

УДК 004.4'236:004.424.2

Ю.А. КУЗНЕЦОВА

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина*

## МЕТОД ВИЗУАЛИЗАЦИИ УПРАВЛЯЮЩИХ АЛГОРИТМОВ РЕАЛЬНОГО ВРЕМЕНИ

*Показано, что для повышения наглядности программного обеспечения, реализующего управляющий алгоритм, снижения трудоёмкости процессов тестирования и отладки управляющих алгоритмов реального времени, а также возможности их дальнейшей модификации и редактирования без привлечения высокооплачиваемых квалифицированных программистов необходимо разрабатывать эффективные подходы и средства визуализации таких алгоритмов. Универсальным средством представления управляющих алгоритмов являются графы. Показано, что методы «фокус+контекст» позволяют более наглядно представлять графовые структуры пользователю. Предложен метод визуализации управляющих алгоритмов реального времени, в основе которого лежат такие понятия, как «мера центральности» и «априорная значимость вершины». Представлены результаты расчёта функций нелинейного масштабирования и графический интерфейс пользователя программного обеспечения для визуализации управляющих алгоритмов.*

**Ключевые слова:** *управляющий алгоритм, визуализация, графы, методы «фокус+контекст», меры центральности, априорная значимость вершины.*

### Введение

Наиболее важными свойствами программных средств систем реального времени (СРВ) являются детерминированность, реактивность и надёжность. Описание типовой СРВ сводится к построению иерархической модели со сложной структурой информационного взаимодействия подсистем и включает несколько тысяч информационных сигналов и состояний, выполняемых в установленных последовательностях и находящихся в сложных причинно-следственных связях. Программное обеспечение (ПО), реализующее управляющий алгоритм реального времени (УА РВ), в значительной мере определяет логику, динамику и точность функционирования объекта управления. Поэтому проверка корректности работы такого ПО чрезвычайно важна.

При разработке ПО, реализующего УА РВ, основное время уходит на его анализ и отладку. Так, например, наиболее значимая часть испытаний бортовых систем космических аппаратов, относящихся к классу СРВ, проводится в автоматическом режиме, а продолжительность отдельных режимов испытаний составляет часы. При этом сложная иерархическая структура управляющего алгоритма усложняет понимание оператором как структуры УА в целом, так и отдельных его составных частей, что не позволяет ускорить процесс обнаружения и устранения ошибок, которые могут привести к возникновению нештатной ситуации. Основным направлением решения этой проблемы является использование

интерактивной визуализации УА РВ. Такой подход должен учитывать широкий круг различных параметров, в том числе психофизические характеристики пользователей, разнообразие и взаимосвязь решаемых пользователем задач и используемых при их решении данных, неравномерность, слабую предсказуемость и различную приоритетность поступающих запросов.

### 1. Анализ исследований и публикаций

**Интерактивная визуализация информации** представляет собой непрерывный процесс взаимодействия с информацией через визуальное отображение [1]. Пользователь может исследовать данные и манипулировать ими через визуальные метафоры.

Методы визуализации информации предназначены для повышения интерактивности представления абстрактных данных пользователю и усиления его когнитивных возможностей [2]. Такая информация, как правило больших объёмов, содержит полуструктурированные или многомерные данные, не имеющие явной физической интерпретации.

Процесс создания интерактивной визуализации (ИВ) состоит из двух шагов. Первый шаг заключается в отображении данных на геометрическую плоскость. Это проблема визуализации решается чаще всего при помощи методов рисования или размещения графа. К сожалению, ни один алгоритм размещения не может «справиться» в одиночку с графами большого объёма, используемыми в приложениях.

Поэтому вторым неотъемлемым элементом ИВ является навигация, обеспечивающая пошаговое изменение изображения и позволяющая пользователю достичь необходимой информации.

Качественная визуализация информации способствует быстрому нахождению пользователем необходимого элемента в иерархии, пониманию отношения элемента к его контексту и обеспечению возможности прямого доступа к информации.

Бен Шнайдерман сформулировал вербальную схему интерактивной визуализации следующим образом: «Сначала общий план, потом масштабирование и фильтрация, затем детали по требованию», имея в виду действия, необходимые пользователю при взаимодействии с системой [3].

Известно достаточно много графических представлений управляющих алгоритмов реального времени (УА РВ), таких как блок-схемы, временные диаграммы, диаграммы состояний и переходов, деятельности и т.д. Тем не менее, *универсальным средством представления управляющих алгоритмов являются графы*. Поэтому методы визуализации графовых структур представляют собой теоретическую основу методов визуализации УА.

Стандартная формулировка задачи построения изображения графа: дан граф  $G = (V, E)$ , где  $V$  – множество вершин (vertices) графа  $G$ ,  $E$  – множество его рёбер (edges); необходимо построить изображение его вершин ( $V$ ) и рёбер ( $E$ ) на плоскости.

Из-за сложности УА, а также большого количества разнообразных параметров, которыми оперируют алгоритмы данного типа, задача построения изображения графа сильно усложняется из-за невозможности отображения на экране монитора всех вершин и рёбер графа.

Поэтому существует необходимость в разработке методов визуализации графов для представления УА испытаний бортовых систем КА, позволяющих отображать информацию (представлять граф управляющего алгоритма) таким образом, чтобы пользователь мог сфокусировать свое внимание на существенных (интересующих его) объектах (вершинах), игнорируя второстепенные.

Одним из направлений интерактивной визуализации информации являются **методы «фокус+контекст»**, которые предназначены для взаимодействия с изображениями большого объёма и позволяют совместить в одном изображении глобальный вид всей структуры и детали некоторого фрагмента, находящегося в фокусе.

Эффективность этих методов для пользователей была продемонстрирована экспериментально на примере иерархических кластеризованных сетей [4]. Эти методы с переменным успехом применяются к большому количеству различных структур данных,

включая графы, деревья, меню, календари, табличные данные и т.д. Одним из самых известных приложений является TableLens [5].

Помимо стандартных методов навигации, таких как масштабирование и прокрутка, существуют и специальные методы взаимодействия с графовой информацией. Эти методы можно разбить на три большие подгруппы: семантическая и (или) структурная фильтрация вершин, геометрическое или семантическое искажение изображения, абстракция (кластеризация) вершин графа.

## 2. Постановка задачи

Из-за сложности и иерархичности УА, изменения последовательности, объёма и режимов проверок становится затруднительным описание правил представления алгоритмов сбора и преобразований данных, формирования управляющих воздействий и выдачи сообщений оператору. Также отсутствует возможность дальнейшей модификации и редактирования УА без привлечения высокооплачиваемых квалифицированных программистов.

Поэтому *целью* данной работы является повышение наглядности программного обеспечения, реализующего управляющий алгоритм.

Для достижения поставленной цели необходимо разработать метод визуализации УА.

## 3. Метод визуализации управляющего алгоритма

В качестве основы предлагаемого метода применяется известный подход, основанный на семантической фильтрации вершин графа «рыбий глаз».

Под *семантической фильтрацией вершин* понимается удаление из графа подмножеств вершин либо на основе значений индивидуальных атрибутов вершин, либо исходя из структурных свойств графа с целью сокращения визуальной сложности изображения. На данный момент общепринятыми являются две модели фильтрации: фильтрация «рыбий глаз» и глобальная фильтрация. Глобальная фильтрация осуществляется при помощи функции значимости вершин, в то время как фильтрация «рыбий глаз» использует помимо значимости вершин ещё и расстояние до фокусной вершины. Кроме этого, глобальная фильтрация удаляет вершины насовсем, а фильтрация «рыбий глаз» делает их временно невидимыми.

**Метод фильтрации вершин графа «рыбий глаз».** Одним из методов «фокус+контекст», позволяющим более наглядно представлять графовые структуры пользователю, является геометрическое искажение, имитирующее линзу «рыбий глаз» [6].

Базовая идея состоит в искажении информационного пространства таким образом, чтобы элементы,

находящиеся в фокусе, увеличивались в размере, а элементы на периферии – уменьшались (рис. 1).

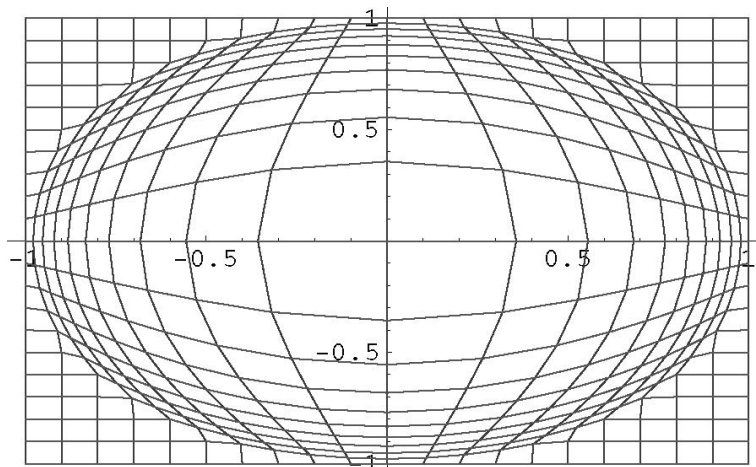


Рис. 1. Искажение регулярной сетки на плоскости «рыбий глаз»: коэффициент искажения равен 4.

В модели Фурнаса [7] функция  $\mathbf{DUI}(\mathbf{v})$  сопоставляет каждому элементу изображаемой структуры число, которое указывает, насколько пользователь может быть заинтересован в том, чтобы увидеть данный элемент.

Первоначально пользователю предоставляется некоторое изображение, на котором он может выбрать одну из вершин в качестве фокусной. Функция степени интереса для всех вершин вычисляется по формуле (1).

$$\mathbf{DUI}(\mathbf{v}) = \text{Antecedent\_Significance}(\mathbf{v}) + \text{Distance\_to\_Focus}(\mathbf{v}), \quad (1)$$

где  $\text{Antecedent\_Significance}(\mathbf{v})$  ( $\mathbf{AS}(\mathbf{v})$ ) – это априорная значимость вершины ( $\mathbf{v}$ ).

Для того, чтобы применить «рыбий глаз» к графу, можно взять в качестве функции расстояния расстояние по графу от вершины до фокуса  $\mathbf{f}$ , т. е.  $D(\mathbf{u}, \mathbf{v}_f) = d(\mathbf{u}, \mathbf{v}_f)$ . Что касается функции  $\mathbf{AS}$ , то её значение можно выбирать либо на основе значений индивидуальных атрибутов вершины, либо на основе структурных свойств графа. Если в приложении визуализации важны структурные свойства графа, то желательно, чтобы функция  $\mathbf{AS}$  вершины показывала, насколько вершина явно или неявно связана с другими вершинами. Для измерения этого свойства подходят меры центральности, введенные для изучения свойств социальных сетей [8]. Наиболее распространёнными являются три меры центральности: степень, близость и промежуточность.

Центральность степени  $\mathbf{C}_D(\mathbf{u})$  – это нормализованное количество рёбер, инцидентных данной вершине  $\mathbf{u}$ :

$$\mathbf{C}_D(\mathbf{u}) = \frac{\text{deg}(\mathbf{u})}{n-1}, \quad (2)$$

где  $n$  – это количество всех вершин графа.

Центральность близости  $\mathbf{C}_C(\mathbf{u})$  оценивается как сумма кратчайших путей, соединяющих данную вершину со всеми остальными вершинами:

$$\mathbf{C}_C(\mathbf{u}) = \frac{\sum_{\mathbf{v} \in \mathbf{V}} d(\mathbf{u}, \mathbf{v})}{n-1}, \quad (3)$$

где  $d(\mathbf{u}, \mathbf{v})$  – это кратчайший путь между вершинами  $\mathbf{u}$  и  $\mathbf{v}$ , который равен количеству рёбер в нём.

Центральность промежуточности  $\mathbf{C}_B(\mathbf{u})$  вычисляется как отношение количества кратчайших путей между двумя вершинами, проходящих через вершину  $\mathbf{u}$ , к количеству всех возможных кратчайших путей в графе:

$$\mathbf{C}'_B(\mathbf{u}) = \sum_{j=1}^n \sum_{k=1}^{j-1} \frac{g_{jk}(\mathbf{u})}{g_{jk}}, \quad (4)$$

а её нормализованное значение:

$$\mathbf{C}_B(\mathbf{u}) = \frac{\mathbf{C}'_B(\mathbf{u})}{(n-1)(n-2)/2}, \quad (5)$$

где  $g_{jk}$  – это количество кратчайших путей между вершинами  $j$  и  $k$ , а  $g_{jk}(\mathbf{u})$  – это количество кратчайших путей, содержащих вершину  $\mathbf{u}$ .

Известно, что данные три меры центральности могут давать противоречивые результаты на одном и том же графе. Поэтому в качестве априорной значимости вершины разумнее использовать комбинированное значение:

$$\mathbf{AS}(\mathbf{u}) = w_1 \mathbf{C}_D(\mathbf{u}) + w_2 \mathbf{C}_C(\mathbf{u}) + w_3 \mathbf{C}_B(\mathbf{u}). \quad (6)$$

Для нормализации функции расстояния  $\mathbf{D}$  необходимо разделить её значение на величину, равную максимальному расстоянию по всем парам вершин, что соответствует диаметру графа:

$$\mathbf{D} = \max d(\mathbf{u}, \mathbf{v}_f). \quad (7)$$

Тогда функция степени интереса для графа имеет вид:

$$DUI(u) = AS(u) - D(u, v_f) = AS(u) - d(u, v_f)/D. \quad (8)$$

При такой нормализации, значение **DUI(u)** принадлежит отрезку  $[-1, 1]$ .

Для получения множества **V'** видимых вершин необходимо задать пороговое значение **t**, тогда:

$$V' = \{u \mid u \in V \wedge DUI(u) \geq t\}. \quad (9)$$

В результате такой фильтрации будут показаны все вершины в окрестности фокусной вершины, в то время как из более удалённых вершин отобразятся наиболее значимые.

**Применение геометрических искажений «рыбий глаз» к изображениям графа.** Саркар и Браун [9] обобщили подход Фурнаса на случай произвольного графа, предложив использовать в качестве функции априорной значимости **AS** один из численных атрибутов, связанных с каждой вершиной, а в качестве расстояния до фокусной вершины – евклидово расстояние между вершинами уже существующего изображения.

Начальное изображение графа называется нормальным изображением, а позиции вершин этого изображения имеют нормальные позиции  $P_{norm}(v)$ . Каждая вершина помимо позиции имеет размер  $S_{norm}(v)$ , который соответствует длине стороны охватывающего прямоугольника вершины **v**. Также каждой вершине **v** соответствует число, характеризующее априорную значимость **AS(v)**. Координаты графа в искажённом изображении называются координатами «рыбьего глаза»  $P_{fish\_eye}$ . Помимо возможности удаления и добавления элементов в изображение структуры Саркар и Браун ввели возможность управления размером и уровнем детализации изображения вершин в соответствии со следующими правилами.

1. Позиция вершины **v** в «рыбьем глазе» зависит от её позиции в нормальном изображении и от расстояния до фокуса **f**:

$$P_{fish\_eye}(v, f) = F_1(P_{norm}(v), P_{norm}(f)). \quad (10)$$

2. Размер вершины **v** в деформированном изображении «рыбий глаз»  $S_{fish\_eye}(v, f)$  зависит от её размера в нормальном изображении  $S_{norm}(v)$ , расстояния до фокуса  $P_{norm}(f)$  и априорной значимости  $AS(v)$ :

$$S_{fish\_eye}(v, f) = F_2(S_{norm}(v), P_{norm}(f), AS(v)). \quad (11)$$

3. Уровень деталей (detail level), которые размещаются вместе с изображением вершины **v** в деформированном изображении  $DL_{fish\_eye}(v, f)$ , зависит от размера вершины после деформации  $S_{fish\_eye}(v, f)$  и максимально изображенной детали:

$$DL_{fish\_eye}(v, f) = F_3(S_{fish\_eye}(v, f), DL_{max}(v)). \quad (12)$$

4. Решение о том, появится ли данная вершина в деформированном изображении, принимается на основе значения «визуальная значимость»:

$$Visual\_Significance = VS(v, f), \quad (13)$$

зависящего от **AS(v)** и от расстояния между вершиной **v** и фокусом **f** в нормальном изображении:

$$VS(v, f) = F_4(P_{norm}(v), P_{norm}(f), AS(v)). \quad (14)$$

В реализации Саркара и Брауна генерация деформированного изображения происходит в два шага. Сначала функция искажения применяется к нормальному изображению для того, чтобы вычислить координаты вершин в зависимости от расстояния до фокуса **f**, а затем используется априорная значимость **AS** каждой вершины для вычисления её окончательного размера, уровня детализации и визуальной значимости. Например, для построения декартового искажения «рыбий глаз», координаты **x** и **y** каждой вершины **v** вычисляются независимо по следующей формуле (функция  $F_1$ ):

$$P_{fish\_eye}(v, f) = \begin{bmatrix} G\left(\frac{D_{norm_x}}{D_{max_x}}\right) D_{max_x} + P_{f_x}, \\ G\left(\frac{D_{norm_y}}{D_{max_y}}\right) D_{max_y} + P_{f_y} \end{bmatrix}, \quad (15)$$

где  $D_{max_x}$  и  $D_{max_y}$  – это расстояния по горизонтали и по вертикали между границей экрана и фокусом **f** изображения в нормальных координатах, а  $D_{norm_x}$  и  $D_{norm_y}$  – расстояния по горизонтали и по вертикали между вершиной **v** и фокусом **f** в нормальных координатах. Отношения  $D_{norm_x}/D_{max_x}$  и  $D_{norm_y}/D_{max_y}$  принимают значения в интервале между 0 и 1. В качестве функции искажения  $G(x) = \frac{(d+1)x}{dx+1}$  целесообразно использовать функцию, которая монотонно возрастает и непрерывна на интервале  $[0; 1]$ , при этом  $G(0) = 0$ ,  $G(1) = 1$ , а константа **d** – коэффициент искажения.

Если  $d = 0$ , то  $G(x) = x$ , поэтому изображение остаётся неизменным. Если  $d = 5$ , то наклон функции  $G(x)$  вблизи фокуса становится больше единицы, а более отдалённые элементы имеют наклон меньше единицы.

Функция преобразования Саркара и Брауна **G(x)** является типичным представителем класса одномерных функций нелинейных преобразований. Хотя эта функция не является слишком затратной с вычислительной точки зрения, у неё есть неудобное требование нормализации координат перед применением. В качестве альтернативного варианта можно подобрать другую функцию  $h(x)$ , не нуждающуюся в нормализации. Таких функций существует немало. Можно, например, использовать гиперболическую функцию  $h(x) = \tanh(x)$ . Также её можно снабдить параметром  $\beta$ , управляющим степенью увеличения:  $h(x, \beta) = \tanh(\beta x)$ , и она не нуждается в предварительной нормализации. Еще одной функцией, дающей результат, аналогичный  $\tanh(x)$  (рис. 2), является функция  $q(x, \alpha) = [2/(1 + e^{-2\alpha x})] - 1$ .

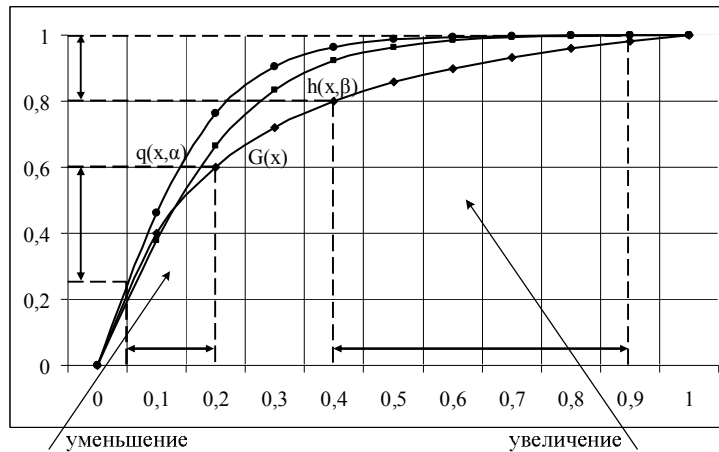


Рис. 2. Графики функций  $G(x)$ ,  $h(x, \beta)$  и  $q(x, \alpha)$  при значении  $d = 5$

С практической точки зрения, основным недостатком навигации при помощи функций нелинейного масштабирования является отсутствие инструментов, позволяющих явно управлять выбором одной или нескольких фокусных областей, количеством пространства, выделяемым для каждой области и уровнем увеличения/уменьшения.

Все эти параметры вычисляются неявно на основе задаваемого пользователем коэффициента искажения  $d$ .

На рис. 3, 4 представлен графический интерфейс пользователя ПО для визуализации графов.

### Заключение

Таким образом, ключевые свойства программного обеспечения, реализующего УА РВ, могут быть достигнуты при разработке методов визуализации таких алгоритмов, что в итоге способствует повышению наглядности ПО, снижению трудоёмкости процессов тестирования и отладки управляющих алгоритмов реального времени, а также возможности их дальнейшей модификации и редактирования без привлечения высокооплачиваемых квалифицированных программистов.

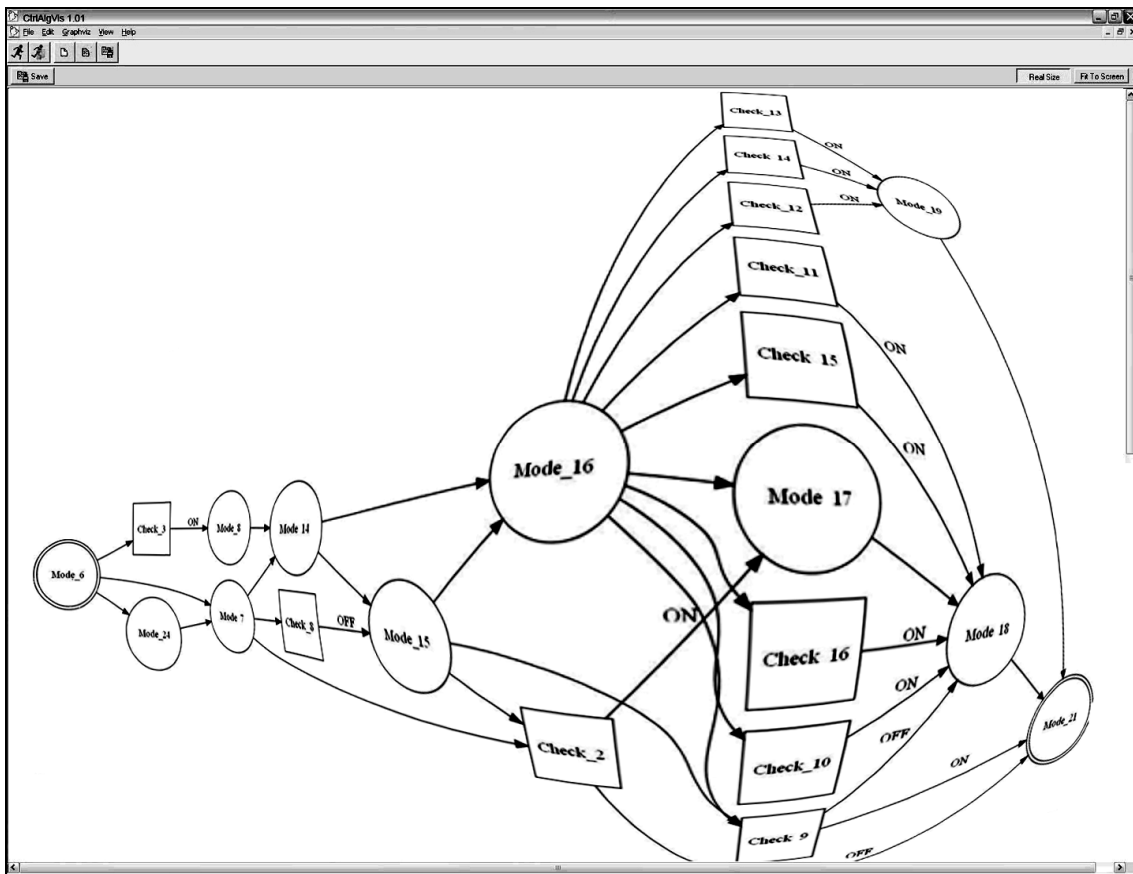


Рис. 4. GUI ПО для визуализации графов: искажение изображения графа «рыбий глаз»

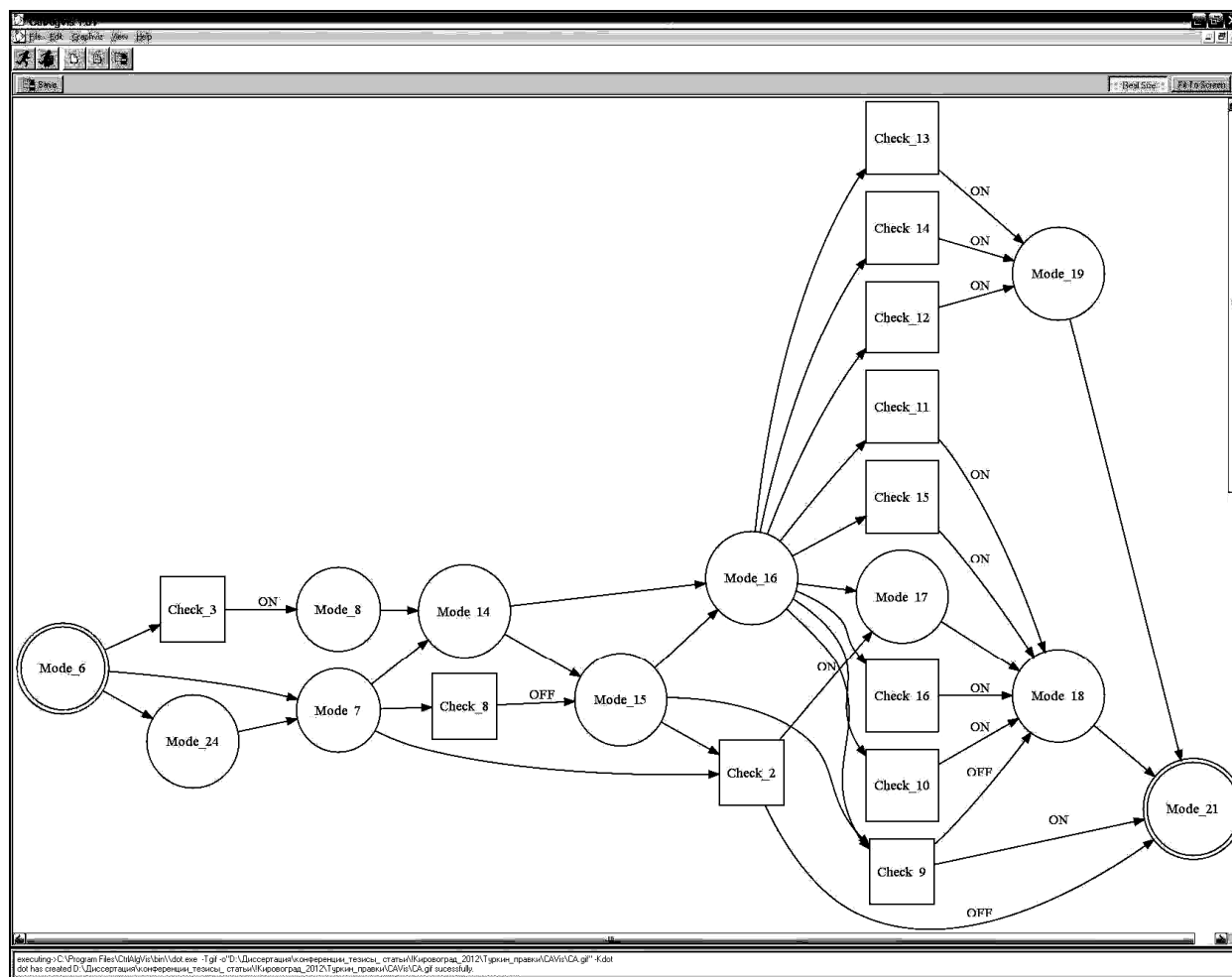


Рис. 3. GUI ПО для визуализации графов: нормальное изображение графа

## Литература

1. Структуры компьютерных метафор [Текст] / В.Л. Авербух, А.Ю. Байдалин, М.О. Бахтеев, Д.Р. Исмаилов и др. // Труды ин-та мат. и мех. УрО РАН. – № 2 (13). – 2007. – 39 с.

2. Panas, Th. A 3D Metaphor for Software Production Visualization [Text] / Th. Panas, R. Berrigan, J. Grundy // 7th International Conference on Information Visualization. – 2003. – P. 314 – 319.

3. Shneiderman, B. The eyes have it: A task by data type taxonomy of information visualizations [Text] / B. Shneiderman // IEEE Symposium on Visual Languages. – Los Alamos, CA: IEEE, 1996. – P. 336 – 343.

4. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Method [Text] / D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, M. Roseman // ACM Transaction on Computer-Human Interaction (TOCHI). – 1996. – P. 162 – 188.

5. Rao, R. TableLens: A Clear Window for Viewing Multivariate Data [Электронный ресурс] / R. Rao. – Режим доступа: <http://www.perceptualedge.com/articles/b-eye/tablelens.pdf>. – 17.02.2012 г.

6. Lin, L. The Fish-eye Visualization of Foreign Currency Exchange Data Streams [Text] / L. Lin, L. Cao, C. Zhang // Asia-Pacific symposium on Information visualisation (APV'05). – 2005. – Volu. 45. – P. 91 – 96.

7. Furnas, G.W. Generalized Fisheye Views [Text] / G.W. Furnas // Human Factors in Computing Systems Conference (CHI'86). – 1986. – P. 16 – 23.

8. Freeman, L.C. Centrality in Social Networks: Conceptual Clarification [Text] / L.C. Freeman // Social Networks. – 1978. – № 1. – P. 215 – 239.

9. Sarkar, M. Graphical Fish-eye views of graphs [Text] / M. Sarkar, M.H. Brown // Human Factors in Computing Systems Conference (CHI '92). – ACM Press, 1992. – P. 83 – 91.

Поступила в редакцию 17.02.2012

**Рецензент:** д-р техн. наук, проф., зав. каф. Экономии и маркетинга В.М. Вартамян, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

## МЕТОД ВІЗУАЛІЗАЦІЇ КЕРУЮЧИХ АЛГОРИТМІВ РЕАЛЬНОГО ЧАСУ

*Ю.А. Кузнецова*

Показано, що для підвищення наочності програмного забезпечення, що реалізує керуючий алгоритм, зменшення трудомісткості процесів тестування і налагодження керуючих алгоритмів реального часу, а також можливості їх подальшої модифікації та редагування без залучення високооплачуваних кваліфікованих програмістів необхідно розробити ефективні підходи та засоби візуалізації таких алгоритмів. Універсальним засобом представлення керуючих алгоритмів є графи. Показано, що методи «фокус+контекст» дозволяють більш наочно представляти графові структури користувачеві. Запропоновано метод візуалізації керуючих алгоритмів реального часу, в основі якого лежать такі поняття, як «міра центральності» і «апріорна значимість вершини». Наведено результати розрахунку функцій нелінійного масштабування та графічний інтерфейс користувача програмного забезпечення для візуалізації керуючих алгоритмів.

**Ключові слова:** керуючий алгоритм, візуалізація, графи, методи «фокус+контекст», міри центральності, апріорна значимість вершини.

## THE REAL-TIME CONTROL ALGORITHM VISUALIZATION METHOD

*Yu.A. Kuznetsova*

This paper shows that need to develop effective approaches and tools for visualization of real-time control algorithms for the visibility increase of the software that implements control algorithm, and for the complexity reduction of the real-time control algorithm test processes and debugging, as well as for the possibilities of their further modification and editing without the involvement of high-paid and highly qualified programmers. Universal means of control algorithm representation are graphs. It is shown that the "focus+context" methods can more clearly represent the graph structure to the user. The real-time control algorithm visualization method, which is based on concepts such as "a measure of centrality" and "a priori importance of the top" are suggested in the paper. The calculation results of the nonlinear scaling functions and a graphical user interface of software for control algorithm visualization are also presented in the paper.

**Key words:** control algorithm, visualization, graphs, "focus+context" methods, a measure of centrality, a priori importance of the top.

**Кузнецова Юлія Анатольевна** – аспірант каф. інженерії програмного забезпечення, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: JK-Sv@yandex.ru.