

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

Излагается общая концепция построения объектно-ориентированной среды разработки приложений. Анализируются основные принципы реализации подобных систем в мировой практике. Предлагается вариант представления логической модели предприятия в виде декомпозиции объектов разрабатываемой системы.

© Н.И. Алишов, Н.А. Назаров ,
2002

УДК681.324

Н.И. АЛИШОВ, Н.А. НАЗАРОВ

КОНЦЕПЦИЯ И АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ РЕАЛИЗАЦИИ ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ СРЕДЫ РАЗРАБОТКИ ПРИЛОЖЕНИЙ

Тенденции развития современных информационных технологий приводят к постоянному возрастанию сложности информационных систем (ИС), создаваемых в различных областях экономики. Современные крупные проекты ИС характеризуются, как правило, следующими особенностями:

- сложность описания (достаточно большое количество функций, процессов, элементов данных и сложные взаимосвязи между ними), требующая тщательного моделирования и анализа данных и процессов;
- наличие совокупности тесно взаимодействующих компонентов (подсистем), имеющих свои локальные задачи и цели функционирования (например, традиционных приложений, связанных с обработкой транзакций и решением регламентных задач, и приложений аналитической обработки, использующих нерегламентированные запросы к данным большого объема);
- отсутствие прямых аналогов, ограничивающее возможность использования каких-либо готовых проектных решений и прикладных систем;
- необходимость интеграции существующих и вновь разрабатываемых приложений;
- функционирование в неоднородной среде на нескольких аппаратных платформах;
- разобщенность и разнородность отдельных групп разработчиков по уровню квалификации и сложившимся традициям исполь-

зования тех или иных инструментальных средств;

существенная временная протяженность проекта, обусловленная, с одной стороны, ограниченными возможностями коллектива разработчиков, с другой - масштабами организации-заказчика и различной степенью готовности отдельных ее подразделений к внедрению ИС.

Для успешной реализации проекта объект проектирования (ИС) должен быть адекватно описан, построены полные и непротиворечивые функциональные и информационные модели ИС.

Накопленный к настоящему времени опыт проектирования ИС показывает, что это логически сложная, трудоемкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов. Однако до недавнего времени проектирование ИС выполнялось в основном на интуитивном уровне с применением неформализованных методов, основанных на искусстве и практическом опыте системных аналитиков, экспертных оценках и дорогостоящих экспериментальных проверках качества функционирования ИС. Кроме того, в процессе создания и функционирования ИС информационные потребности пользователей могут изменяться или уточняться, что еще более усложняет разработку и сопровождение таких систем [1].

Перечисленные факторы способствуют появлению программно-технологических средств специального класса – CASE средств, реализующих CASE-технологию создания и сопровождения ИС. Под разрабатываемой средой понимаются программные средства, поддерживающие процессы создания и сопровождения ИС, включая анализ и формулировку требований, проектирование прикладного ПО (приложений) и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы. Эти средства, совместно с системным ПО и техническими средствами, образуют универсально полную среду разработки ИС.

Одним из ярких представителей семейства объектно-ориентированных систем разработки является Rational Rose. Rational Rose - CASE средство фирмы Rational Software Corporation (США) - предназначено для автоматизации этапов анализа и проектирования ПО, а также для генерации кодов на различных языках и выпуска проектной документации. Rational Rose использует синтез-методологию объектно-ориентированного анализа и проектирования, основанную на подходах трех ведущих специалистов в данной области: Буча, Рамбо и Джекобсона. Разработанная ими универсальная нотация для моделирования объектов (UML - Unified Modeling Language) претендует на роль стандарта в области объектно-ориентированного анализа и проектирования. Конкретный вариант Rational Rose определяется языком, на котором генерируются коды программ (C++, Smalltalk, PowerBuilder, Ada, SQLWindows и ObjectPro). Основным вариантом - Rational Rose/C++ - позволяет разрабатывать проектную документацию в виде диаграмм и спецификаций, а также генерировать программные коды на C++. Кроме того, Rational Rose содержит средства реинжиниринга программ, обеспечивающие повторное использование программных компонент в новых

проектах. В основе работы Rational Rose лежит построение различного рода диаграмм и спецификаций, определяющих логическую и физическую структуры модели, ее статические и динамические аспекты. В их число входят диаграммы классов, состояний, сценариев, модулей, процессов. В составе Rational Rose можно выделить 6 основных структурных компонент: репозиторий, графический интерфейс пользователя, средства просмотра проекта (browser), средства контроля проекта, средства сбора статистики и генератор документов. К ним добавляются генератор кодов (индивидуальный для каждого языка) и анализатор для C++, обеспечивающий реинжиниринг - восстановление модели проекта по исходным текстам программ.

В отличие от Rational Rose, которая является универсальной системой, предлагаемая среда разработки приложений будет в большей степени ориентирована на задачи управления производственной деятельностью. Для этого в состав системы будут включены специальные дополнительные компоненты, ориентированные на эту сферу деятельности.

При ведении разработки предполагается использование структурного подхода. Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и т.д. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. При разработке системы "снизу-вверх" от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

Все наиболее распространенные методологии структурного подхода базируются на ряде общих принципов. В качестве двух базовых принципов используются следующие:

принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;

принцип иерархического упорядочивания - принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

Выделение двух базовых принципов не означает, что остальные принципы являются второстепенными, к которым относятся

принцип абстрагирования - выделение существенных аспектов системы и отвлечения от несущественных;

принцип формализации - необходимость строгого методического подхода к решению проблемы;

принцип непротиворечивости - обоснованность и согласованность элементов;

принцип структурирования данных - данные должны быть структурированы и иерархически организованы [2].

При структурном анализе будут использоваться в основном две группы средств, иллюстрирующих функции, выполняемые системой и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными среди которых являются следующие:

SADT (Structured Analysis and Design Technique) модели и соответствующие функциональные диаграммы;

DFD (Data Flow Diagrams) диаграммы потоков данных;

ERD (Entity-Relationship Diagrams) диаграммы "сущность-связь".

На стадии проектирования ИС модели могут расширяться, уточняться и дополняться диаграммами, отражающими структуру программного обеспечения: архитектура ПО, структурные схемы программ и диаграммы экранных форм (см. рисунок).

Перечисленные модели в совокупности дают полное описание ИС независимо от того, является ли она существующей или вновь разрабатываемой. Состав диаграмм в каждом конкретном случае зависит от необходимой полноты описания системы.

Для успешного внедрения предлагаемой архитектуры организация данной среды должна обладать следующими качествами:

понимание ограниченности существующих возможностей и способность принять новую технологию;

готовность к внедрению новых процессов и взаимоотношений между разработчиками и пользователями;

четкое руководство и организованность по отношению к наиболее важным этапам и процессам внедрения;

простота использования как самой среды, так и созданными на ее базе приложениями конечными пользователями.

Если организация среды не будет обладать хотя бы одним из перечисленных качеств, то внедрение может закончиться неудачей независимо от степени тщательности следования различным рекомендациям по внедрению. На самой первой стадии разработки также следует оценить проблемы, которые могут затруднить или вообще привести в тупик внедрение системы. Среди наиболее важных проблем выделим следующие:

достоверная оценка отдачи от системы затруднительна ввиду отсутствия приемлемых метрик и данных по проектам и процессам разработки ПО;

внедрение системы может представлять собой достаточно длительный процесс и может не принести немедленной отдачи. Возможно даже краткосрочное снижение продуктивности в результате усилий, затрачиваемых на внедрение. Вследствие этого руководство организации-пользователя может утратить интерес к CASE-средствам и прекратить поддержку их внедрения;

отсутствие полного соответствия между теми процессами и методами, которые поддерживаются системой и теми, которые используются в данной организации, может привести к дополнительным трудностям;

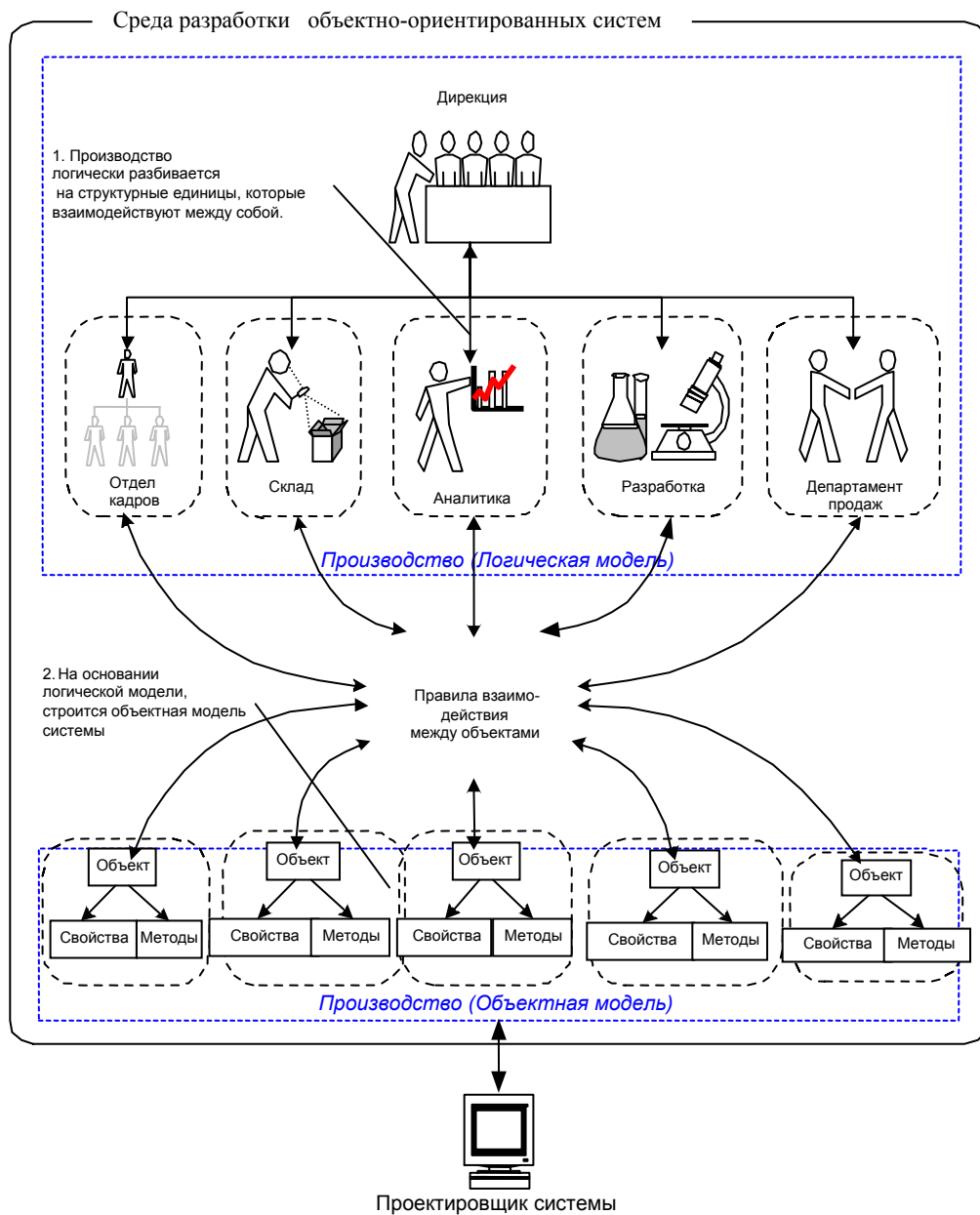


РИСУНОК. Архитектура среды разработки приложений

CASE-средства зачастую трудно использовать в комплексе с другими подобными средствами. Это объясняется как различными парадигмами, поддерживаемыми различными средствами, так и проблемами передачи данных и управления от одного средства к другому;

негативное отношение персонала к внедрению новой технологии может быть главной причиной провала проекта.

Если заранее обратить внимание на перечисленные проблемы, то успешное внедрение среды должно обеспечить следующие выгоды:

высокий уровень технологической поддержки процессов разработки и сопровождения ПО;

положительное воздействие на некоторые или все из перечисленных факторов: производительность, качество продукции, соблюдение стандартов, документирование;

приемлемый уровень отдачи от инвестиций в CASE-средства.

Технология реализации объектно-ориентированной среды разработки приложений для управления производственным процессом в малых и средних предприятиях заключается в следующем. В соответствии с архитектурой системы (см. рисунок) создаются объектно-ориентированные компоненты структурных единиц производственного предприятия (например, склад сырья и материалов, склад готовой продукции, отдел сбыта, производственные участки и т.п.). С этой целью предварительно реализуются библиотека функционально-объектных компонент, свойства и методы которых взаимнооднозначно характеризуют задачи, решаемые в рамках отдельных структурных единиц.

Функционирование этих компонент предопределяется свойствами не визуальных компонент, характеризующие взаимные информационные связи одноранговых задач. И, наконец, для целостности среды создаются информационно-процедурные модели объектов, определяющие изменения свойств всех иерархически взаимосвязанных компонент. Все свойства, методы и информационные события, генерируемые процедурными компонентами, реализуются с учетом параметрических, временных, хронологических и пространственных характеристик производственных процессов.

Подобная организация среды разработки приложений позволяет, с одной стороны, принимать непосредственное участие в процессе проектирования ИС специалистов по менеджменту автоматизируемого предприятия, с другой - моделировать производственный процесс для оценки характеристик функционирования разрабатываемой ИС.

Проектирование ИС в рамках предлагаемой среды разработки приложений в общем случае происходит по следующему сценарию. Прежде всего создается визуальная графическая структура автоматизируемого предприятия с соответствующими подразделениями. При этом каждой структурной единице ставятся в соответствие объектные компоненты верхнего уровня операционной среды. Далее, по подобной схеме происходит детализация каждой структурной единицы. В процессе детализации выделяются в отдельную визуально-графическую

область все взаимосвязанные свойства компонент, участвующих в ИС. Проектировщику предоставляется возможность осуществлять "связку" свойств посредством параметрических, временных, хронологических или пространственных характеристик, определяемых целевой задачей производственного процесса. Эти характеристики являются сутью не визуальных компонент, которые выбираются из соответствующей библиотеки. Окончательный этап проектирования - включение в процесс разработки ИС прикладных программистов для написания кодов тех участков программ, для которых по той или иной причине отсутствуют необходимые процедурно-объектные компоненты.

Таким образом, предлагаемые архитектурные и технологические решения для создания универсально полной среды разработки ИС является интеллектуальной системой с высокой степенью внутренней организации взаимодействия процессов. Выбор объектно-ориентированной технологии продиктован необходимостью согласования в процессе функционирования ИС семантико-синтаксических, хронологическо-временных и пространственных характеристик циркулирующих в рамках ИС информационных ресурсов.

1. Создание информационной системы предприятия // Computer Direct – 1996. - 2. – С. 10-12.
2. В а r k e r R. CASE Method. Entity-Relationship Modeling // Copyright Oracle Corporation UK Limited, Addison-Wesley Publishing Co. - 1990. - P. 153-160

Получено 01.07.2002