МЕХАНИЗМЫ АДАПТАЦИИ В СОВРЕМЕННОЙ ВИРТУАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ СРЕДЕ

А.М. Бершадский, А.С. Бождай, Ю.И. Евсеева
Пензенский государственный университет
Пенза

В последние годы технологии электронного обучения (E-Learning) стремительно набирают обороты в своем развитии. В связи с этим актуальными становятся вопросы, связанные с повышением качества программного обеспечения виртуальных образовательных систем: увеличение срока непрерывной эксплуатации программ, повышение их надежности и гибкости. Перечисленные характеристики напрямую зависят от способности программной системы адаптироваться к изменениям в предметной области, условиям внешней среды и особенностям пользователей.

В настоящее время среди специалистов отсутствует единое определение понятию адаптивности программного обеспечения. Ситуация такова, что для разных типов программных систем адаптивность может определяться различным образом. Если для информационных систем под адаптивностью прежде всего понимается наличие у системы реакций на изменения в предметной области, то для виртуальных тренажеров адаптивность — это чаще всего способность программы изменять свою структуру и поведение в зависимости от действий и личностных особенностей пользователя. Взгляды на реализацию процесса адаптации могут также различаться и в пределах рассмотрения одного класса систем и сильно зависеть от выбранной методологии проектирования.

Тем не менее, несмотря на кажущиеся существенные различия в определениях и подходах, можно выделить одно общее свойство всех систем, называемых адаптивными: способность к самомодификации (на уровне структуры системы в целом либо на уровне отдельных компонентов). Требования, накладываемые внешней средой, особенностями пользователей и предметной области, обычно таковы, что невозможно адекватно адаптироваться к ним без внесения изменений в архитектуру. Однако для каждого типа программной системы процесс реструктуризации по-прежнему будет осуществляться собственным образом и зависеть от разных факторов. Проведенный анализ адаптивного программного обеспечения позволил выделить 4 основных механизма адаптации в программных системах. Характеристики механизмов представлены в таблице.

Таблица. Характеристики механизмов адаптации в программных системах

Механизм адаптации	Используемые модели	Используемые методы
Адаптация предметной области	Модели предметной области	Методы инженерии знаний
Рефлексивная адаптация	Модели самонаблюдения	Методы интеллектуального анализа данных, динамического анализа программного кода
Runtime-адаптация	Модели времени выполнения	Методы искусственного интеллекта, работающие с малыми объемами данных и характеризующиеся относительным быстродействием
Адаптация на основе наблюдения за информационной средой	Модели наблюдения за внешней средой	Методы работы с большими данными (Big Data)

Runtime-адаптация осуществляется программной системой в процессе ее функционирования и характеризуется относительной быстротой реструктуризации программы. Наиболее часто адаптацией по runtime-типу характеризуются системы-тренажеры: собирая данные об обучаемом в процессе выполнения, они без существенных задержек в функционировании программы формируют индивидуальную траекторию обучения.

Другая область возможного применения моделей времени выполнения — построение так называемых самовосстанавливающихся систем [1]. Такие системы должны контролировать собственную надежность и безопасность, а также быть способными автоматизировать задачи, которые зачастую приводят к сбоям системы и требуют внимания специалистов. Самовосстанавливающиеся программные системы должны обеспечивать максимальную устойчивость к отклонениям в условиях эксплуатации.

Реализация крупных образовательных сред (виртуальных, университетов, лабораторий, технопарков) требует повышенной отказоустойчивости, а, следовательно, и включения в них механизмов самовосстановления.

Широкой известностью пользуется предложенный IBM подход к организации runtime-адаптации. Подход был назван MAPE-K (от англ. Monitor, Analyze, Plane and Execute with a Knowledge), и в его основе лежит специфическая архитектура программной системы, реализующая цикл адаптации, состоящий из 4 основных этапов: наблюдения и сбора необходимой информации, анализа полученной информации, планирования поведения программы и выполнения запланированных операций [2].

Ряд других подходов к реализации runtime-адаптации, в частности, подходы, описанные в работах [3, 4], построены на интеграции принципов программной инженерии и теории управлении.

Успешной реализацией runtime-адаптации в E-Learning можно считать проект Smart Sparrow, используемый рядом австралийских университетов и школ [5].

Адаптация предметной области используется преимущественно в информационных системах и системах поддержки принятия решений (СППР). От runtime-адаптации она отличается, во-первых, обязательным участием пользователя (проектировщика или эксперта) при внесении изменений в модели предметной области, а во-вторых, необходимостью временного вывода из эксплуатации модифицируемой системы.

Впервые адаптация предметной области была реализована в ERP-системах. ERP-система — это программный продукт, реализующий принципы концепции ERP (от англ. Enterprise Resource Planning, планирование ресурсов предприятия [6]). Концепция ERP определяет организационную стратегию интеграции производства. Основой ERP-систем является принцип создания единого хранилища данных, содержащего всю корпоративную бизнес-информацию, к которой должен быть обеспечен одновременный доступ определенного количества сотрудников предприятия с различным уровнем полномочий. Поскольку предметные области ERP-систем относятся к быстроизменяющимся, достаточно быстро возникла необходимость в наделении таких систем адаптивными свойствами. Были разработаны 2 подхода к адаптации — оригинальное и типовое проектирование. Оригинальное проектирование основано на использовании CASE-технологий (например, SilverRun) и предполагает генерацию (пересоздание) информационной системы всякий раз, как возникает необходимость в изменениях. Основой типового проектирования являются системы компонентного проектирования (R/3, BAAN, Prodis и т.д.), а вместо регенерации осуществляется конфигурация (адаптация к особенностям экономического объекта) программных систем. В основе обоих подходов лежит принцип постоянно развиваемой модели предметной области, для хранения которой используется специальная база знаний — репозиторий. Именно на основе данных, хранящихся в репозитории, осуществляется генерация или конфигурация программы. Адаптация в таких системах сводится к своевременной корректировке модели предметной области, для построения и последующего изменения которой используется специальный программный инструментарий.

Подходы, разработанные изначально для ERP-систем, в настоящее время распространились на информационные системы, используемые в других областях человеческой деятельности. Не стала исключением и область электронного образования. К числу таких систем в сфере E-Learning относится, например, используемая в немецком Институте Бизнес-информатики платформа электронного обучения OpenUSS [7].

Рефлексивная адаптация характеризуется наличием и сочетанием черт механизмов адаптации предыдущих рассмотренных типов. Как и runtime-адаптация, она не требует существенного участия эксперта для самомодификации системы, а также вывода модифицируемой системы из эксплуатации. Однако, как и в случае с адаптацией предметной области, ей необходимо определенное время для анализа текущего состояния программы и подготовки рекомендаций о ее последующей реструктуризации. В отличие от runtime-адаптации, рефлексивная адаптация не приводит к автоматической самомодификации в процессе выполнения. Ее основным назначением является "offline"-анализ поведения системы за счет использования информации о ее внутреннем устройстве и формирование на его основе решений о возможной реструктуризации. К информации такого рода могут относиться протоколы поведения системы на достаточно длительном промежутке времени либо протоколы процедур динамического анализа исходного кода.

Рефлексивная адаптация может быть реализована в обучающих системах, в частности, виртуальных тренажерах. Протоколирование поведения различных пользователей на протяжении определенного промежутка времени и последующий анализ полученных данных позволит устранить из поведения тренажера проблемные моменты. Под проблемными моментами понимаются те допущенные в процессе проектирования тренажера неточности, которые, согласно результатам анализа, снижают эффективность его функционирования при работе с большим числом обучающихся. К ним могут относиться чрезмерная сложность или простота выполнения отдельных этапов функционирования тренажера, излишняя концентрация на одних этапах в ущерб другим, плохая организация интерфейса, приводящая к неоправданному увеличению времени на освоение программы, и т.д.

Другое назначение рефлексивной адаптации — поиск системой в собственном программном коде фрагментов, способных привести к сбоям в работе или неэффективному функционированию (задача динамического анализа программного кода). Помимо способности к нахождению проблемных мест в

собственной структуре, идеальная адаптивная система также должна обладать способностью к их устранению. Особенно актуальным такое свойство становится для систем, организованных по сервис-ориентированному принципу: добавление некорректно написанного модуля в общую структуру в целом работоспособной системы может привести к непредвиденным сбоям в ее функционировании.

Таким образом, рефлексивная адаптация является эффективным механизмом самооптимизации программной системы. Способность системы к самонаблюдению и последующему формированию решений о реструктуризации — свойство, наличие которого позволяет максимально исключить участие разработчиков в сопровождении программы и увеличить время ее непрерывной эксплуатации.

Элементы рефлексивной адаптации успешно реализованы в системе электронного обучения WebCT, разработанной и впервые внедренной в Университете Британской Колумбии [8]. В WebCT реализован механизм Web Mining — оптимизации интерфейса в соответствии с запросами пользователя.

Адаптация на основе наблюдения за информационной средой является пока что слабо проработанным механизмом реализации адаптивного поведения в программных системах, и большая часть работ в данном направлении посвящена в основном перспективам использования технологии Big Data в построении адаптивных систем подобного плана [9]. Основная идея, лежащая в основе этого механизма адаптации, заключается в использовании различных методов сбора и анализа большого количества данных, относящихся к предметной области программной системы, и последующей реструктуризации системы на основе полученных в результате анализа выводов. В качестве информационной среды, из которой будет осуществляться сбор необходимых сведений, может выступать глобальная сеть Интернет. Данный тип адаптации во многом схож с адаптацией предметной области, но отличается от последнего значительным снижением участия человека в процессе формирования и принятия решения о перестройке программы.

Вопрос об успешной реализации данного механизма адаптации в системах электронного обучения в настоящее время является дискуссионным [10].

Влияние моделей уровня рефлексивной адаптации и уровня наблюдения за информационной средой на модели предметной области и времени выполнения в таких системах хоть и является, на первый взгляд, очевидным фактом, но в настоящее время не имеет достаточной изученности и представляет собой серьезный вопрос для дальнейших исследований. Другим вопросом, представляющим интерес для исследователей, является интеграция рассмотренных механизмов адаптации в рамках единой программной системы.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-07-01553.

ЛИТЕРАТУРА

1. Keromytis A.D. Characterizing Software Self-healing Systems // Computer Network Security. Berlin: Springer, 2007. P. 22-33.
2. Iglesia D.G. MAPE-K Formal Templates for Self-Adaptive Systems: Specifications and Descriptions. Smaland: Linnaeus University, 2014. 43 p.
3. Villegas, N.M., Muller, H.A., Tamura, G., Duchien, L., Casallas, R. A framework for evaluating quality-driven self-adaptive software systems // Proceeding of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. Honolulu: HI, 2011. P. 80-89.
4. Hellerstein, J., Diao, Y., Parekh, S. Feedback control of computing systems. Hoboken: Wiley Interscience, 2004.
5. Smart Sparrow. URL: https://www.smartsparrow.com/ (дата обращения: 22.05.2017).
6. The Next Evolution of ERP: Adaptive ERP // ERP the Right Way: Changing the game for ERP Cloud implementations URL: https://gbeaubouef.wordpress.com/2012/09/05/adaptive-erp/ (дата обращения: 14.05.2017).
7. WWU Munster // OpenUSS. URL: https://www.uni-muenster.de/studium/orga/openuss.html (дата обращения: 22.05.2017).
8. WebCT. URL: http://www.cuhk.edu.hk/eLearning/c_systems/webct6/ (дата обращения: 22.05.2017).
9. Заметки о Big Data // ЕС-Лизинг URL: http://www.ec-leasing.ru/public/publikatsii/index.php?ELEMENT_ID=39 (дата обращения: 28.04.2017).
10. Big Data in eLearning: The Future of eLearning Industry // eLearning Industry. URL: https://elearningindustry.com/big-data-in-elearning-future-of-elearning-industry (дата обращения: 22.05.2017).