

АЛГОРИТМИЗАЦИЯ ЦЕЛЕУСТРЕМЛЕННОСТИ В СИСТЕМАХ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

М.П. Ревотюк, В.В. Наймович, Р. Хормози

*Белорусский государственный университет информатики и радиоэлектроники,
Минск, Беларусь, rmp@bsuir.by*

Abstract. Template of abstract successor's class for service control systems design, based on finite state machine and network model, was presented.

Рассматривая процесс дистанционного обучения как управляемый процесс, можно говорить о цели управления как поиску и реализации оптимальных переходов между априорно определяемыми состояниями. Набор состояний и переходов обычно предопределен учебными планами и рабочими программами дисциплин, однако связи состояний и критерии оптимизации формально не определяются, а лишь обозначаются. Не претендуя на формализацию аспектов организации образовательной деятельности, рассмотрим подход к представлению моделей систем в рамках известных объектно-ориентированных технологий проектирования, пригодных для изучения в дисциплинах системно-технической направленности [1,2].

Модель целеустремленного поведения может быть задана в конечно-автоматном виде, если выделить уровень планирования и реализации плана на состояниях:

начальное и конечное плановое состояния;

начальное и конечное фактическое состояния.

Каждое из этих состояний может характеризоваться специфичным набором атрибутов, но ключевая информация – тип состояния и момент времени являются инвариантами рекуррентного процесса функционирования любой целеустремленной системы.

Для краткости изложения будем использовать терминологию задач оптимизации перемещений на графах транспортных сетей. Например, пусть процесс освоения изучаемого материала представлен нагруженным ориентированным графом, где вершины соответствуют технологическим задачам дисциплин, а дуги – связям задач с оценкой сложности изучения. Тогда процесс планирования состоит в решении задачи поиска кратчайшего пути на графе. Известно, что использование графовых моделей – стандартный прием описания многошаговых процессов, независимо от интерпретации понятий состояния и переходов между состояниями. Сетевая версия модели весьма привлекательна для представления задач проектирования и программной реализации систем управления в терминах шаблонов проектирования [2,3].

Алгоритмы поиска кратчайшего пути на разреженных графах преследуют цель построения дерева кратчайших путей от исходной вершины до момента достижения конечной вершины. Обратный проход определяет план решения исходной задачи. Здесь множество состояний и оценки сложности решения подзадач определяют план-график работы, где состояния привязаны к моментам времени.

Построение дерева кратчайших путей требует память для хранения описания дерева, объем которой пропорционален размерности пространства поиска. Вместе с тем, задачи поиска путей между задаваемыми узлами формируют результат с ограниченным количеством этапов. Таким образом, при построения системы регулярного решения задач поиска координируемых процессов представляет интерес сокращение потребности в памяти.

Предлагается рассматривать процесс поиска как волновой процесс деятельности независимых порождающих агентов, конкурирующих за ресурсы или решаемые задачи.

Результат поиска в этом случае соответствует генеалогическому дереву порождения агентов из начального состояния до момента достижения некоторым агентом целевого состояния. Отображение состояния поиска проводится на приоритетной очереди, но в отличие от применяемых для синхронизации процессов ветвления двусвязных списков, вершины текущего генеалогического дерева после выборки из очереди дополнительно будем связывать в линейный список фактически использованных элементов памяти.

После построения кратчайшего пути обратным проходом от целевой вершины пометим целевые вершины плана, а затем возвратим в пул свободной памяти непомятые вершины генеалогического дерева.

Часто на этапе реализации плана возможны отклонения от планового графика. Любое отклонение требует поиска нового пути к поставленной цели. Однако при этом полное описание предметной области не требуется, так как достаточно рассматривать область определения дерева кратчайших путей из текущего состояния до цели. Нетрудно заметить, что после построения план-графика и перехода к его реализации возможно упреждающее построение дерева кратчайших путей от конечной вершины до исходной вершины на графе с инвертированными дугами. Предполагая неизменность целевого состояния, в случае отклонения от планового графика достаточно выполнить процедуру поиска из текущего состояния в любую из вершин, принадлежащих дереву на графе с инвертированными дугами.

Таким образом, обработка последствий обнаружения отклонения от планового графика не требует полной перестройки остатка плана. При этом траектория переходов системы между состояниями остается оптимальной, так как переход в новое состояние не нарушает принцип оптимальности Беллмана.

Рассмотренная конечно-автоматная модель целеустремленного поведения может быть задана сетью системы формальных продукций. Однако, в отличие от известных моделей потоков работ или диаграмм Ганта, в контуре управления сеть используется как для поиска решений, так и обнаружения возмущений запланированных траекторий посредством расширения набора продукционных правил. Задачи координации процессов на сети, возникающие из-за свободы выбора альтернатив, могут быть решены посредством упорядочения выходных связей элементов сети.

На примерах задач управления обслуживанием в серверных системах и оптимизации маршрутов на транспортных сетях обсуждаются особенности их представления шаблонами классов и функций, допускающие специализацию на условия применения [2,3]. Рассматриваются варианты программной интерпретации сетевых моделей в реальном времени, а также задачи организации распределенных вычислений.

Литература

1. Ларман, К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку/К. Ларман//М.: Вильямс, 2009. – 736 с.
2. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования/ Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес//СПб.: Питер, 2013. – 368 с.
3. Фаулер, М. Архитектура корпоративных программных приложений/М. Фаулер//М.: Вильямс, 2006. – 544 с.