

**A Lotus Software White Paper**

Август 2002

**Lotus** software



## **Технологическая стратегия и планы разработчиков Domino**



## Содержание

Аннотация

Технологическая стратегия и планы разработчиков Domino.

Значение приложений Domino и модели разработки приложений.

Значение приложений J2EE и модели разработки приложений.

Выбор магистрального направления

1. Цель приложения и его использование
2. Требования к масштабируемости
3. Требования к данным
4. Срок службы приложения
5. Факторы среды

Будущее систем быстрой разработки приложений (RAD)

Заключение: Подготовка к будущему

## Аннотация

В этом документе представлены взгляды IBM на будущее систем для совместной работы и оценка текущего состояния приложений Domino и навыков разработчиков в плане возможной степени их усиления для использования в следующем поколении приложений для групповой работы. Здесь разъясняются планы IBM продолжить инвестиции в Lotus Notes и Domino, которые являются дополнительной и параллельной архитектурой к IBM WebSphere, подтверждая лидерство Lotus в области сред для совместной работы в IBM и в индустрии в целом. В настоящее время и в обозримом будущем IBM не планирует объединять Lotus Domino с WebSphere и не стремится к тому, чтобы приложения Domino выполнялись исключительно в среде WebSphere.

Этот документ поможет вам также понять основные достоинства и преимущества как Lotus Domino, так и WebSphere. Здесь обсуждается ряд факторов и даны рекомендации, при в каких случаях лучше использовать Lotus Domino или WebSphere или же оба эти продукта. Документ заканчивается призывом к действию, подготавливая вас к поддержке и развертыванию в будущем интегрированных решений в области сред для совместной работы.

## Технологическая стратегия Lotus и основные планы разработчиков

В начале 2002 г. подразделение IBM Lotus Software анонсировало свою технологическую стратегию в области разработки сред для групповой работы нового поколения. Стратегия разработки следующего поколения этих продуктов состоит в совершенствовании функций коллективного использования в продуктах Lotus, что позволит им легче интегрироваться друг с другом и с приложениями собственной разработки или других производителей. Планы Lotus таковы:

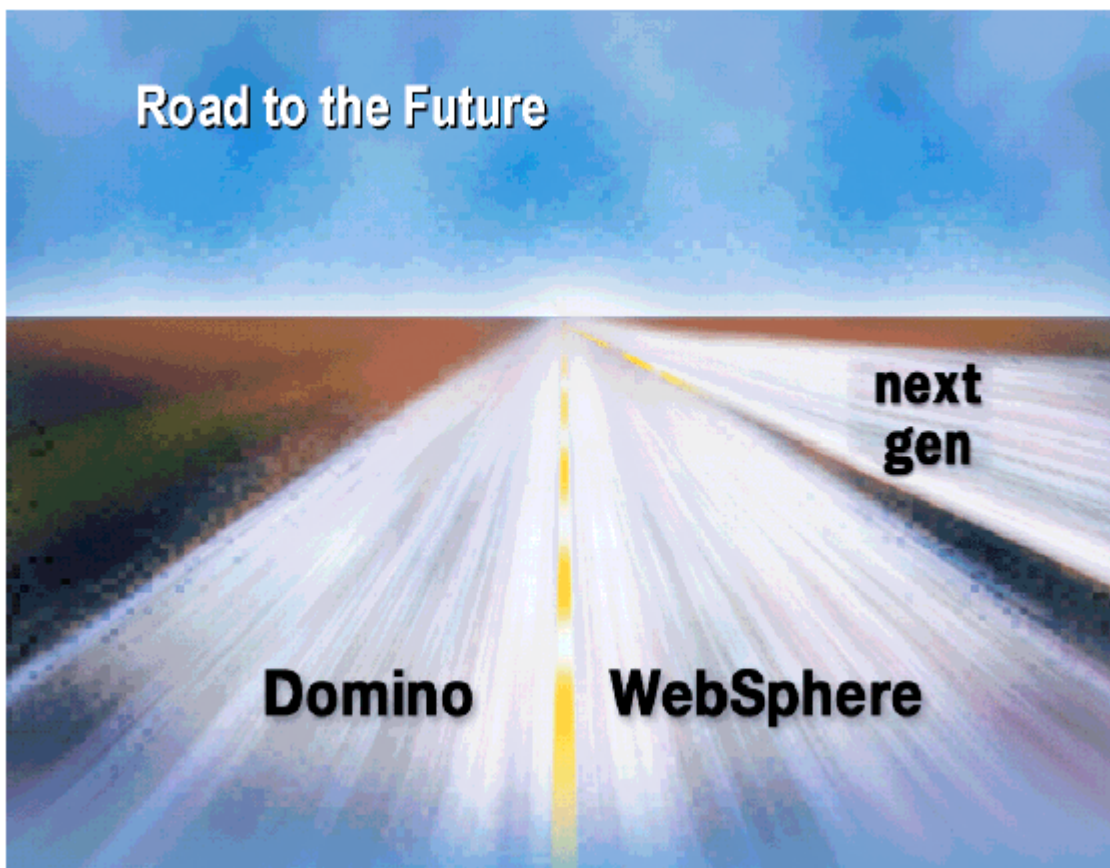
- Распространить возможности совместной работы на новые сферы деятельности
- Еще более повысить продуктивность работы служащих
- Дать возможность территориально рассредоточенным группам эффективно работать в контексте делового процесса

IBM полагает, что появится новый принцип работы – “контекстно-зависимое сотрудничество” (“contextual collaboration”), когда служащие сотрудничают друг с другом “в контексте” деловых процессов. Вместо использования отдельных, обособленных инструментальных средств для сотрудничества, пользователи всегда и везде будут иметь возможности организации совместной работы — будь это приложение портала, приложение Windows, Web-приложение или мобильное коммуникационное устройство.

Для претворения в жизнь своих технических принципов IBM Lotus предоставит модульные сервисы для организации совместной работы, что позволит продуктам Lotus легче интегрироваться друг с другом и с приложениями других производителей или собственной разработки заказчика. Технически это будет выполнено в виде набора сервис-ориентированных интерфейсов для всех серверных продуктов Lotus, включая Lotus Domino. На сегодняшний день API программных продуктов Lotus широко различаются от продукта к продукту, что делает их интеграцию довольно затруднительной. Lotus упростит интеграцию за счет введения общих принципов в разных продуктах и моделях данных, чтобы сделать API разных продуктов более однородными. Благодаря этим API вашей организации станет проще формировать интегрированные из разных программных продуктов приложения. Сначала Lotus выпустит Java-интерфейсы, но будет работать и над Web-сервисами по мере становления этих технологий более зрелыми.

Это не отразится на имеющихся у вас приложениях собственной разработки, сформированных с использованием продуктов Lotus, что послужит стимулом для дальнейшего усиления ваших инвестиций в приложения. В будущем ваши приложения смогут использовать преимущества следующего поколения сервисов для совместной работы. Например, сервисы по приему входящих сообщений Domino Inbox или сервисы-планировщики работы (календари) будут открыты для интеграции в любое приложение, которое может использовать Web-сервисы, так что ваши разработчики легко смогут встроить их функции во множество бизнес-приложений разного типа.

Для лучшего понимания новой стратегии и ее применения к приложениям Domino и их разработке вообразите многополосную автомагистраль, на которой есть полоса Lotus Domino, полоса WebSphere и полоса продуктов следующего поколения. Полоса Lotus Domino тянется вдаль, насколько может видеть глаз, и идет параллельно полосе WebSphere, которая тоже тянется вдаль, насколько может видеть глаз. Где-то вдали, в будущем, полоса продуктов следующего поколения сливается воедино с этими двумя полосами.



На магистральной полосе Lotus Domino среда Domino для совместной разработки и развертывания приложений дает вам возможность быстро разрабатывать приложения и делать их автономными, собирающими вместе людей, процессы и данные, чтобы улучшить как продуктивность е-бизнеса, так и способность к быстрому принятию решений. Lotus и в будущем сохранит текущую модель разработки приложений Domino и хранения данных (NSF), усовершенствовав ее в соответствии с нуждами разработчиков и заказчиков. По традиции IBM, заказчики Lotus Notes и Domino продолжают получать всестороннюю поддержку во всем обозримом будущем. Кроме того, Lotus Domino увеличит поддержку платформы Java 2, Enterprise Edition (J2EE) и стандартных инфраструктур, таких как теги Java Server Page (JSP), Java-интерфейсы прикладного программирования, LDAP и RDB-интеграцию, чтобы помочь разработчикам, заинтересованным в работе на обеих полосах магистрали – Domino и WebSphere.

На магистральной полосе WebSphere в качестве платформы для разработки приложений усиливается спецификация J2EE. J2EE предоставляет особую архитектуру для формирования, развертывания и управления приложениями на разных уровнях, часто вторгаясь в области представления, логики и данных. Эта архитектура нацелена на обеспечение масштабируемости, гибкости и управляемости. Хотя J2EE – могучая платформа разработки приложений, она небогата функциями для поддержки совместной работы, поэтому она использует для этих целей возможности Lotus Domino. Приложения, разработанные для использования магистральных полос Lotus Domino и WebSphere, сочетают мощные возможности по организации совместной работы со значительной транзакционной масштабируемостью, обеспечивая сквозные решения для е-бизнеса.

В ближайшем будущем третья магистральная полоса – следующее поколение сервисов для совместной работы – сольется с полосами Lotus Domino и WebSphere. Магистральная полоса продуктов следующего поколения предоставит сообществам разработчиков Lotus Domino и WebSphere модульную архитектуру для формирования интегрированных решений по организации совместных работ. Для усиления и совершенствования своих приложений разработчики Domino смогут использовать технологии обеих магистральных полос – WebSphere, а затем и сервисов следующего поколения. Разработчики WebSphere и J2EE через Web-сервисы смогут интегрировать возможности для совместной работы из Lotus Domino и других продуктов Lotus. В обоих случаях интеграция решений по организации совместной работы станет проще и не потребует от разработчиков специальных навыков.

Задача Lotus – разработать программные продукты, которые не только принесут что-то новое, но станут проще в использовании и дешевле. Именно в этом и заключается суть стратегии — в новой архитектуре организации совместных работ следующего поколения, чтобы предоставить решения по сотрудничеству, которые будут интегрироваться в любые области деятельности предприятия, повышая производительность и способность к взаимодействию во всех сферах вашего бизнеса.

## **Значение приложений Domino и модели разработки приложений**

### **Значение приложений Domino**

Lotus Domino – всесторонняя прикладная платформа для организации совместной работы, которая поддерживает обработку данных и приложений, как в интерактивном режиме, так и при автономной работе. Многие заказчики, прежде всего, покупают Lotus Domino для обеспечения предприятия системой поддержки электронной почты и календарного планирования, делая эти типы приложений наиболее широко развернутыми приложениями для сотрудничества. Однако многие заказчики используют и другие возможности, помимо почтовых, которые поддерживают процессы основного бизнеса и дают возможность служащим эффективно и надежно взаимодействовать друг с другом. Lotus Domino – всесторонняя платформа, так как обеспечивает полноценную инфраструктуру, необходимую для создания, тестирования, развертывания и управления распределенных, многоязычных приложений, включая каталоги, базы данных, сервер приложений, администрирование, системы безопасности, связь, Web-сервер, почтовый сервер, механизм планирования и так далее — полный комплекс функций.

Разработчики Domino могут проектировать приложения для клиента Lotus Notes, Web-браузеров, мобильных телефонов и других карманных устройств, а чаще всего – для гибридной среды, доступной для разнородных клиентов. Гибридные клиентские приложения Domino позволят в полной мере раскрыть возможности сервисов репликации и автономной работы для создания надежных синхронизированных приложений, которые работают как в автономном режиме, так и в режиме связи с сервером. Функция репликации дает пользователям возможность сохранять в локальной файловой системе копию приложения Domino и его данных и периодически синхронизировать данные, чтобы пользователи могли работать производительнее и эффективнее даже при отключении от сети. Если сервисы автономной работы Domino (Domino Off-Line Services – DOLS) добавлены к клиенту браузера, то пользователи смогут без потери эффективности работать с Web-приложениями Domino и в автономном режиме.

Примерами решений Domino могут служить централизованное управление документооборотом и потоками операций, в частности групповые офисы (teamrooms) проекта, архивы документов, дискуссионные форумы, расширение возможностей торгового штата и приложения для самообслуживания сотрудников. Приложения Domino принесут пользу предпринимателям любого уровня.

Военно-морской флот Соединенных Штатов использует Lotus Domino для надежного взаимодействия между судами во флотских подразделениях. Проект “Сотрудничество на море” (“Collaboration @ Sea”), реализованный на Военно-Морском Флоте за каких-то пару месяцев, раскрывает возможности доступа через Web-браузер, а также функции хранения дублированных и распределенных данных, обеспечивая всех участников синхронизированными данными на случай проблем с пропускной способностью сети.

Приложения Domino были также за считанные часы сформированы для города Нью-Йорк в период трагических событий 11 сентября. Они помогали управлять распределенным архивом о состоянии организаций и пропавших людях в эпицентре событий. Открытая платформа Lotus Domino давала доступ к собранному данным множеству служб, работающих на месте трагедии. Lotus Domino – зрелый и широко развернутый продукт для разработки и развертывания приложений, следующей будет его шестая версия.

### **Значение разработки приложений Domino**

Как всесторонняя прикладная платформа, Lotus Domino включает инструмент для быстрой разработки приложений (RAD), документо-ориентированную объектную модель и широкую поддержку языков программирования для формирования собственных приложений для совместной работы. Благодаря такому выбору, ваша организация сможет усилить навыки своих разработчиков в разработке приложений Domino. В частности, при необходимости создания приложения для сотрудничества типа дискуссионного форума или библиотеки документов, вы, при минимальных навыках программирования, легко сможете создать такое приложение Domino на основе готового шаблона без необходимости написания специальной программы.

Если вам понадобится добавить новую операцию в ориентированный на задачи процесс документооборота, вы можете ввести в первоначальный шаблон новые макрокоманды и формулы. Если потребуется добавить условную логику для отправки уведомлений в соответствии с содержанием документа или типом данных, разработчик с базовым уровнем навыков программирования также может модифицировать приложение. Для самостоятельного формирования типичного приложения Domino в большинстве случаев достаточно самых основных навыков программирования.

Для более сложных решений разработчики могут использовать Java, Microsoft COM, C / C ++ или CORBA. Наличие различных интерфейсов Lotus Domino для единой объектной модели дает разработчику возможность выбрать для поставленной задачи наиболее удобный для него язык и использовать собственные навыки в новых приложениях и решениях.

Некоторые решения требуют данных, не поддерживаемых Notes, и поддержки нескольких национальных языков. Возможности Lotus Domino в области быстрой разработки приложений облегчают создание таких решений. Используя визуальные методы сопоставления данных, разработчик может легко и быстро интегрировать реляционные данные с данными Notes. Для применения технологии программных соединителей Lotus Connectors не требуется никакого программирования. Приложения Domino, которые нужно развертывать в различных регионах земного шара, можно снабдить всеми элементами, переведенными на ряд языков с помощью нескольких довольно простых форм. Все эти функции RAD имеются в интегрированной среде разработки Domino Designer.

Особенности разработки приложений Domino в версии Domino 6 могут дать выигрыш вашей организации за счет многократного использования программных кодов и модулей, упрощения процесса совместной разработки и за счет общего усовершенствования и расширения функциональности по сравнению с версией 5 – все это поможет повысить производительность разработчика. В число главных новых особенностей входят:

- Управление возможностями XML (синтаксический анализатор DOM, синтаксический анализатор SAX, XSLT-преобразователь, утилиты импорта и экспорта XML)
- Программная поддержка формата RTF
- Интерфейс JSP-тегов
- Функции совместной разработки, например, модуль блокировки элементов проектирования
- Более тесная интеграция с внешними системами, такими как реляционные базы данных
- Инструментальные средства для динамического формирования Web-страниц

Как видите, Lotus продолжает совершенствовать модель приложений Lotus Notes и Domino, вводя новые возможности, в которых нуждаются заказчики. Вы можете быть уверены, что в готовящихся к выпуску новых версиях Lotus Notes и Domino компания Lotus останется верна своей стратегии, основанной на платформе с открытыми стандартами, добавляя функции для разработки мобильных/беспроводных приложений, упрощая пользование Web-сервисами и расширяя поддержку XML.

Направление развития Domino в области разработки приложений базируется на фундаментальной предпосылке сделать Lotus Domino гибкой и открытой платформой, о чем свидетельствует широкая поддержка XML и языков программирования. Гибкость и открытость служат ключами для способности приложений Domino усилить J2EE, сливаясь с другой, упомянутой ранее, полосой на магистрали приложений. Вы можете расширить вклад ваших приложений Domino в данные и логику приложений, реализуя соответствующие приложения и данные с помощью Web-сервисов (через LotusScript или Java) или тегов JSP для интеграции с приложениями следующего поколения или приложениями WebSphere.

Наряду с прогрессивными новшествами, которые Lotus добавит к Domino Designer и к модели программирования Domino, продолжится интеграция с WebSphere Studio. Этот процесс не придет на смену Domino Designer, а облегчит работу группам разработчиков, использующих в их среде серверы приложений и Domino, и WebSphere. Такие группы извлекут выгоду, используя при формировании приложений могущество обеих систем. Дополнительную информацию о будущем разработке приложений и точках интеграции Domino Designer с WebSphere Studio см. в разделе “Будущее быстрой разработки приложений”, далее в этом документе.

## Приложения Domino и Web-сервисы

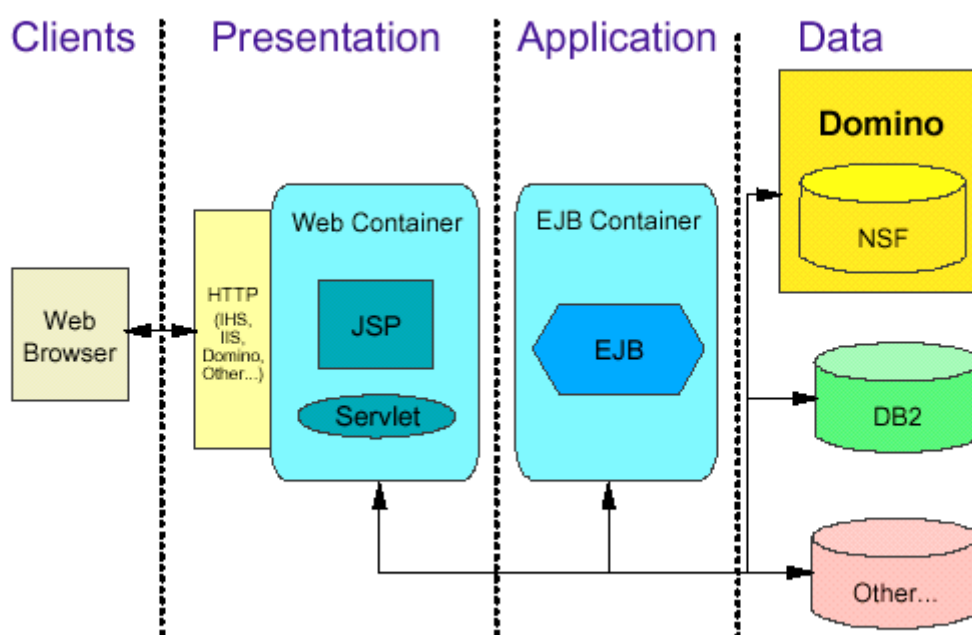
Web-сервисы – это набор стандартов, находящихся на стадии становления, который упрощает интеграцию приложений, обеспечивая стандартизированный протокол доступа, известный как простой протокол доступа к объектам (Simple Object Access Protocol или SOAP). Разработчики могут выразить интерфейс приложения средствами XML. Сейчас разработчики Domino могут на любой платформе добавлять интерфейсы Web-сервисов к существующим приложениям Domino версии 5 с помощью примерно 35 строк программного кода LotusScript (BASIC) или Java. Единственными инструментами, необходимыми для использования Web-сервисов в ваших приложениях Domino, являются сервер Lotus Domino и Domino Designer. Примеры можно найти по адресу <http://advisor.com/Articles.nsf/aid/DEVEG02>, а набор инструментальных средств можно загрузить со страницы <http://www.alphaworks.ibm.com>.

В Lotus Domino можно использовать основанные на Java Web-сервисы, которые отразят функциональные возможности и данные Domino. Используя комбинацию J2EE-сервера типа WebSphere, соответствующих SOAP-классов и Java-объектов Domino, разработчики могут представить желаемые компоненты их текущих приложений Domino в виде Web-сервисов. Это требует некоторых навыков Java-разработки и знания объектной модели Domino. Сервер Приложений IBM WebSphere Studio поставляется с необходимыми SOAP-классами, а среда разработки WebSphere Studio снабжена мастерами для создания, использования, управления и развертывания Web-сервисов. Это поможет повысить производительность разработчиков и ориентировать их на реализацию значительной части приложений Domino в форме Web-сервисов.

## Значение приложений J2EE и модели разработки приложений

Обычно приложения J2EE используют многоуровневую модель приложения. Основные элементы приложения подразделяются на отдельные компоненты, которые выполняют свои конкретные функции в среде, отдельной от других элементов. Простейшая многоуровневая модель разбита на логику представления, логику приложения и размещение данных. Каждый из этих уровней выполняет конкретную функцию приложения и может размещаться на одной или нескольких системах. Для извлечения максимальных выгод предприятие может развертывать элементы приложения на любом типе устройств и в любом территориальном подразделении. Такая многоуровневая модель обеспечивает замечательную гибкость при формировании, развертывании и многократном использовании приложений. Основанный на этом типе архитектуры уровень, отвечающий за логику представления информации, может быть территориально расположен вблизи пользователей, на недорогих устройствах; а уровень логики приложений – в центральном офисе, на более мощной, дорогой системе. Данные стоит размещать на мэйнфрейме, который бы работал десятилетиями. Этот пример – довольно экстремальное, но возможное использование уровней J2EE-модели приложения. Еще одно удобство этой модели заключается в возможности многократного использования элементов приложения, которые можно легко приспособить для пользы дела в другой сфере бизнеса.

Наглядным примером может служить инвентаризационное приложение, сформированное на основе этой модели, которое позволяет выяснить количество единиц товара на складе и сколько товара нужно произвести, согласно клиентам также нуждается в приложении, которое поможет ему знать, сколько единиц товара находится на складе, и каковы у производителей планы выпуска этого товара. Вместо создания нового торгового приложения, которое дублирует функции производственного приложения, можно воспользоваться готовой логикой производственного приложения и ввести ее в новое торговое приложение.



Клиенты	Представление		Приложение	Данные
Web-браузер	HTTP (IIS, IIS, Domino, другое...)	Web Container JSP Servlet	EJB Container EJB	Domino NSF DB2 Другое



## **J2EE-серверы приложений**

Поскольку организации вкладывают значительные капиталы в платформу J2EE-сервера, выбор наиболее подходящего сервера для их конкретной среды может показаться трудной задачей. Тем не менее, хотя выбор велик, основные функциональные возможности всех J2EE-серверов более или менее одинаковы. Эти основные элементы модельного J2EE-сервера определены в спецификации J2EE и обеспечивают широкий круг сервисов, которые будут предоставлены и поддержаны для приложений, выполняющихся на этом сервере. Сервер поддерживает работу различных элементов приложения, снабжая их свежими контекстными данными, аутентификационной информацией текущего пользователя, транзакционным контекстом для высоко чувствительных приложений и так далее.

Для аспектов, не определенных в спецификации J2EE, поставщики J2EE-серверов могут предоставить расширения, которые делают один сервер более привлекательным или приспособляемым к конкретной среде заказчика. Это позволяет поставщикам серверов вводить дополнительные ценные функциональные возможности, не принося в жертву обязательные стандарты, которым должны отвечать серверы. Таким образом, базовые модели серверов от различных поставщиков эквивалентны и заказчики могут укреплять навыки администраторов и разработчиков с помощью J2EE-серверов от любых поставщиков.

## **Domino и J2EE**

Одно из основных различий между J2EE-сервером приложений и сервером приложений Domino в том, что Lotus Domino обеспечивает полностью интегрированную среду. Он поддерживает почти все аспекты среды приложения – от выполнения приложения до аутентификации пользователя, обслуживания каталогов, размещения данных и их отображения – все в одной системе.

Модель J2EE также имеет элементы для обеспечения приложения той же самой информацией, но J2EE-сервер отвечает за выполнение не всех задач, которые способен выполнять сервер Domino. Сервер J2EE обращается к различным компонентам среды заказчика – для выполнения запросов на данные, информацию по каталогам и так далее. Например, поскольку в спецификации J2EE не определено, как сервер может доставить данные из хранилища данных в приложение, использующее JDBC, спецификация J2EE не требует, чтобы сам сервер содержал диспетчер баз данных.

## **Domino дает приложениям возможность интеграции с J2EE**

Можно получить много преимуществ, сочетая приложения Domino и J2EE, особенно за счет добавления возможности вмешательства человека в масштабируемые и транзакционные приложения на базе J2EE. Существует два основных способа обращения к Lotus Domino при интеграции Lotus Domino с основанными на J2EE приложениями:

- Объекты Domino для Java
- Специальные JSP-теги Domino 6

По сути, объекты Domino для Java – это Domino Java API. Эти объекты предусмотрены для основанного на объектах доступа в хранилище данных для ключевых процедур базы данных: create/read/update/delete, которые чаще называют CRUD-процедурами. Этот API дает доступ не только к данным Domino, но также и к некоторым главным сервисам, таким как регистрация пользователей, запуск агентов, отправка почты и другим.

Специальные JSP-теги Domino 6 предназначены для использования только внутри страниц JavaServerPage. JSP-теги – это XML-теги, вложенные в JSP и обеспечивающие доступ к данным, ввод данных и контроль за процессом. Эти теги выделяют объекты Domino для Java и обеспечивают быстрый цикл разработки J2EE-приложения, которое использует данные и сервисы Domino.

Lotus Domino обладает также широкими XML-возможностями, которыми можно воспользоваться из J2EE-приложений. Для обращения к неструктурированным документам Domino J2EE-приложение может использовать интерфейс XML; для этого не требуется знание специфики данных, сохраненных в Lotus Domino. Затем J2EE-приложение может обработать полученный XML и преобразовать его в формат, необходимый приложению. Когда приложение собирается обновить данные из Lotus Domino, можно воспользоваться тем же самым XML-интерфейсом и обновить базу данных через XML, обеспечивая таким образом обратимую передачу данных в и из Lotus Domino.

## **Интеграция Domino с сервером приложений WebSphere**

В IBM команды разработчиков Domino и WebSphere Application Server координируют свои усилия по разработке и внедрению основных точек интеграции, которые отвечают нуждам заказчиков. Каждый из

вводимых элементов создается с учетом удобства для пользователя, более простого администрирования и интеграции со всей инфраструктурой. Поскольку стандарты Java и J2EE позволяют на довольно широком уровне интегрировать разнородные приложения, группы разработчиков Lotus и WebSphere разработали мощные точки интеграции, которые есть только в связке Domino – WebSphere Application Server. Например:

- Многие заказчики выбрали для своей среды стратегический HTTP-сервер. Lotus Domino имеет встроенный HTTP-сервер, который используют многие заказчики, но заказчики хотят также, чтобы Lotus Domino был лучше интегрирован в их инфраструктуру. Новшеством в Lotus Domino 6 является возможность разместить перед Lotus Domino, в качестве основной точки входа для пользователей Web-браузера, HTTP-сервер от сторонних поставщиков. Эта возможность реализуется, благодаря набору подключаемых HTTP-программ, которые были разработаны командой сервера приложений WebSphere, и это позволяет использовать любые из основных HTTP-серверов. Когда Lotus Domino 6 станет повсеместно доступным, будет введена начальная поддержка для Microsoft Internet Information Server (IIS) на платформе Windows NT 4.0 SP6 и для Windows 2000 Advanced Server, а также для IBM HTTP Server (IHS) на AIX. Поддержка дополнительных HTTP-серверов, включая Apache, iPlanet и Domino на различных операционных системах, будет добавлена к будущим эксплуатационным выпускам.
- Для заказчиков, стремящихся стандартизировать каталоги отдельных пользователей, Lotus Domino может предоставить службу каталогов Domino Directory другим приложениям, использующим облегченный протокол службы каталогов (LDAP). LDAP быстро становится де-факто-стандартом для службы каталогов и используется также сервером приложений WebSphere для аутентификации пользователей. Поскольку Lotus Domino может служить LDAP-сервером, WebSphere может использовать эту службу, давая заказчикам возможность из одного места управлять идентификаторами пользователей для Lotus Domino и для WebSphere, упрощая задачи и необходимый уровень навыков администрирования в среде заказчика.
- В случаях, когда сервер приложений WebSphere и Lotus Domino используют один и тот же каталог LDAP (не ограничиваясь LDAP-возможностями только Domino), заказчики могут установить единую регистрацию (Single Sign-On – SSO) для Lotus Domino и WebSphere. SSO позволяет пользователю браузера ввести свои идентификационные данные только один раз и хранить их в безопасном месте, доступном и для WebSphere, и для Lotus Domino. При включенной функции SSO пользователи, переключающиеся между приложениями WebSphere и Domino не должны многократно проходить аутентификацию, как им пришлось бы при заблокированной функции SSO.
- Кроме того, объекты Domino для Java (Java-API Domino) предусматривают безопасную обработку данных в Lotus Domino, благодаря использованию маркеров доступа SSO для формирования сеанса в Lotus Domino. Разработчик, как пользователь текущего Web-приложения, может использовать маркеры доступа SSO прямо из сервера приложений WebSphere, запрограммировав защищенный сеанс Lotus Domino. При этом приложению не требуется знать имя и пароль конкретного пользователя при доступе к Lotus Domino, и к тому же уменьшается риск, связанный с необходимостью поддержки паролей пользователей каждым отдельным приложением.

## Выбор магистрального направления

Цель данного раздела – служить оценочным инструментом, помогающим вам понять, какие факторы следует принимать во внимание при планировании интегрированных решений на платформах IBM-сервера. Раздел разбит на подразделы, соответствующие разным областям функциональности. Для каждой области обсуждаются вопросы, которые помогут вам понять, какие факторы определяют наиболее подходящую для конкретного решения архитектуру — иными словами, какие компоненты приложения должны быть разработаны для каждой из “магистральных полос”.

Читая этот раздел, держите в памяти конкретное приложение (имеющееся или запланированное) и отвечайте на каждый вопрос, применительно к этому приложению.

### 1. Цель приложения и его использование

#### *Контент приложения*

**Будет ли приложение включать редактирование, создание и воспроизведение контента в расширенном формате?**

Если основное назначение приложения приходится на операции с форматированным текстом – создание/сохранение/обновление/удаление – или если приложение должно отображать, сохранять и обрабатывать документы Microsoft Office, то это приложение точно укладывается в магистральную

полосу Domino. Кроме того, для приложения, вероятно, лучше всего подойдет клиент Lotus Notes, поскольку он имеет встроенный текстовый редактор и хорошо интегрируется с Microsoft Office.

### **Обработка транзакций**

#### **Предназначено ли приложение для выполнения транзакционных задач?**

Транзакционное приложение – это приложение, в котором ряд шагов отслеживается, чтобы в случае необходимости процесс можно было полностью обратить. Классическим примером является банковская система, где кредит счета должен быть отменен, если предшествующий дебетовый шаг не был завершен. Транзакционная обработка хорошо укладывается в магистральную полосу WebSphere с привлечением приложений Domino, обеспечивающих коллективные сервисы для вмешательства людей, необходимого при разрешении исключений, которые встречаются в деловых транзакциях.

### **Операции, основанные на заполнении форм пользователями**

#### **Должно ли ваше приложение для завершения операции после ряда predetermined шагов выдавать пользователям электронные формы?**

Примером может служить приложение "справочный стол", где служащий заполняет форму, которая по представлении направляется электронной почтой соответствующему профессионалу "справочного стола". После решения проблемы заполненная форма возвращается обратно заполнявшему для подтверждения. После этого форма регистрируется в базе данных "справочного стола". Решения такого типа довольно просто формировать и развертывать на Lotus Domino, благодаря встроенному API электронной почты и гибким идентификаторам на основе ролей и управлению доступом. Еще лучше интегрировать Lotus Domino и WebSphere IBM вместе с MQSeries, используя Lotus Domino для операций, основанных на заполнении форм пользователями, которые встречаются в процессах транзакционной обработки, лучше всего поддерживаемых WebSphere в сочетании с MQSeries. Такие сквозные решения, которые сочетают системные транзакции с чисто человеческими задачами – замечательная возможность, которую не предлагает больше никакой другой производитель.

### **Автономное (внесетевое использование)**

#### **Должно ли приложение полнофункционально использоваться в компьютере, который отключен от сети?**

Полнофункциональное использование приложения (application fidelity) подразумевает, что функции приложения при работе в интерактивном и автономном режимах остаются теми же самыми, включая создание/редактирование/обновление/удаление данных. Это одна из уникальных особенностей Lotus Domino. Только благодаря функции репликации Domino можно автономно работать с приложением – полнофункционально и с доступом к данным из Web-браузера или клиента Lotus Notes. Когда пользователь подключится к сети, работа, которую он проделал, полностью синхронизируется с сервером. Это еще одна особенность, которая имеется только в Lotus Domino.

### **Доступ к клиенту Notes**

Lotus Notes обеспечивает настольную среду для большинства задач пользователей в приложениях для сотрудничества – таких как электронная почта, календарное планирование, управление персональной информацией – и в приложениях для сотрудничества собственной разработки. Lotus Notes предоставляет оконный графический интерфейс пользователя и наращиваемую структуру клиента для выполнения приложений для совместной работы и приложений, ориентированных на работу с документами. Ниже представлен перечень особенностей клиента Lotus Notes, которые отличают его от Web-браузера:

- Расширенный формат редактирования контента (таблицы, проверка правописания, вставная графика, полная поддержка шрифтов, поиск/замена, и т.д.)
- Интеграция с Microsoft Office для прикрепления, сохранения и распределения файлов
- В автономном режиме – функции электронной почты, календарного планирования, персональной информационной системы (PIM) и приложения с фактически полной функциональностью вне сети и синхронизацией данных при подключении
- Шифрование данных с особыми мерами безопасности, шифрование при передаче по проводной сети и цифровые подписи, не имеющиеся в среде браузера

## **Управление доступом**

### **Нужно ли пользователям делегировать свои права?**

Для некоторых приложений требуется, чтобы пользователь имел возможность предоставить свои права доступа другому пользователю. Чаще всего это используется, чтобы один пользователь мог действовать от имени другого, но с возможностью отследить действия каждого в отдельности. Классический пример – администратор, делегирующий свои обязанности помощнику. Помощник должен получить не права администратора, а должен получить право действовать *от имени* администратора. Разница в том, что на случай необходимости проверки любые действия помощника: создать/сохранить/обновить/удалить – отменяются особо от аналогичных действий администратора. Эта функция зависит от собственного RDB-управления доступом и поэтому соответствует магистральной полосе WebSphere.

### **Должен ли доступ пользователя быть ограничен подмножеством записей в таблице или отдельными полями формы?**

Например, менеджеру предоставляется доступ к таблице с заработной платой служащих, но он может видеть только записи, относящиеся к служащим, которыми он руководит. Аналогично, пользователь, заполняющий форму, может видеть только некоторые поля, в то время как администратор, использующий ту же самую форму, имеет расширенный образец с показанными дополнительными полями. Lotus Domino имеет уникальную модель защиты, когда управление доступом может быть очень селективным в отношении мелких подмножеств приложений и данных. Управление доступом можно также изменить программным образом на лету, корректируя, кто что видит, на основе ряда динамических условий. Следовательно, эта особенность соответствует магистральной полосе Lotus Domino, но добавила бы ценности и приложениям WebSphere.

### **Нуждается ли приложение в динамических идентификаторах на основании ролей пользователей?**

Большинство приложений использует группы пользователей, определенные службой каталогов. Однако во многих случаях приложения используют свои особые группы. Сохранять группы для конкретного приложения в общем каталоге групп неблагоразумно с точки зрения управления, так что лучшим решением является включение в приложение пользовательских ролей и сохранение этих ролей вместе с данными приложения, добавив к нему логику для определения роли текущего пользователя.

Хороший пример приложения, которое требует определения ролей – процесс утверждения решений. Например, в приложении по найму (из разряда приложений отдела кадров), люди, которые должны одобрить предложение по заработной плате – это менеджер по кадрам, менеджер по финансам, и менеджер, который нанимает сотрудника. Все они – члены различных групп каталога, но этим группам нельзя предоставить права одобрения заработной платы, так как в них входит слишком много других людей. Новую группу для одобрения заработной платы тоже нельзя создать, потому что в каждом случае найма кадровые агенты разные. Следовательно, нужно создать особую для приложения роль, чтобы пользователи могли выступать в этой роли и отказываться от нее.

Ролевое членство меняется на лету, благодаря логике приложения, а администрирование производится на прикладном уровне, а не на уровне каталогов (системном), упрощая администрирование приложения. Для любого компонента приложения, которому требуются особые роли и/или динамически изменяющиеся роли, подойдет магистральная полоса Lotus Domino. Это еще одно преимущество интеграции Lotus Domino с приложениями WebSphere, потому что многие операции приложений по людским ресурсам (HR) основаны на J2EE.

## **2. Требования к масштабируемости**

### **Количество пользователей**

#### **Будет ли количество параллельно работающих пользователей с Web-браузерами превышать 5000?**

Есть несколько критериев, которые определяют требования к масштабируемости Web-приложения; один из них – количество параллельно работающих пользователей приложения. В общем случае, если количество одновременно активных пользователей превышает 5000 и если приложение сопряжено с Web-браузером, а не с клиентом Lotus Notes, то для него, вероятно, лучше подойдет магистральная полоса WebSphere.

Однако если приложение сопряжено с клиентом Lotus Notes, то лучше выбрать Lotus Domino. Lotus Notes использует архитектуру клиент/сервер, в которой с целью повышения масштабируемости сервера многое обрабатывается на стороне клиента.

## **Настройка инфраструктуры для повышения масштабируемости** **Должно ли ваше приложение масштабироваться горизонтально и/или вертикально?**

Правильная настройка аппаратных средств – главный фактор для соответствия необходимому уровню масштабируемости и доступности приложения. Под горизонтальной масштабируемостью подразумевается распределение приложения на несколько разных физических серверов или на несколько экземпляров сервера в «лезвийной» (“blade”) конфигурации. Вертикальная масштабируемость подразумевает обновление аппаратуры и операционной системы (HW/OS) до более масштабируемой платформы. Не забудьте проанализировать требования к масштабируемости в настоящее время и на будущее. И WebSphere, и Lotus Domino выиграют от вертикальной масштабируемости, так как оба сервера поддерживают широкий диапазон HW/OS платформ от серверов Windows на начальном уровне до IBM e Server z-Series на высшем уровне.

Отличительной особенностью WebSphere является способность к горизонтальному масштабированию. Одно из главных преимуществ WebSphere – способность конфигурировать приложения для работы в распределенной среде с несколькими серверами, которые динамически производят балансировку нагрузки в соответствии с ее изменением в ходе работы.

### **3. Требования к данным**

#### **Модель данных**

#### **Приложение нуждается в структурированной или неструктурированной модели данных?**

Существует несколько типов хранилищ данных; к этому документу имеют отношение два из них: реляционное и иерархическое. Реляционное хранилище сохраняет многомерные связи между записями, имеющими отношение друг к другу. Иерархическое хранилище основано на сравнительно простой причинно-следственной модели взаимосвязей предок/потомок. Как правило, реляционное хранилище лучше подходит для структурированных данных, где между записями существуют связи "многих со многими", например, связи между записями заказчика, счетами, заказами на товары и записями продавца.

Иерархическое хранилище лучше подходит для неструктурированных данных, в частности, документов, списков, презентаций PowerPoint и других видов информации в "вольном стиле". Еще одна особенность иерархических данных – возможность изменять структуру данных, например, по желанию добавлять или удалять поля без необходимости перестройки базы данных. Другими отличительными особенностями неструктурированных хранилищ являются:

- Позволяет присваивать полю несколько значений
- Позволяет одновременно редактировать записи (нет блокировки записей)
- Лучше работает при сравнительно простом (низкой сложности) поиске документов и поиске по содержанию документов
- Лучше работает с данными, которые отображаются в виде категоризированных списков или в единой таблице

Многим приложениям требуется и реляционное, и иерархическое хранение данных, так что это, возможно, самый подходящий случай для формирования приложений, которые охватывают магистральные полосы и Lotus Domino, и WebSphere. Lotus Domino имеет встроенное неструктурированное хранилище данных, а WebSphere надежно работает с СУРБД.

#### **Нужен ли приложению прямой доступ к системе записей?**

Система записей – особенность реляционной базы данных, которая обеспечивает существование одного и только одного экземпляра записи. Выполнение этого условия требуется многим приложениям, работающим в регламентированных сферах деятельности, таких как страхование, здравоохранение и фармацевтика. Для таких приложений подходит именно магистральная полоса WebSphere, поскольку функция репликации Domino противоречит требованиям системы записей.

#### **Объем данных**

#### **Превысит ли объем данных 2 Гб?**

Объем сохраняемых данных, необходимый для приложения – важный критерий при выборе его архитектуры. Есть и другие критерии, связанные с данными: например, уровень использования пользователями операций создания/записи/обновления/удаления (CRUD) и поисковых функций. Начиная с версии Domino 5 жесткого предела на размер базы данных Domino нет, но для управления большими объемами данных с высоким уровнем использования CRUD-операций лучше подойдет магистральная полоса WebSphere.

Однако общим исключением является потребность в сохранении неструктурированных данных, таких как документы, и в поддержке высоко форматированного контента и текстов в расширенном формате, для чего лучше подойдет магистральная полоса Lotus Domino. Идеальным в этом случае будет интегрированное решение. Если необходимо сохранять данные приложения в единой базе данных, то Lotus Enterprise Integrator (LEI) может синхронизировать базу данных Domino с РБД.

### ***Распределение данных***

#### **Должны ли данные приложения (без логики приложения) распределяться на серверы через WAN или Интернет?**

Во многих случаях приложение проектируется для выполнения в среде распределенных серверов, но *данные* остаются централизованными, обычно в интересах защиты целостности данных. Однако бывают ситуации, в которых данные должны распределяться, особенно, когда приложения развернуты в регионах с низкой пропускной способностью линий связи из-за неоднородной и/или дорогой сетевой инфраструктуры. Иногда распределение данных достигается с помощью кэширования распределяемых данных, но при таком подходе часто подвергается риску целостность данных, и он может оказаться очень дорогостоящим и сложным для реализации. Lotus Domino и Lotus Enterprise Integrator дают идеальное решение для распределения по сети всех типов данных, поскольку они гарантируют целостность данных, их безопасность и синхронизацию всех копий набора данных. Распределение и неструктурированных, и реляционных данных с использованием LEI служит примером слияния магистральных полос Lotus Domino и WebSphere в целях удовлетворения требований приложения.

## **4. Срок службы приложения**

### ***Стратегия против тактики***

#### **Насколько стратегично приложение? Или это тактическое решение?**

Этот вопрос ставится, чтобы вы продумали цель приложения и срок его службы. Если предполагается, что приложение будет служить пять лет или более или если приложение очень важно для поддержки основного бизнеса, то оно является стратегическим.

Тактические решения формируются для относительно краткосрочных проектов или деловых инициатив. К числу краткосрочных тактических решений можно отнести также приложения, которые нечасто используются. Например, HR-приложение для анализа производительности служащих, используемое только один раз в год – тактическое решение, так как оно редко используется и только косвенно связано с основной производственной деятельностью.

Благодаря своей среде быстрой разработки приложений, Lotus Domino хорошо подходит для краткосрочных/тактических решений, поскольку разработчики смогут формировать и развертывать приложения за довольно короткий промежуток времени. Если ваша организация развертывает такие приложения на Lotus Domino, не стоит заново перестраивать их на J2EE, так как маловероятно, что это принесет достаточный выигрыш, чтобы покрыть расходы на разработку.

С другой стороны, стратегические приложения должны всегда проектироваться с расчетом на максимальную прибыль за время срока службы приложения. Так что тщательная оценка срока действия приложения и учет факторов, описанных в этом документе, должны помочь в определении наиболее подходящего магистрального направления для стратегических решений.

### ***Оценка времени***

#### **Должно ли ваше приложение быть разработано/модернизировано, сформировано, протестировано и развернуто в пределах 6 месяцев?**

Например, после внутренней ревизии отделу дается три месяца, чтобы привести систему управления в соответствие со стандартами ISO 9000. Времени на длительную разработку нет, но без этого решения отдел не сможет достичь требуемого соответствия. Как упоминалось ранее, одним из главных преимуществ Lotus Domino является его среда быстрой разработки приложений. Можно разработать, протестировать и развернуть приложение за недели, а не за месяцы, как на других платформах сервера

приложений. “Скорость пробега” может меняться, конечно, в зависимости от сложности приложения. Многие приложения, хорошо подходящие для Lotus Domino, формировать довольно просто, особенно, если разработчик может воспользоваться одним из стандартных системных шаблонов, которые поставляются с Lotus Domino.

## **5. Факторы среды**

### **Стандарты ИТ-инфраструктуры предприятия**

#### **Подчиняется ли приложение стратегической политике ИТ или руководству предприятия?**

Многие ИТ-организации для упрощения управления системами и экономии денег за счет консолидации серверов формируют стратегические решения в отношении приложений их инфраструктуры. Большинство компаний основывает свои стратегии на архитектуре J2EE-приложений, LDAP для службы каталогов и реляционных базах данных для хранения данных. Если архитектура приложений управляется в соответствии с такой ИТ-стратегией предприятия, то для нее подходит магистральная полоса WebSphere, с учетом, что может понадобиться и Lotus Domino для функций, еще не доступных на J2EE-платформе.

Еще имейте в виду, что Domino Directory можно сконфигурировать как каталог LDAP, так что ваша существующая служба каталогов Domino Directory будет соответствовать ИТ-политике, ИТ-инфраструктуре и поддержке ресурсов.

#### **Все ли ИТ-ресурсы имеются?**

Всегда следует принимать во внимание расходы, связанные с разработкой, развертыванием и управлением приложениями, включая ИТ-инфраструктуру (база данных, сервер приложений, служба каталогов и меры безопасности) и человеческие ресурсы для системного администрирования, администрирования базы данных, службы каталогов или архитектуры системы безопасности.

Например, рассмотрим ситуацию, когда приложение должно быть развернуто на каком-то участке в далекой стране третьего мира без адекватной локальной ИТ-поддержки или широкополосной WAN. В дополнение к ограничениям по особенностям инфраструктуры еще одним ограничительным фактором является бюджет. Например, предприятию необходимо решение, но оно не может обеспечить адекватное финансирование ИТ-ресурсов. Это только две из множества ситуаций, когда есть потребность в приложении, но нет необходимой ИТ-инфраструктуры и трудовых ресурсов.

Полностью интегрированный (как описано в разделе “Значение приложений Domino”) автономный сервер приложений, предоставляемый Lotus Domino, был бы наиболее оптимальным решением в таких ситуациях. Lotus Domino обеспечивает фактически все, что требуется вашей организации для формирования, запуска и развертывания многих типов приложений.

### **Навыки в разработке приложений**

#### **Есть ли/можно ли быстро приобрести необходимые навыки работы Lotus Notes и/или навыки в Java-разработке и администрировании?**

Для многих приложений нужны разработчики – для их создания, эксплуатационной поддержки – и системные администраторы – для управления ими. Часто архитектура приложений определяется располагаемыми в организации навыками в этих областях. Анализ ресурсов, имеющихся в настоящее время или доступных через наем или контракты – важный фактор при выборе магистрального направления.

## **Будущее систем быстрой разработки приложений (RAD)**

J2EE-архитектуры предлагают платформы с присущими им преимуществами по производительности для разработки Web-приложений вообще и приложений для совместной работы, в особенности. Многие заказчики и поставщики приложений уже получают выгоды в своих системах и решениях. Эти тенденции предвещают огромный рост использования J2EE-архитектуры. Однако для сохранения этого быстрого роста службы поддержки разработок J2EE-приложений должны предоставить атрибуты, которые привлекут более широкую аудиторию разработчиков.

Должны быть развиты инструментальные средства разработки J2EE-приложений, чтобы привлечь в ИТ-отделы сообщества разработчиков для разработки Web-приложений в подразделениях предприятия, помимо традиционных Java-разработчиков (которые преимущественно сосредотачиваются на системном программировании и программировании приложений для предприятия). Развитие требует

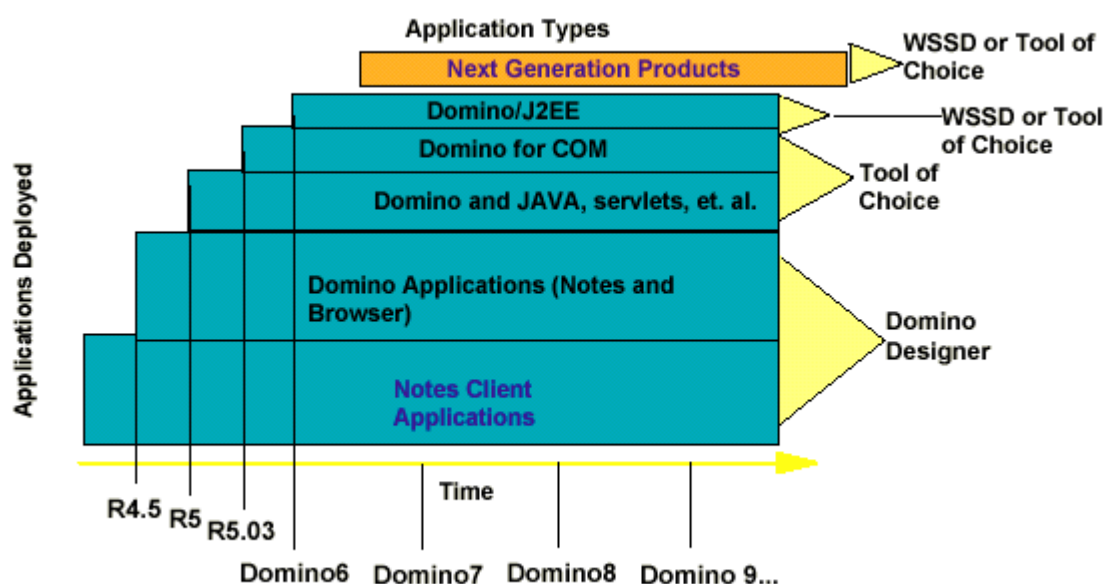
создания инструментария быстрой разработки приложений (RAD), который позволит разработчикам Web-приложений формировать и подключать компоненты, данные и логику бизнес-приложений, не вдаваясь в сложности J2EE-архитектуры.

Наряду с опытом, приобретенным при совершенствовании Domino Designer, Lotus уделяет повышенное внимание нуждам пользователей инструментов RAD, чтобы сформировать первый RAD-инструментарий J2EE, прежде всего ориентированный на J2EE-платформы IBM. В настоящее время этот проект находится в стадии реализации и является совместной разработкой технических групп Lotus e-Workplaces и IBM WebSphere.

Целевая аудитория проекта – корпоративные разработчики Web-приложений. Корпоративные разработчики Web-приложений, в отличие от системных программистов или разработчиков предприятия, обычно ориентируются на навыки пользователей, которые связаны с их поведением (то есть, с выполнением определенных функций). Для этих разработчиков приложений наряду с выполняемыми функциями важен вид и ход работы. Обычно эти разработчики начинают с интерфейса пользователя, так как сначала продумывают приложение именно в плане его внешнего облика. Затем к пользовательскому интерфейсу UI добавляются функции, а страницы связываются вместе, формируя приложение.

Будущий RAD-инструментарий обеспечит простой визуальный способ разработки мощных Web-приложений, которые укрепят J2EE-среду. Важно отметить однако, что хотя RAD-инструментарий поддержит легкое создание приложений, он не будет ограничен формированием лишь простых приложений — он будет полезен и разработчикам, которым нравится работать независимо, обеспечивая им быстрый способ формирования основы приложения, и позволяя затем окончательную настройку через доступ ко всем деталям приложения с помощью инструментальных средств WebSphere Studio. Наконец, будущий RAD-инструментарий будет всесторонним инструментальным средством RAD, охватывающим все фазы разработки приложений: программирование, тестирование и развертывание. Этот будущий RAD-инструментарий от IBM базируется на структуре с открытыми кодами Eclipse, и усиливает инструментальные средства, имеющиеся в продуктах семейства IBM WebSphere Studio, в плане многих ключевых функций, таких как дизайн страниц, редактирование сценариев, отладка, создание bean-компонентов, и так далее. Lotus продолжит совершенствовать Domino Designer как гибридную клиентскую среду разработки приложений для Lotus Domino, добавив со временем утилиты для облегчения взаимодействия с инструментальными средствами WebSphere.

На следующей диаграмме показаны все поддерживаемые Lotus в настоящее время и в будущем модели разработки приложений, с соответствующими инструментальными средствами разработки приложений для каждой:





Развернутые приложения	Типы приложений	
	Продукты следующего поколения	WSSD* или инструменты по выбору
	Domino/J2EE	WSSD или инструменты по выбору
	Domino для COM	Инструменты по выбору
	Domino и JAVA, сервлеты, и т.п..	
	Приложения Domino (Notes и Browser)	Domino Designer
	Приложения клиента Notes	

\* WSSD: WebSphere Studio Site Developer

В настоящее время приложения на платформе Lotus Notes в основной массе представляют собой приложения собственной разработки, развернутые для программных продуктов Lotus. Это приложения, созданные прежде всего с помощью LotusScript и языка формул Lotus Notes. Следующей превалирующей моделью развернутых приложений являются приложения Domino, развернутые на Web-браузерах или гибридные приложения для браузера/Lotus Notes. В обозримом будущем, пока заказчики используют эти модели разработки приложений, Lotus продолжит их поддержку.

Domino Designer – инструмент разработки приложений для формирования приложений Lotus Notes и Domino. Будущее Domino Designer – непрерывное совершенствование, особенно ориентированное на функции, которые позволяют разработку интегрированных приложений и использование сервисов для совместной работы следующего поколения. Нет планов объединения Domino Designer с WebSphere Studio или Eclipse — он останется отдельным RAD-инструментом для Lotus Notes и Domino. В Lotus Notes и версии Domino 5 усовершенствована модель разработки добавлением новых API для Java и COM, которые важны для взаимодействия с внешними приложениями. Главная особенность Lotus Domino 6 – способность к формированию интегрированных приложений Domino/J2EE, использующих теги страниц Domino JavaServer (JSP) и расширенные Java-классы.

Эта модель разработки приложений – переходный мостик к будущему и дает заказчикам два основных преимущества:

- Функциональные возможности Lotus Domino могут быть интегрированы в J2EE-приложения
- Сегодняшние разработчики Domino получают довольно простой способ начать работать с J2EE без необходимости предварительного приобретения специальных навыков.

Lotus планирует выпустить для WebSphere Studio 5 подключаемую программу, которая поможет разработчикам, желающим интегрировать функциональные возможности приложений WebSphere в приложения Domino. Эта подключаемая программа даст разработчику возможность видеть в WebSphere Studio модули базы данных Domino и запускать Domino Designer для модификации элемента, а также возможность копировать базовую структуру формы или представления в JSP, используя теги JSP, доступные в Lotus Domino 6.

Как сказано во вводном разделе, будущая стратегия нацелена на новую модель разработки приложений, которая обеспечит соответствующие ориентированные на сервисы интерфейсы для всех продуктов Lotus на базе стандартов Java и Web-сервисов. Новый RAD-инструментарий для J2EE станет лучшим способом интеграции сервисов Lotus для совместной работы с приложениями J2EE.

## Заключение: Подготовка к будущему

Чтобы подготовиться к получению максимальной пользы от Domino по мере того, как растет его ориентация на J2EE (то есть, к магистральной полосе следующего поколения, когда она сольется с полосами Lotus Domino и WebSphere), советуем следующее:

- Приобретайте навыки в области технологий JavaServer Pages, Java и Web-сервисов, – особенно XML, JSP и Java существенны для создания ценных приложений, которые сегодня колеблются между магистральными направлениями Lotus Domino и WebSphere. Загрузите SoapConnect с Web-сайта <http://www.alphaworks.ibm.com> и учитесь встраивать Web-сервисы в имеющиеся приложения Domino. В будущем они станут основой технологий для интерфейсов приложений следующего поколения, так что сейчас – время приобрести навыки в этих областях.

- Перейдите на службу каталогов LDAP. Если у вас есть служба каталогов Domino Directory, версии Domino 5 или 6, тогда переход на LDAP – вопрос настройки. В будущем Lotus software будет все более и более использовать LDAP для службы каталогов, так что это будет полезный переход.
- Обновите Lotus Domino до версии 6, которая предоставляет возможности переходного периода. Lotus Domino 6 имеет множество особенностей, которые дают прекрасные возможности интеграции приложений с WebSphere. В качестве удобного в работе интерфейса для Java-объектов Domino Lotus Domino 6 предоставляет новую библиотеку JSP-тегов. Интерфейс Java-API Domino был усовершенствован для вынесения его в среду сервера приложений WebSphere. В Lotus Domino 6 также расширена поддержка LDAP, введены свои Java-классы XML, опционально предлагается интеграция с IBM Tivoli для управления системами и более тесной RDB-интеграции с помощью LEI.
- Реорганизуите группы разработчиков, чтобы разработчики Domino и WebSphere вместе работали над созданием интегрированных приложений. Разработчики Domino обладают уникальными навыками и опытом создания приложений для совместной работы, ориентированных на пользователя. Разработчики WebSphere имеют опыт в системном программировании, который очень пригодится для расширения приложений Domino с целью более широкого развертывания вне предприятия. За счет организационного объединения эти группы усилят друг друга, что поможет шире вводить приложения для сотрудничества – в разные области бизнеса и за корпоративные пределы. Это относится также к любому заказчику, использующему J2EE-контейнеры, отличные от WebSphere.
- При планировании проекта помните, что во все типы будущих решений могут быть интегрированы возможности для организации совместной работы для повышения продуктивности пользователей в большем количестве сфер бизнеса, чем это выполнимо сегодня.

IBM продолжает выпускать дополнительную документацию, такую, как информационные отчеты White Paper и книги IBM Redbook, чтобы держать вас в курсе технологических новостей. Для дополнительной информации по теме разработки приложений и интеграции между Lotus Domino и сервером приложений WebSphere см. следующие книги серии IBM Redbook:

SG24-5955-01: *Domino and WebSphere Together* Second Edition

SG24-5331-01: *Lotus Domino Release 5.0: A Developer's Handbook*

Книги IBM Redbook доступны по адресу <http://www.redbooks.ibm.com>