

Дрождин В.В., Масленников А.А., Сергеев А.С. Поддержка способности программных систем к адаптации. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей VIII Всерос. научно-техн. конф. – Пенза: ПДЗ, 2008. – С. 30-32.

ПОДДЕРЖКА СПОСОБНОСТИ ПРОГРАММНЫХ СИСТЕМ К АДАПТАЦИИ

В.В. Дрождин, А.А. Масленников, А.С. Сергеев

Пензенский государственный педагогический университет
им. В.Г. Белинского,
г. Пенза

Современный уровень развития информационных технологий позволяет создавать сложные программные системы, предназначенные для автоматизации решения широкого круга прикладных задач. Среди таких задач можно выделить хранение и обработку данных, выполнение сложных вычислительных операций, экспертную оценку некоторой информации, прогноз и предсказание каких-либо событий и т.д. При этом все большую актуальность приобретает потребность в создании программного обеспечения, способного адаптироваться, т.е. приспосабливаться к изменениям внешней среды, а также повышать качество своего функционирования посредством изменения собственной организации в процессе эволюции [1]. Отсутствие поддержки автоматической адаптации программных систем может приводить к неоправданно большим экономическим потерям в процессе их эксплуатации, а в ряде случаев – к невозможности создания программного обеспечения, полностью удовлетворяющего потребности пользователя.

В рамках императивной парадигмы разработки программного обеспечения решение любой задачи представляется в виде определенной последовательности действий (алгоритма), преобразующей исходную информацию в требуемый результат. Таким образом, решение прикладной задачи подчиняется жесткой логике, что в значительной степени препятствует свободному общению пользователя с программной системой. В качестве примера можно привести экспертную систему, функционирующую в режиме интерактивного взаимодействия с пользователем. Задачи, реализованные в такой экспертной системе, генерируют ответы в жестко установленной форме, что не позволяет системе адекватно реагировать на недостаточно четкие или не полностью определенные запросы пользователя. Допущенная пользователем ошибка или неточность в формулировке запроса может привести к невозможности обработки этого запроса, к генерации некорректного ответа или даже к сбою и аварийному завершению программы.

Одним из путей повышения эффективности функционирования программного обеспечения за счет повышения гибкости решения прикладных задач является использование принципов нечеткой логики. В ряде предметных областей такой подход можно рассматривать как единственно (на сегодняшний день) применимый для достижения требуемой эффективности программного продукта. В качестве примера можно привести программу отбора клиентов по нестрогим критериям в маркетинговой системе. Однако даже при использовании нечеткой логики

реализация гибкого решения задачи (т.е. способ определения наиболее оптимального варианта решения) осуществляется строго детерминированно. Таким образом, использование нечеткой логики при решении прикладных задач упрощает взаимодействие пользователя с программной системой за счет повышения степени их «взаимопонимания», однако принципиально не решает проблему поддержки способности системы к адаптации, поскольку реализуемые в ней задачи остаются пассивными по отношению к внешней среде.

В качестве возможного способа реализации способности программных приложений к адаптации можно предложить представление программной системы как совокупности взаимодействующих программных компонентов, каждый из которых инкапсулирует решение конкретной прикладной задачи. Такие компоненты, соответствующие прикладным задачам, можно назвать программными задачами по аналогии с программными объектами, соответствующими объектам предметной области. В программных задачах логика стороны клиента может частично или полностью встраиваться в саму задачу, повышая ее автономность и увеличивая потенциальные возможности активного взаимодействия с внешней средой. Свойства автономности и активности позволяют рассматривать программные задачи в качестве агентов и использовать их при создании распределенных программных приложений на основе мультиагентной архитектуры [2]. Для адекватного функционирования такой программной системы необходима поддержка способности к адаптации как в отдельных программных задачах, так и в системе в целом. Поддержка подобных изменений наиболее полно может быть реализована на основе механизмов самоорганизации.

Реализация поддержки способности систем к адаптации позволит повысить эффективность автоматизации решения прикладных задач. Кроме того, способность программных приложений к адаптации позволит увеличить сроки эксплуатации программ, уменьшить затраты на их сопровождение, а следовательно, повысить конкурентоспособность таких программных продуктов на рынке программного обеспечения.

Библиографический список

1. Дрождин, В.В. Системный подход к построению модели данных эволюционных баз данных // Программные продукты и системы. – 2007. – № 3. – С. 52 – 55.
2. Хьюз, К., Хьюз, Т. Параллельное и распределенное программирование на C++; пер. с англ. – М. : Издательский дом «Вильямс», 2004.