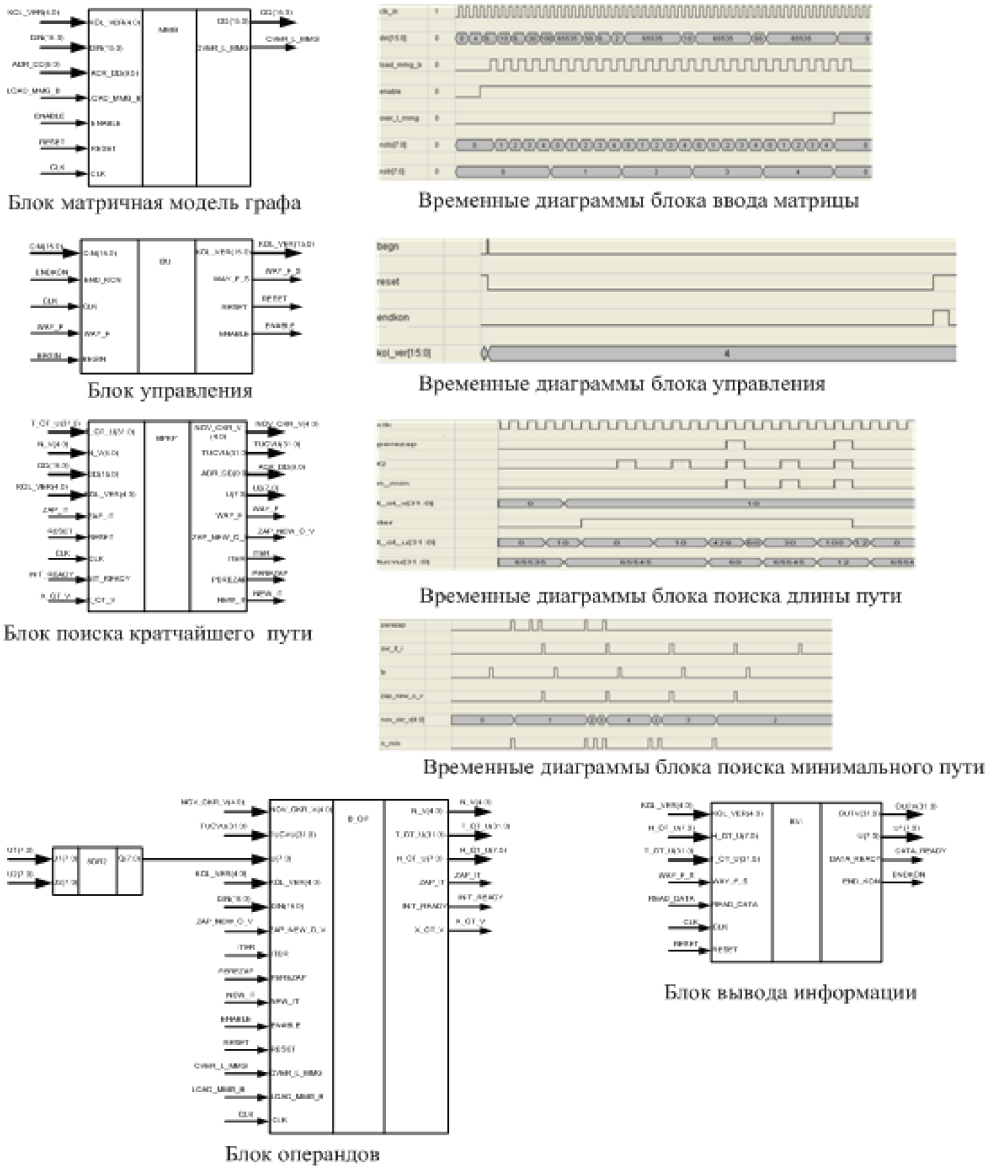


Рис. 2. Функциональная схема блоков ЦУ и их временные диаграммы

Список литературы

1. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ. 2-е изд. – М.: «Вильямс», 2006. – 1296 с.

2. Федосеева Л.И. Устройство оптимизации кратчайшего пути между вершинами графа на примере задачи коммивояжера / Л.И. Федосеева // XXI век: итоги прошлого и проблемы настоящего плюс: пер. науч. изд. – Пенза, 03/2011, с. 132-138.

АППАРАТНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ РЕШЕНИЯ NP ПОЛНЫХ ЗАДАЧ

Юрлов А.А., Федосеева Л.И.

Пензенский государственный технологический университет, Пенза, e-mail: oslit@yandex.ru

Рассмотрим некоторые, формулируемые на языке теории графов задачи, общая особенность которых состоит в том, что они формулируются на взвешенных графах. В зависимости от содержательной интерпретации заданные веса ребер этих графов отражают значения времени, расстояния, стоимости и т.д. Каждый из этих алгоритмов является лишь вычислительной схемой, для которой остается реализовать вычислительный процесс.

В общем виде математическую постановку задачи на графах можно описать следующим образом. Задается n-вершинный взвешенный граф $G=(X, A)$, где $A=\{a_i\}$, i =1, 2,..., m – множество ребер, $X=\{x_i\}$, $i = 1, 2, ..., n$ – множество вершин и $C = \{c_i\}$, $i = 1, 2, 3,…$, m – множество характеристик дуг. Во взвешенном графе G через $c_i(a_i)$ обозначается вес ребра, $a \in A$. Заданием графа неконструктивно определяется множество или система $S_n=\{s\}$ некоторых объектов. Часто s – это некоторая часть графа G. Задается также целевая функция $F= f(s)$, определенная на элементах множества S_n. Задача состоит в том, чтобы в множестве S_n выделить элемент s_0, который является экстремальным согласно целевой функции F. Как правило, мощность множества S_n очень велика по сравнению с числом вершин графа G. Например, если S_n – множество гамильтоновых путей в полном ориентированном графе G, то мощность $|S_n| = n!$.

Алгоритмы решения таких задач принадлежат классу NP полных задач, экспоненциально зависят от размера входа. Решать такие задачи методом перебора всех элементов из S_n не представляется возможным. Поэтому в более точной формулировке смысл решения задачи на графе состоит в том, чтобы построить достаточно эффективный алгоритм, который гарантированно находит оптимум в случае, если множество допустимых решений S_n не пусто. В графе G с числом вершин, равным n, алгоритм можно считать достаточно эффективным, если он находит точное решение задачи и при этом затрачивает порядка n^k операций, где k – константа. Операция – это сравнение двух чисел или арифметическая операция. Теоретический интерес представляют алгоритмы трудоемкости $t=t(n)$, для которых справедливо соотношение $\lim(t(n)/|S_n|)=0$. Если задача сформулирована на взвешенном графе, то множество допустимых её решений обозначаем через $Y=Y(G)=\{y\}$, где $y=(X_y, A_y)$ – это удовлетворяющий соответствующим условиям подграф графа G, $X_y \subset X$ и $A_y \subset A$. Качество допустимых решений y определяется целевой функцией $F(y)$, которая представляет собой суммарный вес ребер подмножества A_y.

Для реализации были выбраны задачи, которые являются классическим примером NP-задач. Это задача коммивояжёра и задача китайского почтальона. Задача коммивояжера заключается в отыскании самого выгодного маршрута, проходящего через указанные города как минимум один раз с последующим возвратом в исходный город. Таким образом, коммивояжер сталкивается с задачей поиска гамильтонова контура минимальной длины. Гамильтонов контур возможен только в связном графе с четными степенями каждой вершины. Поэтому выполняется решение общей задачей коммивояжера, т. е. поиск минимального маршрута с возвратом в точку старта для всех возможных путей. При этом при необходимости вершины посещаются не один раз [2].

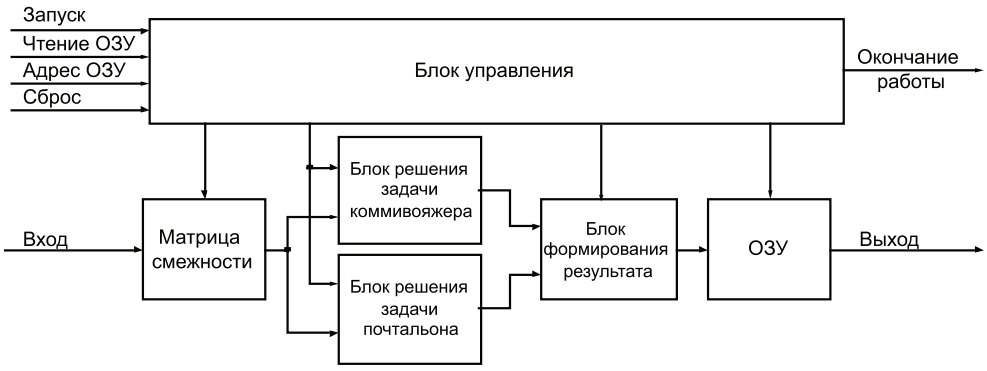
Сформулируем задачу в терминах теории графов. Во взвешенном графе $G=(X, A)$, каждому ребру (x_i, x_j) которого сопоставлен вес $c(x_i, x_j)$, требуется найти гамильтонов контур наименьшей стоимости, причем контур не обязательно должен содержать все ребра графа. Указывается критерий выгодности маршрута и соответствующие матрицы расстояний, стоимости и т.п. Целью решения является нахождения маршрута, удовлетворяющего всем условиям и при этом имеющего минимальную сумму затрат.

Решить задачу можно разными способами, которые отличаются точностью и скоростью решения. В данной работе в основе реализации задачи коммивояжера лежит алгоритм Прима – для нахождения на графе минимального остовного дерева и алгоритм Дейкстры [1, 3] – для построения матрицы кратчайших путей между всеми вершинами графа.

Задача почтальона заключается в том, чтобы пройти все улицы маршрута и вернуться в его начальную точку, минимизируя при этом длину пройденного пути. Сформулируем задачу в терминах теории графов. Во взвешенном n-вершинном связном неографе $G=(X, A)$, каждому ребру (x_i, x_j) которого сопоставлен вес $c(x_i, x_j)$, требуется найти такой кратчайший замкнутый маршрут, который посещает каждое ребро $a_i \in A$ (через некоторые ребра проходит многократно). В условиях задачи указывается критерий выгодности маршрута и соответствующие матрицы расстояний, стоимости и т.п.

Эта задача имеет оптимальное решение только в Эйлеровом графе. В этом случае замкнутый маршрут проходит по каждому ребру единожды. Граф является Эйлеровым тогда и только тогда, когда он связен и степени всех его вершин четны. Если в графе имеются вершины с нечетной степенью, то почтальон должен повторно обойти нечетное число ребер, при этом повторно обходимые ребра образуют цепи, началом и концом которых являются вершины с нечетной степенью. Т. е. почтальон должен решить, какие ребра, соединяющие вершины с нечетной степенью, будет обходить повторно. Эта задача сводится к полиномиальной задаче нахождения оптимального совершенного паросочетания в полном взвешенном графе размерности m<n. Для ее решения использован алгоритм построения паросочетаний с максимальным весом и алгоритм Дейкстры – для построения матрицы кратчайших путей между всеми вершинами графа [2, 3].

Для реализации рассмотренных алгоритмов был разработан специализированный процессор (СП), структурная схема которого приведена на рисунке.



Электрическая структурная схема СП

Матрица смежности содержит электронную модель графа. Каждый элемент матрицы отображает связи между вершинами неографа. Матрица представляет собой блок регистров на 4096 ячеек (64 регистра по 64 бита). Загрузка данных происходит по строкам матрицы, для чего используются регистры сдвига.

ОЗУ предназначено для хранения результатов работы устройства, и по завершении работы может быть считано с помощью сигналов «Чтение ОЗУ» и «Адрес ОЗУ».

Для реализации СП использованы программируемые логические интегральные схемы (ПЛИС) семейства VirtexII Pro.

Список литературы

1. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ. 2-е изд. – М.: «Вильямс», 2006. –1296 с.

2. Майника Э. Алгоритмы оптимизации на сетях и графах. – М.: «Мир», 1981. – 324 с.
3. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.
4. Федосеева Л.И., Юрлов Аппаратная оптимизация графовых задач / Л.И. Федосеева А.А. Юрлов // XXI век: итоги прошлого и проблемы настоящего плюс: пер. науч. изд. – Пенза, №10 (14), 2013, с. 172-175.

Секция «Проблемы моделирования, проектирования и разработки программных средств», научный руководитель – Рыбанов А.А., канд. техн. наук, доцент

ИССЛЕДОВАНИЕ МЕТОДОВ ОЦЕНКИ КАЧЕТСВА ОНТОЛОГИЧЕСКИХ МОДЕЛЕЙ

Андрич О.Ф., Макушкина Л.А.

Волжский политехнический институт, филиал Волгоградского государственного технического университета, e-mail: andrich_olga@mail.ru

Вопрос оценки качества создаваемых онтологических моделей является одной из актуальных проблем современного онтологического инжиниринга. Процесс разработки онтологических моделей важен в практическом плане, и это является причиной того, что разными группами ученых разработано множество различных подходов в области оценки онтологических моделей.

В настоящее время известно более десятка методов, и задача выбора подходящей методики для решения конкретной задачи становится все более сложной.

Целью данной работы является повышение качества онтологических моделей за счет выработки рекомендаций по их построению.

Существует несколько методов оценки качества построенных онтологических моделей [1]:

- FIGO
- OntoMetric
- EvaLexon
- Natural Language Application metrics
- OntoClean
- Declarative Methods

Данные методы проводят оценку онтологических моделей по следующим критериям:

- Полнота и точность словаря предметной области.
- Адекватность структуры с точки зрения таксономии, отношений и т.п.
- Воспринимаемость (с когнитивной точки зрения).
- Производительность.
- Выбор лучшей онтологии из нескольких имеющихся.

Для построения более качественных онтологических моделей необходимо проанализировать существующие методы оценки онтологических моделей, определить недостатки в данных методах, и устранить их, а также в результате анализа усовершенствовать существующие онтологические модели.

Разрабатываемая система оценки качества готовых онтологических моделей предназначена для проведения оценки онтологической модели на основе методов: FIGO, OntoMetric; EvaLexon; Natural Language Application metrics; OntoClean; Declarative Methods – для выдачи рекомендаций по повышению качества модели: по классам, связям (где устранить лишнюю связь, либо добавить новую), а также для оценки сложности модели.

В результате проведённых исследований будет представлено формализованное описание математической модели модуля оценки качества онтологических моделей, а также разработано программное средство оценивающее качество онтологических моделей.

Список литературы

1. Hartmann J. Methods for ontology evaluation // Knowledge Web Deliverable 2005. С. 11-29.
2. Горовой В.А. Модель классификации методов оценки онтологий // Материалы 2-й международной молодежной конференции «Искусственный интеллект: философия, методология, инновации». Санкт-Петербург, 15-17 ноября 2007 – с. 307-310.
3. Gangemi A., Catenacci C., Ciaramita M., Lehmann J. Ontology evaluation and validation // An integrated formal model for the quality diagnostic task. 2005 С. 30-36.
4. Сайт междисциплинарных исследований – http://www.agpl.ru/forum/2-----/4---.html.

ШАБЛОН ПРОЕКТИРОВАНИЯ MVC КАК ЭФФЕКТИВНОЕ СРЕДСТВО ПОСТРОЕНИЯ АРХИТЕКТУРЫ ПРОГРАММНОЙ СИСТЕМЫ

Симонова О.Н., Лясин Д.Н.

Волжский политехнический институт, филиал Волгоградского государственного технического университета, Волжский, Россия, e-mail: olga_troshchenko@mail.ru

В настоящее время весьма часто перед начинающим программистом встает проблема структуризации кода. Наиболее действенным методом в решении данной проблемы является применение шаблона проектирования (паттерна).

Паттерн, или шаблон проектирования представляет собой модель взаимодействия классов для решения какой-либо типичной задачи.

Применение шаблонов проектирования при разработке программного продукта обеспечивает:

1) одинаковое понимание последовательности действий, которые необходимы для решения поставленной задачи, и, как следствие, сокращение времени выполнения поставленной задачи.

2) использование шаблонов проектирования грамотно структурирует программный код, что благотворно влияет на эффективность работы разрабатываемого приложения.

Таким образом, можно сделать вывод: шаблон проектирования (паттерн) является важнейшим инструментом, позволяющим облегчить работу начинающим сотрудникам и увеличить эффективность работы специалиста.

Выделяют 4 основных группы шаблонов:

1. Фундаментальные шаблоны;
2. Порождающие шаблоны;
3. Структурные шаблоны;
4. Поведенческие шаблоны.

Отдельно можно выделить аналитические, коммуникационные, организационные шаблоны, шаблон MVC (данный шаблон заслуживает более детального рассмотрения).

Model-view-controller (MVC)– шаблон проектирования, с помощью которого его компоненты (модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем) разделены на три отдельных так, что модификация одного из них оказывает минимальное воздействие на остальные. Модель предоставляет данные и методы работы с ними, реагирует на запросы, изменяя своё состояние. Представление отвечает за визуализацию.