



ОПТИМИЗАЦИЯ ТОПОЛОГИИ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ И ОЦЕНКА ЭФФЕКТИВНОСТИ СТРУКТУРЫ ПРОГРАММНОГО СРЕДСТВА

Чайников С.И., Солодовников А.С.

Харьковский национальный университет радиоэлектроники

В условиях повышенных требований к времени выполнения вычислительных процессов, посредством диалога с пользователем или без его участия, актуальным остается разработка и применение информационных технологий, позволяющих планировать и оптимизировать выполнение вычислительных процессов с сохранением заданного уровня качества информационной системы (ИС).

В целях оптимизации времени работы проблемно-ориентированной информационной системы и диалога пользователя с системой, необходимо предоставить конечному пользователю возможность выполнения запросов на отображение служебной информации об информационных зависимостях между вычислительными процессами различных подразделений объекта автоматизации, а также предоставить возможность восстановления данных о состоянии процедур для заданного момента времени в случаях обнаружения ошибок в работе отдельных программных модулей.

Постановка задачи.

Пусть задан граф, являющийся формализованным представлением ПрО [1]:

$$Pr = (V_{pr}, X_{pr}),$$

где V_{pr} – множество вершин графа, которым соответствуют подсистемы объекта автоматизации (функциональная подсистема), а X_{pr} – множество ориентированных дуг, соединяющих вершины, которым соответствуют потоки данных между подсистемами.

Пусть также задан функтор отображения вида [1]:

$$\vartheta : Pr \rightarrow G,$$

где $G = (V, U)$ граф, описывающий структуру программного средства, для которого V – множество вершин v , описывающих программные модули информационной системы, а U – множество дуг $u_{ij} = (v_i, v_j)$ графа G , соединяющих вершины v_i и v_j между собой, с заданным направлением, которые, в свою очередь, соответствуют информационным связям между модулями. Причем $\deg^+(v_1) = \deg^-(v_n) = 0$, где n – количество вершин графа.

Необходимо получить такую структуру информационного графа программного средства, которая позволяет получить для заданной вершины v_k подграф $G_k = (V_k, U_k)$, на графе G такой, что $\forall i \in [1, k], \{v_i \in V_k \mid \exists \mu[v_i, v_k]\}$, где $\mu[v_i, v_k]$



– маршрут между вершинами v_i и v_k . Так же необходимо получить множество вершин, которые не имеют информационной зависимости между собой и могут быть размещены на одном ярусе графа, приведенного к ярусно-параллельной форме.

Метод решения.

Контроль и управление вычислительными процессами возможен на граф-моделе, которая описывает ПрО на разных уровнях детализации, включая данные об информационных и управляющих графах для подпрограмм, позволяя получать и хранить истории реализаций [2]. Исходя из того, что в процессе реализации программы могут формироваться разного рода требования к данным, получаемым на этих графах, граф-модель на самом верхнем уровне абстракции должна обладать свойством ацикличности. На ней не должно быть висячих и изолированных вершин с нулевой полустепенью захода для первой вершины и нулевой полустепенью исхода для конечной вершины. Для организации управления вычислительными процессами на информационном графе граф-модель должна быть приведена к конденсированному графу (ацикличная форма), а затем – к ярусно-параллельной форме.

Структура ПС оценивается на основании правил максимизации качества полученной топологии графа и минимизации временных трудозатрат на разработку и тестирование ПС. Данные правила можно сформулировать следующим образом.

Пусть Q – обобщенная оценка качества полученной топологии графа; Q^* – граничное значение качества; C – обобщенная оценка временных трудозатрат; C^* – граничное значение оценки C . Тогда при условии наличия ограничений на временные трудозатраты получаем:

$$T_1^{opt} = \arg \max_{T \in T^*} (Q(T) : C(T) \leq C^*),$$

где T^* – множество всех возможных топологий графа.

А при наличии ограничений на уровень качества структуры:

$$T_2^{opt} = \arg \min_{T \in T^*} (C(T) : Q(T) \geq Q^*).$$

В роли критериев, позволяющих оценить качество структуры ПС, выступают критерии связности, сцепления модулей и невязки топологии структуры программного средства [3].

Для приведения исходного графа к конденсированному графу (графу Герца) [4] $G_{cond} = (V_c, U_c)$ (где V_c – множество конденсированных вершин v_i , для которых индекс i задается согласно правилам топологической сортировки) используется алгоритм Косарайо [5], модифицированный операцией упрощения структуры графа. А именно, вершины v_i, v_{i+1}, \dots, v_j при $i < j$



объединяются друг с другом в супервершину, если существует маршрут $\mu[v_i, v_j]$, в котором найдется вершина v_k , обладающая свойством:

$$\forall k \in (i, j), \{v_k \mid \deg^+(v_k) = \deg^-(v_k) = 1\}.$$

В дальнейшем граф G_{cond} приводится к ярусно-параллельной форме, что позволяет разместить на ярусах независимые по данным вершины графа, обеспечивая параллельность выполнения задач.

Далее для каждого программного модуля (вершины конденсированного графа), задается спецификация, описывающая входные и выходные структуры данных. Эти спецификации позволяют сформировать множество структур данных по выделенному подграфу, связанных с выбранной вершиной, которые хранятся программным средством и служат основой для отката вычислительного процесса (*backtracking*).

Конкретные результаты.

Преимущество подхода заключается в описании предметной области с помощью граф-модели, что позволяет структурировать программу с учетом степени связанности программных модулей. Структура программного средства, основанная на граф-модели, позволяет управлять развертыванием программных компонент на аппаратные ресурсы, и сбором, хранением и отменой результатов вычислительных процессов при их параллельной обработке.

1. Чайніков С.І. Формалізований опис граф-моделі предметної галузі [Текст] / С.І. Чайніков, А.С. Солодовніков // Біоніка інтелекту: наук.-техн. журнал. – 2013. – №1– С.77-81.
2. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл. В. Воеводин. – СПб.: БВХ-Петербург. – 2002. – 608 с.
3. Орлов С.А. Технологии разработки программного обеспечения. Учебник для вузов / С.А. Орлов, Б.Я. Цилькер. – СПб.: Питер. – 2012. – 608 с.
4. Евстигнеев В.А. Применение теории графов в программировании / В.А. Евстигнеев. – М.: Наука. – 1985. – 352 с.
5. Седжвик Р. Фундаментальные алгоритмы на C ++. Часть 5. Алгоритмы на графах / Р. Седжвик. – М.: ДиаСофтЮП. – 2002. – 496 с.