

– на другому – загальна модель;
– на третьому – відображення нормативної моделі на загальну модель з класифікацією вимог і сполучення атрибутів.

1. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу : НД ТЗІ 1.1-002-99. – Офіц. вид. – К. : НіКС : Департамент спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України, 1999. – III. 15 с. (Нормативний документ системи технічного захисту інформації).
2. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу : НД ТЗІ 2.5-004-99. – Офіц. вид. – К. : НіКС : Департамент спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України, 1999. – IV. 53 с. (Нормативний документ системи технічного захисту інформації).
3. Положення про державну експертизу у сфері технічного захисту інформації. Затверджено наказом Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України від 29 грудня 1999 року № 62. Зареєстровано в міністерстві юстиції України 24 січня 2000 р. за № 40/4261.
4. Настанови з керування безпекою інформаційних технологій. Частина 3. Методи керування захистом інформаційних технологій : ДСТУ ISO/IEC TR 13335-3:2003. – [Чинний від 2004 – 10 – 01]. – К. : Держспоживстандарт України, 2005. – V. 42 с. – (Національний стандарт України).

Поступила 8.02.2010р.

УДК 004.021:334.722.1(045)

С.В. Карпенко, О. П.Ткалич, П. А. Андрухович

СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА И АРХИТЕКТУРА, УПРАВЛЯЕМАЯ МОДЕЛЯМИ

In the article the service-oriented architecture (SOA) manageable with the help of the models is described. The main concepts of SOA are discussed. The main aspects and advantages of implementation of SOA in organizations of different types are shown. In comparison the Model-driven architecture concepts of creation of organization model are described.

Необходимость интеграции и взаимодействия приложений в рамках совокупности большого количества информационных систем (ИС) предприятия (или нескольких предприятий), оказывают существенное влияние на используемые программные архитектуры. Прежде всего, это сервисная модель взаимодействия между приложениями общей системы в рамках так называемой сервис-ориентированной архитектуры (Service-

oriented architecture SOA) и реализация архитектуры, "управляемой моделями" (Model-driven architecture MDA).

Концепция сервис-ориентированной архитектуры была сформулирована специалистами в области информационных технологий (ИТ), но в действительности это был прямой ответ на потребности сегодняшнего дня, когда становится уже не совсем понятно, где заканчиваются бизнес-функции организации и начинаются информационные технологии, их обеспечивающие, и наоборот. Ведущие поставщики ИТ, такие как Microsoft [1] и IBM [2], развивают эту концепцию в рекомендациях по проектированию ИС на своих программных платформах. А такие компании, как Gartner, считают, что SOA будет ведущим принципом проектирования новых критически-важных прикладных систем и бизнес-процессов в ближайшем будущем.

В основе бизнес-архитектуры должна быть процесс-ориентированная модель функций предприятия. Комбинация этого подхода с концепцией SOA ИТ позволяет лучше увязать процесс разработки компонент ИС с миссией, основными задачами и функциями организаций. С помощью SOA организации имеют потенциальную возможность разрабатывать набор реализаций различных бизнес-процессов, которые могут быть многократно использованы предприятием как готовые сервисы.

Под SOA понимается подход к проектированию прикладных ИС, который руководствуется следующими принципами [3]:

- явное отделение бизнес-логики прикладной системы от логики представления информации;
- реализация бизнес-логики прикладной системы в виде некоторого количества программных модулей, которые доступны извне, чаще всего в режиме "запрос-ответ", через четко определенные формальные интерфейсы доступа;
- при этом "потребитель услуги", который может быть прикладной системой или другим сервисом, имеет возможность вызвать сервис через интерфейсы, используя соответствующие коммуникационные механизмы.

В целом, SOA представляет собой модель взаимодействия компонент, которая связывает различные функциональные модули приложений между собой с помощью четко определяемых интерфейсов. Интерфейсы сами по себе не зависят от используемых аппаратных платформ, операционных систем или языков программирования, используемых для разработки этих приложений. Это позволяет отдельным сервисам взаимодействовать между собой одним и тем же стандартным, но в то же время универсальным способом. Такая особенность использования интерфейса, независимого от окружения и платформы, получила название модели "слабой связи". Ее очевидным преимуществом является повышенная гибкость и адаптируемость, поскольку замена или модернизация одной из компонент системы не сказывается на остальных [4].

Само понятие SOA не является чем-то принципиально новым, так как сходные возможности реализовывались и ранее, например, с помощью технологий обмена сообщениями. Принципиальным фактором является то, что современные подходы к реализации SOA охватывают не только технологический уровень обмена данными, но и уровень бизнес-операций. Принятие SOA как целевой архитектуры будет подразумевать и соответствующий подход к разработке приложений (SODA – service-oriented development architecture). Для задач электронного бизнеса соответствующая функциональность SOA реализуется на уровне web-сервисов. Под web-сервисами понимаются программные системы, которые используют XML в качестве формата данных, стандарты Web Services Description Language (WSDL) для описания своих интерфейсов, Simple Object Access Protocol (SOAP) для описания формата принимаемых и посылаемых сообщений и стандарт Universal Description, Discovery and Integration (UDDI) для создания каталогов доступных сервисов. И хотя принципы SOA создания информационных систем не обязательно предполагают использование технологий web-сервисов, связь между этими двумя направлениями в развитии ИТ является достаточно тесной. При этом web-сервисы являются технологическими спецификациями, в то время как SOA является принципом проектирования архитектуры программных систем.

С учетом существующих тенденций перехода к бизнесу "реального времени" и создания систем так называемого "расширенного предприятия", объединяющих само предприятие, его поставщиков, партнеров и клиентов в единую систему, становится очевидно, что технологии web-сервисов найдут свое применение на всех уровнях корпоративных ИС. Предполагается, что в будущем практически все взаимодействие приложений как в рамках одной ИС, так и между системами отдельных участников бизнес-процесса, будет осуществляться с использованием такого механизма, так что достаточно критическими становятся вопросы согласованной работы этих сервисов.

Более того, даже интеграцию унаследованных приложений целесообразно проводить с применением данной технологии, когда определенная, наиболее важная часть существующей функциональности инкапсулируется и представляется стандартизированным интерфейсом. При наличии существующей инфраструктуры web-сервисов на предприятии можно ожидать существенного сокращения сроков и затрат на интеграцию.

Таким образом, влияние сервис-ориентированных подходов на изменения в архитектуре можно охарактеризовать как сбалансированный переход от централизованной инфраструктуры ИТ и замкнутого на себе функционала прикладных систем в сторону архитектуры, обеспечивающей возможности быстрого создания новых систем из набора доступных сервисов, т.е. более гибкой, динамичной и способной к взаимодействию. Ориентация на сервисную архитектуру позволяет построить комплексную ссылочную модель архитектуры предприятия, которая в единой манере описывает как бизнес, так и ИТ [5]. Эта модель состоит из следующих

основных компонент (рис 1):

- презентационный уровень описывает интерфейсные сервисы для взаимодействия пользователей с ИС, включая корпоративные и публичные порталы, доступ с мобильных устройств, а также различные преобразования информации при взаимодействии с внешними системами и устройствами;
- на уровне бизнес-сервисов формируются модели и осуществляется управление выполнением бизнес-процессов предприятия с использованием специализированных средств, а также координация автоматизированных и "ручных" операций;
- интеграционные сервисы обеспечивают взаимодействие между приложениями, которое может быть реализовано, в частности, с использованием средств обмена сообщениями или в рамках единой среды исполнения, такой как сервер приложений J2EE;



Рис. 1 Ссылочная модель сервис-ориентированной архитектуры предприятия

- сервисы уровня данных реализуют средства извлечения и повторного использования данных из СУБД и приложений. Явное выделение такого уровня позволяет изолировать вышестоящие компоненты архитектуры от изменений в технологиях, а также обеспечить единый унифицированный подход к выполнению операций с данными;
- уровень инфраструктуры, приложений и СУБД является основой для всей структуры, и именно здесь концентрируются основные инвестиции в ИТ.

Взаимодействие между этими уровнями, однако, осуществляется не напрямую, а через сервисы, выделенные на уровень обработки событий.

Сервисы этой компоненты архитектуры обеспечивают сбор данных о событиях в масштабе всего предприятия, необходимое преобразование и маршрутизацию этих данных между разными уровнями, а также "обратную связь" между сервисами каждого отдельного уровня.

В предложенной модели, наряду с рассмотренными уровнями, отвечающими за взаимодействие различных групп сервисов в процессе деятельности предприятия, выделяется отдельная компонента архитектуры, которая описывает аспекты, связанные с жизненным циклом сервисов – то есть их созданием, эксплуатацией и оптимизацией.

MDA является еще одной важной архитектурной концепцией создания информационных систем. Концепция MDA была предложена консорциумом OMG (Object Management Group), в который сегодня входит более 800 известных производителей программного и аппаратного обеспечения. MDA является определенным обобщением идей SOA, с одной стороны, и повторно используемых программных компонент (шаблонов, паттернов) с другой, предназначенным для повышения гибкости разрабатываемых приложений масштаба предприятия, чтобы обеспечить простоту обеспечения соответствия требованиям бизнеса в условиях изменения используемых инфраструктурных платформ. MDA по определению является открытой и "нейтральной" по отношению к используемым технологиям интеграции. Она основана на следующих принципах [6]:

- основой для разработки приложений масштаба предприятия являются детальные модели с общепринятой нотацией;
- построение систем может быть организовано в соответствии с рамочной системой моделей, которые позволяют отделить бизнес-логику приложений от конкретной реализации. Исходной является так называемая независимая модель вычислений (Computational Independent Model), которая путем последовательных трансформаций через платформу-независимые (PIM) и платформу-специфичные модели (PSM) автоматически или с минимальным участием программиста приводится к исполняемому коду и соответствующим структурам данных;
- существует формальное описание используемых моделей на основе системы метамodelей (в частности, для таких областей как распределенные вычисления и транзакции, операции в реальном времени и т.п.);
- принятие и широкое использование этого подхода основано на открытости промышленных стандартов и на поддержке со стороны производителей соответствующих средств разработки.

В рамках такого подхода сначала создается архитектура, которая описывает модель бизнес-функциональности и поведения прикладной системы независимо от технических деталей реализации. Эта разработка должна вестись в контексте всей корпоративной архитектуры организации.

На основе этой модели, не зависящей от платформы реализации, может быть разработана одна или несколько специфических для конкретной платформы моделей, в зависимости от того, какая платформа используется и поддерживается организацией. Уже на основе этих специфических для конкретной платформы моделей разрабатывается код конкретной прикладной системы, как показано на рисунке 2.

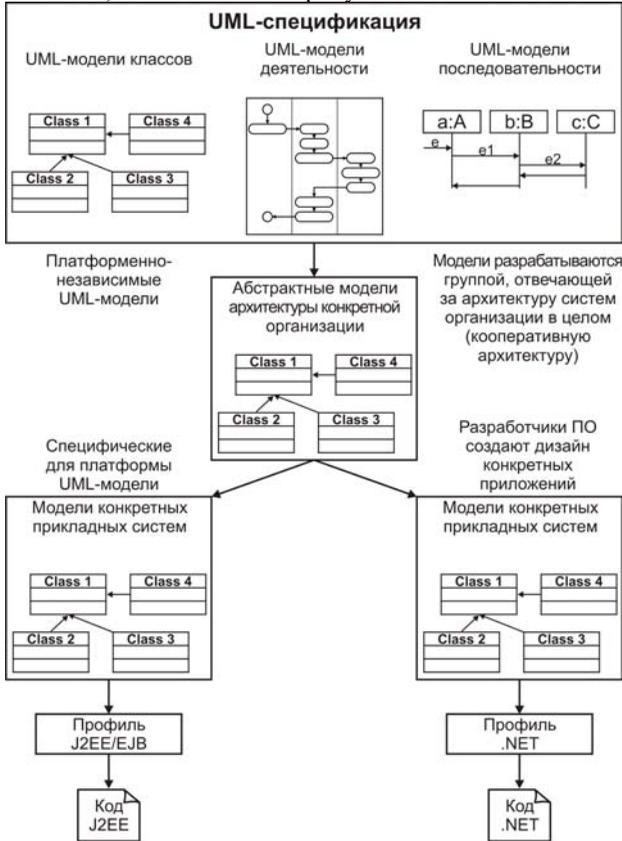


Рис. 2 Создание прикладных систем в соответствии с подходом MDA

Этот подход не определяет, какие языки разработки, операционные системы или программное обеспечение промежуточного слоя будут использоваться на практике. Наоборот, упор делается на описание того, как прикладные системы организованы с точки зрения процессов и как они интегрированы между собой.

После того как эти высокоуровневые связи определены, могут использоваться соответствующие средства для разработки приложения с использованием конкретных языков и ПО промежуточного слоя. Таким

образом, процес дозволяє скоротити цикл розробки ІТ-систем и в то же время дає гнучкість и можливість швидкого внесення змін.

1. *Mullender, M. and M. Burner* Application Architecture: Conceptual View, 2003
2. *Burbeck, S.* The Tao of E-Business Services, 2004, IBM
3. Introduction to Service-Oriented Architecture Gartner, 2003
4. <http://www.osp.ru/os/2003/09/034.htm>
5. <http://www.doculabs.com/default.aspx>
6. *Buchanan R., Soley R* Aligning Enterprise Architecture and IT Investments with Corporate Goals. 2005

Поступила 16.02.2010р.

УДК 159.9.072.432

А.К. Оленич

ДЕЯКІ АСПЕКТИ МОДЕЛЮВАННЯ ОСОБЛИВОСТЕЙ ВПЛИВУ БАТЬКІВСЬКИХ УСТАНОВОК НА СПРИЙМАННЯ РЕКЛАМИ ДІТЬМИ

Метою дослідження є визначення моделі сприймання реклами дітьми, що отримали різні батьківські установки щодо соціально-економічних функцій реклами.

Об'єктом дослідження є діти та їх батьки.

Предметом дослідження є ефективність застосування батьківських установок щодо реклами

Дане дослідження повинне виконати ряд певних завдань. Визначимо їх перелік:

- Визначення впливу на дітей референтних та нереферентних членів сім'ї;
- Виявити присутність батьківських установок щодо реклами;
- Визначити яким чином впливають батьківські установки на сприйняття дитиною реклами;
- Надання батькам рекомендацій по правильному позиціонуванні реклами;

Серед **методів** дослідження було обрано опитування для виявлення референтного члена сім'ї і бесіда з дітьми та їх батьками для визначення відношення до реклами, як дітей так і батьків, та присутність установок щодо неї.

Для опитування було обрано проєктивну візуально-вербальну методику Рене Жіля. Метою її є дослідження соціальної пристосованості дитини молодшого шкільного віку, сфери його міжособистісних стосунків і їх