**Слайд 1. Титульник.**

Для упрощенного представления различных систем широко применяются графические схемы как абстрактные модели, опускающие детали и отражающие общую структуру системы или ее поведение. Использование моделей помогает правильно спроектировать систему и выявить возможные противоречия посредством имитационного эксперимента. Примером визуального представления алгоритмов и процессов может служить стандарт выполнения блок-схем (ГОСТ 19.701-90) или соответствующие ему UML-диаграммы деятельности.

Диаграмма деятельности отличающеится от классической блок-схемы наличием элементов для представления многопоточной обработки. При этом в многопоточных системах встают вопросы о достижимости состояний и наличии блокировок. Для решения этих вопросов часто диаграмму деятельности представляют в виде простой сети Петри, при этом моделируется лишь процесс выполнения без привязки к данным. В отличие от простых сетей Петри, в раскрашенных сетях немаловажную роль играет типизация данных, основанная на понятии множества цветов, которое анало­ гично типу в декларативных языках программирования. Таким образом, представив диаграмму в виде раскрашенной сети Петри можно провести моделирование работы с учетом типов входных данных, что максимально приблизит процесс моделирования к реальному функционированию процесса.

Cуществует ряд работ посвященной этой тематике, таких авторов как Котов, Вишняков, Мэллор, но в них незначительное внимание уделяется рассмотрению раскрашенных сетей Петри. Поэтому целью работы…

**Слайд 2-3. Цели и решаемые задачи.**

Является разработка метода представления диаграммы деятельности раскрашенной сетью Петри, позволяющего выявить блокировки и недостижимые состояния.

**Слайд 4. Классификация методов анализа диаграмм деятельности.**

Часто анализ диаграмм производится с помощью конечных автоматов. Автоматные методы анализа широко известны, но такой подход накладывает серьезные ограничения, т.к. с их помощью невозможно представить блоки многопоточной обработки.

Исполняемый UML представляет собой модель системы с описанием ее действий (обычно в виде конечных автоматов), что позволяет по этой модели выстроить рабочий программный код.

Анализ, основанный на простой сети Петри позволяет выявить проблемы, связанные с самой структурой диаграммы, в то время, как раскрашенная сеть позволяет провести моделирование работы системы с привязкой к данным.

**Слайд 4. Функциональная модель метода.**

Функциональная модель метода представлена на слайде.

**Слайд 5. Представление UML диаграмм.**

Основным стандартом описания UML диаграмм является XMI – стандарт OMG для обмена метаданными с помощью языка UML. Для гибкости модели связи между вершинами не описываются напрямую. Вместо этого для каждой вершины приписывается список входных и выходных переходов. Каждый переход содержит информацию о исходящей и входящей вершине и логическое условие для срабатывания перехода.

**Слайд 6. Отображение диаграммы деятельности.**

Исходя из специфики диаграммы деятельности можно представить ее в виде ориентированного графа. Тогда для ее изображения будем использовать алгоритм полилинейного изображения планарного графа. Алгоритм состоит из трех этапов.

**Слайд 7. Построение ассоциированного орграфа.**

На первом этапе необходимо построить ассоциированный с исходным графом граф G\*. Он представляет собой граф связности граней исходного графа. Для его построения необходим получить список граней графа G. Будем понимать под гранью простой цикл графа, не содержащий в себе других циклов.

На рисунках приведен пример нахождения граней графа. Для поиска циклов будем использовать алгоритм поиска в глубину, и, если в процессе обхода было найдено обратное ребро, то выделяем цикл и удаляем первое ребро цикла. Т.к. поиск в глубину происходит слева на права, при удалении ребра разрушается только текущий цикл.s

Топологическая сортировка представляет собой алгоритм упорядочивания вершин бесконтурного ориентированного графа согласно частичному порядку, заданному ребрами орграфа на множестве его вершин.

**Слайд 8. Мозаичное представление графа.**

Мозаичное представление графа – это такое его изображение, в котором каждая вершина, ребро и грань изображается с помощью плитки – прямоугольника, стороны которого параллельны осям координат.

На основе мазаичного представления легко конструируется полилинейное отображение графа. Для этого каждая вершина G может быть представлена некоторой точкой соответствующего вершинного отрезка, а каждая дуга (p, q) графа G ломанной, среднее звено которого образовано частью реберного отрезка, изображающего дугу.

На слайде слево приведено изображение исходного графа G и ассоциированного с ним орграфа G\*. Справо представлено его полилинейное изображение.

**Слайд 9. Раскрашенные сети Петри.**

Раскрашенная сеть Петри представляет собой направленный граф с двумя типами вершин — позициями и переходами, при этом дуги не могут соединять вершины одного типа, т.е. граф является двудольным.

Множество позиции описывают состояния системы. Каждая позиция имеет определенное значение из множества цветов.

Переходы описывают условия изменения состоянии. Переход может сработать, если во всех входных позициях для данного перехода присутствует хотя бы одна фишка и выполнено логическое выражение, ограничивающее переход.

Оснывными отличиями раскрашенных сетей Петри от простых является то, что:

1. каждая позиция имеет свой цвет и определяет тип фишек, которые могут там находиться.
2. каждый переход может иметь спусковую функцию, ограничивающее движение фишек.
3. для манипуляции цветом применяются функции и переменные, цвет переменной может меняться при прохождении перехода.

**Слайд 10. Этапы построения раскрашенной сети Петри.**

Этапы преобразования диаграммы деятельности в раскрашенную сеть Петри приведены на слайде.

**Слайд 11-13. Преобразование диаграммы деятельности в раскрашенную сеть Петри.**

Процесс преобразования диаграммы деятельности в раскрашенную сеть Петри состоит из четырех этапов:

На первом этапе для каждого элемента диаграммы выделяется список переменных и определяется структура каждой переменной.

На втором этапе производится построение сети Петри по структуре диаграммы.

На третьем этапе вводятся начальные значения и определяется тип переменных, строиться предварительное множество раскрасок на основе списков переменных в каждой вершине.

На четвертом этапе для каждой переменной определяется максимальная область ее видимости и с учетом этого формируется результирующая раскраска сети.

**Слайд 14. Исследование причин возникновения блокировок.**

Основной задачей исследования сетей Петри является выявление недоступных состояний и блокировок. Выявим причины, по которым они могут возникнуть в диаграммах деятельности. В процессе исследования представим диаграмму деятельности в виде простой и раскрашенной сети Петри и сравним результаты.

Блокировки в простых сетях Петри могут быть вызваны лишь ситуацией, когда переходу принадлежит два и более места, но не все места имеют фишки.

Ситуация блокировки, изображенная на сладйе искусственная, т.к. по логике выход за пределы параллельных процессов невозможен.

Вторая ситуация, изображенная на слайде возможна, и в результате в системе возникает блокировка, т.к. переход никогда не сработает. Для избежания такой ситуации используется введение дополнительной вершины, из которой будет единственная дуга в блок синхронизации.

Блокировка в раскрашенных сетях может быть вызвана не только отсутствием фишки в позиции, но и не срабатыванием спусковой функции из-за невыполнения логического условия. По этой причине, раскрашенная сеть Петри не всегда способна выявить некорректную структуру диаграммы, т.к. при конкретной начальной разметке, для которой построена сеть, не все переходы будут активны.

**Слайд 15. Исследование причин возникновения блокировок.**

Если в диаграмме имеется хотя бы один цикл, то при наложении раскраски возникает вопрос об определении области видимости переменных. В связи с этим получается два подхода:

1. Определять раскраску всех элементов цикла как кортеж переменных наибольшей мощности.
2. Определять раскраску не учитывая обратную дугу цикла.

Так как фишка хранит в себе набор переменных, сопоставленных ее цвету, то при реализации первого подхода на переходах внутри цикла будут меняться лишь значение переменных. Такой подход будет вести к некорректному моделированию работы, т.к. если какая либо позиция в цикле содержит несколько фишек, то в процессе моделирования будет использована лишь одна из них, и весь процесс работы будет привязан лишь к значениям переменных.

При реализации второго подхода в сети может случиться блокировка, т.к. фишка из раскраски меньшей мощности будет переходить к более мощной раскраске в позиции не содержащей фишек новой раскраски. Другими словами, при моделировании работы диаграммы не сможет сработать переход t6 даже если будет выполнено логическое условие, ограничивающее этот переход, т.к. мощность раскраски позиции p6 содержащей фишку, меньше мощности позиции p2.

**Слайд 16. Сравнение скорости работы простой и раскрашенной сети.**

На слайде представленно сравнение скорости работы простой и раскрашенной сети Петри. В результате исследования получается, что время работы раскрашенной сети значительно больше. Это объясняется тем, что на этапе преобразования диаграммы помимо самого преобразования в сеть Петри происходит так же анализ множества используемых переменных и построение раскраски. Так же во время моделирования для раскрашенной сети на каждом переходе происходит вычисление новых значений переменных, а для каждой дуги из позиции в переход вычисляется спусковая функция.

**Слайд 17. Сравнение скорости работы простой и раскрашенной сети.**

На слайде представленно сравнение потребляемых ресурсов для работы простой и раскрашенной сети.

**Слайд 18. Выводы.**

1. Проведена классифицкация существующих методов анализа диаграмм деятельности.
2. Разработан метод представления диаграммы деятельности в виде раскрашенной сети Петри.
3. Разработано программное обеспечение, реализующее предложенный метод.
4. Исследованы факторы, влияющие на появление блокировок.
5. Исследована корректность построения раскраски сети.