

Модель атрибутированных иерархических графов с портами для визуализации структурированной информации

В. Н. Касьянов, email: kvn@iis.nsk.su^{1,2}

Е. В. Касьянова, email: kvn@iis.nsk.su^{1,2}

¹ Институт систем информатики им. А.П. Ершова СО РАН

² Новосибирский государственный университет

***Аннотация.** В данной работе рассматриваются модель атрибутированных иерархических графов с портами для визуализации структурированной информации. Модель позволяет строить наглядные изображения для тех приложений, где объекты, моделируемые вершинами графа, являются сложными и могут содержать по несколько разных логических частей, через которые эти объекты находятся во взаимосвязи, моделируемой ребрами. На базе этой модели в Институте систем информатики им. А.П. Ершова СО РАН ведутся работы по расширению возможностей системы Visual Graph по визуализации структурированных данных большого размера.*

***Ключевые слова:** атрибутированные иерархические графы, визуализация информации, графовые модели, порты, системы визуализации, Visual Graph.*

Введение

Современное информационное общество нельзя представить себе без применения теоретико-графовых методов и алгоритмов. Широкая применимость графов связана с тем, что они являются естественным и наглядным средством объяснения сложных ситуаций на интуитивном уровне. Эти преимущества представления сложных структур и процессов графами становятся еще более ощутимыми при наличии хороших средств их визуализации. Поэтому неслучайно в настоящее время в мире растет интерес к методам и системам визуальной обработки графов и графовых моделей [5, 6, 14, 17], и в настоящее время на рынке широко представлены наукоемкие программные продукты, использующие методы визуализации информации на основе графовых моделей, такие как Cytoscape [7], Higes [18], Gephi [8], Graphviz[9], Tulip [11], yEd [12], Visual Graph [4] и многие другие.

Поскольку информация, которую желательно визуализировать, постоянно увеличивается и усложняется, возникает все больше ситуаций, в которых классические графовые модели перестают быть

адекватными. Требуются и возникают более мощные графовые формализмы для представления информационных моделей, обладающих иерархической структурой [3, 15, 16, 19]. Одним из них являются так называемые иерархические графы [3]. Этот формализм хорошо применим для визуализации сложных структурированных данных большого размера, возникающих в компиляторах и других системах конструирования программ, и позволяет для этих данных строить их наглядные визуальные представления в рамках системы Visual Graph [4].

Во многих приложениях объекты, моделируемые вершинами графа, являются сложными и могут содержать по несколько разных логических частей, через которые эти объекты находятся во взаимосвязи, моделируемой ребрами. Например, в графе авиационных связей страны (или некоторой другой территории) населенные пункты, моделируемые вершинами графа, могут соединяться авиационными рейсами, соединяющими разные аэропорты этих населенных пунктов. А при представлении потока данных между операторами программы в виде так называемого информационного графа у операторов программы, моделируемых вершинами графа, рассматриваются разные их операнды (так называемые информационные входы – разные те места, где данные используются в качестве аргументов операторов, и информационные выходы – разные те места, где данные возникают в качестве результатов операторов), через которые и происходит при исполнении программы обмен данными между операторами (от выходов к входам), и поэтому информационные связи между операторами (вершинами информационного графа) обычно представлены ориентированными ребрами (дугами), которые соединяют соответствующие операнды операторов [2, 6]. При представлении графов с вершинами, моделирующими сложные объекты, в существующих форматах описания графов (см., например, стандартный формат описания графов GraphML [13]) эти разные логические части сложных объектов обычно выражаются с помощью так называемых портов вершин, которые при изображении графа могут представляться разными точками (или разными непересекающимися частями) изображений вершин, в которых соответствующие вершины соединяются с инцидентными им ребрами.

В данной работе рассматриваются модель атрибутированных иерархических графов с портами для визуализации структурированной информации.

1. Иерархические графы и графовые модели

Мы используем без определения ряд стандартных теоретико-графовых обозначений и понятий (см., например, [1]).

Пусть G – граф некоторого типа, например, G может быть неориентированным или ориентированным графом. Граф G определяется двумя конечными множествами V и E , где элементы V – вершины графа G , а элементы E – ориентированные (или неориентированные) ребра графа G . G – тривиальный граф, если $|V| = 1$ и $|E| = 0$.

Граф C называется фрагментом графа G и обозначается $C \subseteq G$, если C – часть графа G , т. е. состоит только из элементов (вершин и ребер) графа G .

Множество фрагментов F называется иерархией вложенных фрагментов графа G , если:

- (1) F содержит граф G , и
- (2) $C_1 \subseteq C_2$, $C_2 \subseteq C_1$ или пусто $C_1 \cap C_2$ для любых $C_1, C_2 \in F$.

Для любых различных $C_1, C_2 \in F$ фрагмент C_1 непосредственно вложен в C_2 , если $C_1 \subseteq C_2$ и не существует такого $C_3 \in F$, отличного от C_1 и C_2 , что $C_1 \subseteq C_3 \subseteq C_2$. Фрагмент $C \in F$ – элементарный, если F не содержит фрагментов, непосредственно вложенных в C .

Иерархический граф $H = (G, T)$ состоит из графа G и корневого дерева T , которое представляет отношение непосредственной вложенности между элементами некоторой иерархии F вложенных фрагментов G . G называется основным графом H . T называется деревом вложенности H .

Иерархический граф $H = (G, T)$ называется простым, если все его фрагменты являются порожденными подграфами графа G . Нетрудно увидеть, что каждый кластерный граф может быть рассмотрен как простой иерархический граф $H = (G, T)$, в котором G – неориентированный граф, а листья дерева T – тривиальные подграфы графа G .

Пример простого иерархического графа $H = (G, T)$ приведен на рис. 1, на котором ребра основного графа G изображены сплошными линиями, а ребра дерева вложенности T – пунктирными. Данный простой иерархический граф H содержит шесть тривиальных фрагментов и два нетривиальных фрагмента: основной граф G и фрагмент $C = \{1, 2, 3, 5\}$.

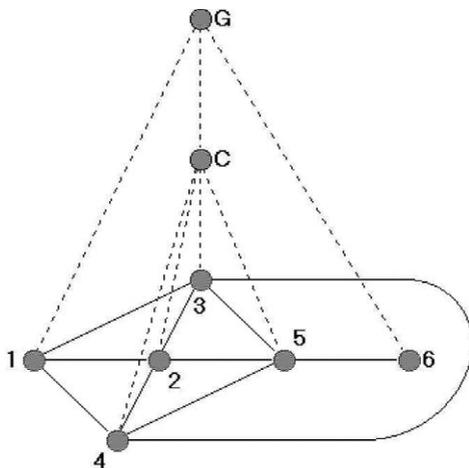


Рис. 1. Простой иерархический граф

Под графовой моделью в общем случае мы понимаем класс графовых объектов, имеющих вид атрибутированных (помеченных) графов, с заданным на нем отношением эквивалентности. При этом при задании графовой модели мы различаем статическую (или синтаксическую) часть описания, определяющую класс помеченных графов, образующих указанную модель, и динамическую (или семантическую) часть, задающую разбиение данного класса графов на подклассы попарно эквивалентных.

Пусть задано множество объектов W , называемых атрибутами, и пусть каждому элементу $w \in W$ сопоставлено в соответствие множество объектов $B(w)$, называемых возможными значениями атрибута w . Например, в качестве $B(w)$ могут использоваться определенные множества чисел, символов или строк (цепочек

символов). Пусть M обозначает множество всех пар (w, v) , образованных из атрибутов $w \in W$ и их значений $v \in B(w)$.

Атрибутированный иерархический граф – это пара (H, L) , где H – иерархический граф, а L – функция атрибутирования, сопоставляющая каждому его элементу h некоторое подмножество $L(h) \subseteq M$.

При изображении атрибутированного иерархического графа атрибуты и их значения для элементов иерархического графа могут либо выражаться неявно через определенные свойства способа представления этих элементов (например, через геометрическую форму изображения соответствующего элемента, его размеры, цветовую гамму и т. д.), либо изображаться явно в виде определенного места и вида пометок соответствующих элементов. Например, явное представление атрибутов может определять место и вид изображений атрибутов в виде текстов внутри областей вершин или рядом с линиями, изображающими ребра.

Что касается динамической части иерархической графовой модели, то она может быть задана разными способами и привносит в визуализацию графовых моделей различные анимационные аспекты.

Можно выделить два разных подхода к заданию семантической части графовой модели: путем явного задания набора инвариантов (свойств, присущих всем эквивалентным между собой моделям), который различает классы эквивалентности графовых моделей, либо через так называемые эквивалентные преобразования графовых моделей, которые сохраняют указанный набор инвариантов. Оба подхода к заданию семантической части графовой модели опираются на преобразования графов и активно развиваются в рамках теории схем программ [2, 6], где первый подход приводит к семантическим моделям программ, используемым главным образом для исследования проблем разрешимости и обоснования корректности преобразований, а второй – к формальным моделям программ, ориентированным на исследование различных формализаций применяемых на практике способов улучшения качества транслируемых программ – так называемых оптимизирующих преобразований.

2. Иерархические графы с портами и их изображение

Определим иерархические графы с портами как подкласс иерархических графов следующим образом.

Пусть $H = (G, T)$ – иерархический граф и пусть $P \subseteq V$ – некоторое выделенное подмножество вершин графа G . Будем называть

портами фрагмента C все те его вершины из P , которые не принадлежат ни одному вложенному в него фрагменту из F .

Нетрудно увидеть, что в любом иерархическом графе с портами $H = (G, T)$ каждая вершина из P является портом некоторого фрагмента $C \in F$ и только одного его фрагмента.

Таким образом, множество всех вершин графа $H = (G, T)$ распадается на три попарно непересекающихся множества: (1) множество всех портов P , (2) множество простых вершин – всех тех вершин его основного графа G , которые не являются портами фрагментов из F , и (3) множество всех тех вершин дерева вложенности T , которые не являются тривиальными фрагментами основного графа.

Например, информационный граф программы может быть рассмотрен как такой иерархический граф с портами $H = (G, T)$, в котором G – ориентированный граф с $P = V$, а каждый фрагмент $C \in F$, отличный от G , является элементарным пустым фрагментом, множество портов которого распадается на два таких непересекающихся подмножества $In(C)$ и $Out(C)$, что нет дуг, исходящих из портов $In(C)$ или заходящих в порты $Out(C)$. При таком рассмотрении элементарные фрагменты $C \in F$ моделируют операторы программы, $In(C)$ и $Out(C)$ – множества информационных входов и выходов соответствующих операторов, а дуги основного графа G – информационные связи между соответствующими выходами и входами операторов.

Изображение (или укладка) иерархического графа с портами $H = (G, T)$ является таким представлением элементов H на плоскости, что выполняются следующие свойства.

1. Каждая вершина графа H представлена некоторой простой замкнутой областью (например, кругом или прямоугольником). Область определяется ее границей (простой замкнутой кривой на плоскости), которая делит оставшуюся часть плоскости на две части: внутреннюю грань и внешнюю грань. Другими словами область, изображающая вершину графа H , состоит из ее границы и ее внутренней грани.

2. Для любых $C_1, C_2 \in F$ пересечение областей фрагментов C_1 и C_2 пусто тогда и только тогда, когда пусто $C_1 \cap C_2$, а область каждого

фрагмента $C \in F$ включает в себя области всех вложенных в него фрагментов и области всех ее портов и простых вершин.

3. Область любого порта и любой простой вершины любого фрагмента $C \in F$ не содержит точек областей других портов и простых вершин фрагмента C и точек областей тех фрагментов, которые вложены во фрагмент C .

4. Каждое ребро графа G представлено простой кривой (со стрелкой, если это ребро ориентировано), соединяющей две точки, принадлежащие границам тех двух областей, которые изображают инцидентные данному ребру вершины.

5. Все ребра любого фрагмента $C \in F$ расположены внутри области фрагмента C .

6. Если простая вершина, порт или ребро h графа H принадлежит некоторому его фрагменту $C \in F$, то область фрагмента C не содержит точек области простой вершины или порта h и не содержит точек кривой, изображающей такое ребро h , которое соединяет вершины, не принадлежащие фрагменту C .

В качестве примера рассмотрим использование введенного формализма для изображения операторной схемы над распределенной памятью (или Р-схемы) [2, 6]. В операторной схеме данного типа управляющий граф (уграф) программы, вершины которого соответствуют операторам программы, а дуги представляют управляющие связи (возможные передачи управления) между ними, как бы дополнен информационным графом, представляющим информационные связи между соответствующими выходами и входами операторов.

На рис 2 приведен пример изображения конкретной схемы над распределенной памятью. Здесь, как и обычно, операторы программы изображены в виде прямоугольников, снабженных кругами, изображающими операнды операторов (входы оператора располагаются сверху соответствующего прямоугольника, а выходы снизу). Управляющие связи изображены сплошными линиями (со стрелками), соединяющими операторы, а информационные связи – пунктирными линиями (со стрелками), соединяющими выходы операторов с входами.

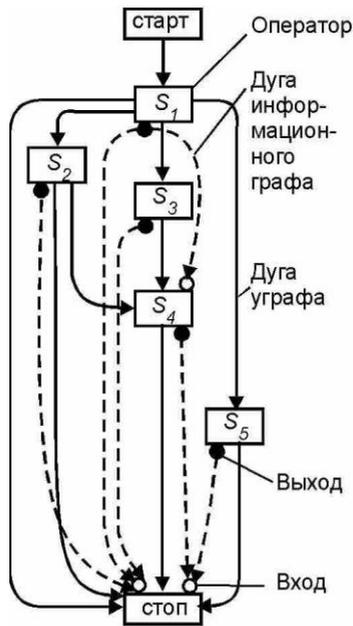


Рис. 2. Р-схема

Каждая Р-схема может быть представлена в виде такого иерархического графа с портами $H = (G, T)$, в котором любой фрагмент $C \in F$, отличный от основного графа G , является элементарным пустым фрагментом, состоящим из одной основной вершины $q \in V \setminus P$ и возможно пустого множества портов $p \in P$, которое распадается на два непересекающихся подмножества $In(C)$ и $Out(C)$. При таком представлении простые вершины q элементарных фрагментов $C \in F$ моделируют операторы Р-схемы, а порты p из множеств $In(C)$ и $Out(C)$ моделируют их информационные входы и выходы. Дуги основного графа G , соединяющие вершины $q \in V \setminus P$, – это управляющие связи, а дуги основного графа G , соединяющие порты $p \in P$, – это информационные связи между соответствующими выходами и входами операторов. Если изображать каждый элементарный фрагмент такого иерархического графа в виде фигуры, образованной прямоугольником, изображающим его вершину, и

примыкающими к нему кругами, изображающими его порты (входы сверху прямоугольника, а выходы снизу), а каждую дугу основного графа рисовать либо сплошной линией со стрелкой, если это управляющая связь, либо пунктирной линией со стрелкой в случае информационной связи, то можно получить стандартное изображение Р-схемы, приведенное на рис. 2.

Заключение

В работе была рассмотрена модель атрибутированных иерархических графов с портами для визуализации структурированной информации. Модель позволяет строить наглядные изображения для тех приложений, где объекты, моделируемые вершинами графа, являются сложными и могут содержать по несколько разных логических частей, через которые эти объекты находятся во взаимосвязи, моделируемой ребрами. На базе этой модели в Институте систем информатики им. А.П. Ершова СО РАН ведутся работы по расширению возможностей системы Visual Graph по визуализации структурированных данных большого размера.

Список литературы

1. Евстигнеев, В.А. Толковый словарь по теории графов в информатике и программировании / В. А. Евстигнеев, В. Н. Касьянов. – Новосибирск: Наука, 1999.
2. Ершов, А. П. Введение в теоретическое программирование. Беседы о методе / А. П. Ершов. – М.: Наука, 1977.
3. Касьянов, В. Н. Иерархические графы и графовые модели: вопросы визуальной обработки / В. Н. Касьянов // Проблемы систем информатики и программирования. – Новосибирск: ИСИ СО РАН, 1999. – С. 7–32.
4. Касьянов, В. Н. Visual Graph – система для визуализации сложно структурированной информации большого объема на основе графовых моделей / В. Н. Касьянов, Т. А. Золотухин // Научная визуализация. – 2015. – Том 7, № 4. – С. 44–59.
5. Касьянов, В. Н. Визуализация информации на основе графовых моделей / В. Н. Касьянов, Е. В. Касьянова. – Новосибирск: НГУ, 2014.
6. Касьянов, В. Н. Графы в программировании: обработка, визуализация и применение / В. Н. Касьянов, В. А. Евстигнеев. – СПб.: БХВ-Петербург, 2003.
7. Система Cytoscape [Электронный ресурс] — Режим доступа: <https://cytoscape.org>

8. Система Gephi [Электронный ресурс]. Режим доступа: <https://gephi.org>
9. Система Graphviz [Электронный ресурс]. URL: <https://graphviz.org>
10. Система JGraph [Электронный ресурс] — Режим доступа: <http://dev.cs.ovgu.de/java/jgraph/tutorial/t1.html>
11. Система Tulip [Электронный ресурс]. URL: <https://tulip.labri.fr/TulipDrupal>
12. Система yEd [Электронный ресурс] – Режим доступа: <http://www.yworks.com>
13. Формат GraphML [Электронный ресурс] – Режим доступа: <http://graphml.graphdrawing.org>
14. Di Battista, G. Graph Drawing: Algorithms for Visualization of Graphs / G. Di Battista, P. Eades [et al]. – Prentice Hall, 1999.
15. Feng, Q. W. Planarity for clustered graphs / Q. W. Feng, R. F. Cohen, P. Eades // Lecture Notes in Computer Science. – 1995. – Vol. 979. – pp. 213–226.
16. Harel, D. On visual formalism / D. Harel //Comm. ACM – 1988. – Vol. 31, № 5. – pp. 514–530
17. Herman, I. Graph visualization and navigation in information visualization: a survey / I. Herman, G. Melançon, M. S. Marshall // IEEE Trans. on Visualization and Computer Graphics. – 2000. – Vol. 6. – pp. 24–43.
18. Lisitsyn, I. A. Higress – visualization system for clustered graphs and graph algorithms / I. A. Lisitsyn, V. N. Kasyanov // Lecture Notes in Computer Science. – 1999. – V.1731. – P.82–89.
19. Sugiyama, K. Visualization of structured digraphs / K. Sugiyama, K. Misue // IEEE Trans. on Systems, Man and Cybernetics. – 1999. – Vol. 21, № 4. –pp. 876–892.