. Mapping UML diagrams to a Petri net notation for system simulation // Proceedings of the international conference on software engineering and knowledge engineering (SEKE), Banff. — Canada, 2004. — С. 213-219.
3. Campos J., Merseguer J. On the integration of UML and Petri nets in software development // Lecture Notes in Computer Science. — Berlin; Heidelberg, 2006. — С. 19-36.
4. Baresi L., Pezze M. On formalizing UML with high-level Petri net // Concurrent object-oriented programming and Petri nets: advances in Petri nets. — Berlin; Heidelberg, 2003. — С. 276-304.
5. Пашенкова А.В. Поиск некорректной начальной маркировки иерархической сети Петри на основе диаграммы состояний UML: магистерская диссертация. — Москва: МГТУ им. Н.Э.Баумана, 2014.