

**Разрешение циклических зависимостей графовой
математической модели триссируемости требований к ПО на
файлы исходного кода**

Содержание

1	Аннотация	2
2	Введение	2
3	Построение математической модели	3
4	Результаты	3
5	Обсуждения	3
6	Заключение	3

1 Аннотация

Некоторый текст анотации

2 Введение

Графовые математические модели нашли широкое применение при решении прикладных задач программирования в самом широком спектре. В настоящее время графовые модели используются для исследования структур компьютерных программ в самых различных аспектах: семантический анализ, иерархическое описание, граф вызова процедур и т.д. Это обусловлено простотой описания актуальных для программирования проблем в терминах графов, а значит они могут быть решены при помощи идентичных хорошо изученных и математически обоснованных механизмов из теории графов.

При описании работы потоков управления, потоков данных, связи составных частей распределенной системы очень важно указывать направление потока или связи. В этих вопросах всегда есть источник, а есть потребитель. Иногда один компонент системы, обозначаемый на графове узлом, является одновременно источником и потребителем. Кроме того, он может быть источником для нескольких потребителей и потребителем от нескольких источников. В ряде задач один узел может быть источником и потребителем для самого себя, например в случае наличия обратных связей в описываемой модели.

Для описания таких связей используются ориентированные графы, в которых по направлению дуг можно судить о принадлежности узла к числу источников или приемников. Благодаря такому подходу может быть решена одна из основополагающих задач при оценке результатов правильности проведенного проектирования системы: описания зависимостей у файлов исходного кода друг на друга, а так же описание трассируемости требований на них.

Благодаря графу трассируемости требований на исходный код с количественной оценкой циклически зависящих друг на друга файлов исходного кода может быть проведена оценка качества проведенного проектирования, выбранной стратегии на разработку ПО, правильность решения о применении или отказе от применения паттернов проектирования и т.д.

В случае описания зависимостей по направлению звеньев из узла и в узел можно судить о том, на какие файлы исходного кода существует зависимость у текущего, а так же для каких файлов он является зависимостью.

В случае описания трассируемости начальными вершинами будут являться файлы исходного кода, а конечными вершинами - требования, реализация которых описана в соответствующих файлах исходного кода.

(рисунок)

Построенный граф отображает, как реализуются требования в файлах исходного кода программы. Заметим, что в звенья данного графа удобно разделить на две группы: группа требований и группа файлов исходного кода. Звенья графа из группы требований не имеют звеньев между друг другом. Кроме того, одно требование может быть реализовано в нескольких файлах исходного кода. Звенья

графа из группы файлов исходного кода имеют звенья между друг другом, так же связи могут образовывать цикл. Однако нет необходимости указывать связь файла на самого себя, так как для цели отображения файлов исходного кода, которые должны войти в сборку такая связь не имеет смысл.

3 Построение математической модели

Объединение звеньев, имеющих друг на друга циклические зависимости, следует производилось итерационно до тех пор пока в результирующем графе не будут отсутствовать циклы. После объединения узлов в одну группу граф видоизменяется. При этом для звеньев, которые объединяются выполняется:

- звенья между друг другом удаляются
- узлы объединяются в группу
- образованная группа является новым звеном графа, при этом:
 - образованная группа является начальным узлом для звеньев между конечными узлами для звеньев, связывающих объединенные узлы
 - образованная группа является конечным узлом для звеньев между начальными узлами для звеньев, связывающих объединенные узлы

При итерационном поиске циклов в графе вышеобозначенные группы учитываются как обычные узлы и таким образом могут участвовать в образовании новых групп по вышеуказанным правилам.

4 Результаты

5 Обсуждения

6 Заключение