

## Конспект лекций по дисциплине

### Методология моделирования систем

#### Основные понятия теории моделирования систем

Все то, на что направлена человеческая деятельность, называется объектом. Выработка методологии моделирования направлена на упорядочение получения и обработки информации об объектах, которые существуют вне нашего сознания и взаимодействуют между собой и внешней средой.

При анализе и синтезе сложных (больших) систем получил развитие системный подход, который отличается от классического (или индуктивного) подхода. Последний рассматривает систему путем перехода от частного к общему и синтезирует (конструирует) систему путем слияния ее компонент, разрабатываемых отдельно.

В отличие от этого системный подход предполагает последовательный переход от общего к частному, когда в основе рассмотрения лежит цель, причем исследуемый объект выделяется из окружающей среды.

Специалисты по проектированию и эксплуатации сложных систем имеют дело с системами управления различных уровней, обладающими общим свойством - стремлением достичь некоторой цели.

*Система* - целенаправленное множество взаимосвязанных элементов любой природы.

*Внешняя среда* - множество существующих вне системы элементов любой природы, оказывающих влияние на систему или находящихся под ее воздействием.

*Модель* - представление объекта, системы или понятия в некоторой форме, отличной от их реального существования.

*Моделированием* называется замещение одного объекта другим с целью получения информации о свойствах объекта-оригинала путем изучения объекта-модели.

*Моделирование* - это, во-первых, построение модели, во-вторых, изучение модели, в-третьих, анализ системы на основе данной модели. Таким образом, моделирование может быть определено как представление объекта моделью для получения информации об этом объекте путем проведения экспериментов с его моделью. Теория замещения одних объектов (оригиналов) другими объектами (моделями) и исследования свойств объектов на их моделях называется *теорией моделирования*.

При системном подходе к моделированию систем необходимо прежде всего четко определить цель моделирования. Применительно к вопросам моделирования цель возникает из требуемых задач моделирования, что позволяет подойти к выбору критерия и оценить, какие элементы войдут в создаваемую модель. Поэтому необходимо иметь критерии отбора отдельных элементов в создаваемую модель.

Целями моделирования являются:

1) *оценка* - оценить действительные характеристики проектируемой или существующей системы, определить, насколько система предлагаемой структуры будет соответствовать предъявляемым требованиям;

2) *сравнение* - произвести сравнение конкурирующих систем одного функционального назначения или сопоставить несколько вариантов построения одной и той же системы;

3) *прогноз* - оценить поведение системы при некотором предполагаемом сочетании рабочих условий;

4) *анализ чувствительности* - выявить из большого числа факторов, действующих на систему, те, которые в большей степени влияют на ее поведение и определяют ее показатели эффективности;

5) *оптимизация* - найти или установить такое сочетание действующих факторов и их величин, которое обеспечивает наилучшие показатели эффективности системы в целом.

Цели 1-4 являются задачами анализа, 5 - задачей синтеза.

Важным для системного подхода является определение *структуры системы* - совокупности связей между элементами системы, отражающих их взаимодействие.

При структурном подходе выявляются состав выделенных элементов системы и связи между ними. Совокупность элементов и связей между ними позволяет судить о *структуре системы*, которая, в зависимости от цели исследования, может быть описана на разных уровнях рассмотрения. Наиболее общее описание структуры - это *топологическое описание*, позволяющее определить в самых общих понятиях составные части системы и хорошо формализуемое на базе теории графов.

Менее общим является *функциональное описание*, когда рассматриваются отдельные функции, то есть алгоритмы поведения системы, и реализуется функциональный подход, оценивающий функции, которые выполняет система, причем под функцией понимается свойство, приводящее к достижению цели.

Простой подход к изучению взаимосвязей между отдельными частями модели предусматривает рассмотрение их как отражение связей между отдельными подсистемами объекта. Такой классический подход может быть использован при создании достаточно простых моделей. Процесс синтеза модели на основе классического (индуктивного) подхода представлен на рис. 1. Реальный объект, подлежащий моделированию, разбивается на отдельные подсистемы (Д), т.е. выбираются исходные данные для моделирования и ставятся цели (Щ отображающие отдельные стороны процесса моделирования. По отдельной совокупности исходных данных (Д) ставится цель (Ц) моделирования отдельной стороны функционирования системы, на базе этой цели формируется некоторый компонент (К) будущей модели. Совокупность компонентов объединяется в модель (М).

Разработка модели на базе классического подхода означает суммирование отдельных компонентов в единую модель, причем каждый из компонентов решает свои собственные задачи и изолирован от других частей модели. Поэтому классический подход может быть использован для реализации сравнительно простых моделей, в которых возможно разделение и взаимно независимое рассмотрение отдельных сторон функционирования реального объекта. Для модели сложного объекта, к которым относятся системы качества, такая разобщенность решаемых задач недопустима,

так как приводит к значительным затратам ресурсов при реализации модели на базе конкретных программно-технических средств.

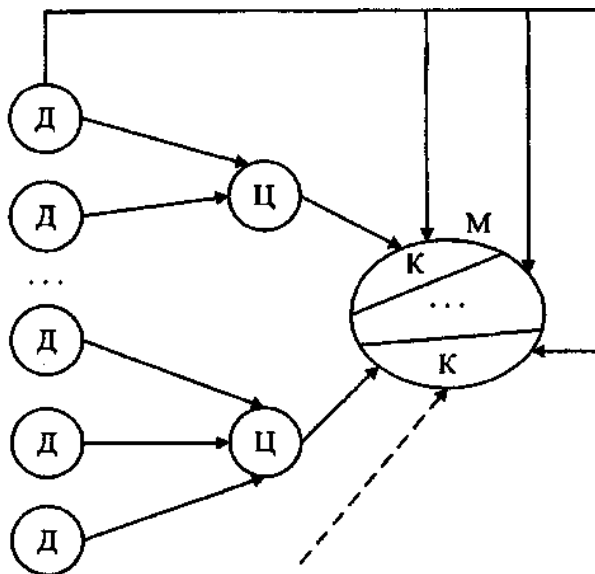


Рис. 1. Процесс синтеза модели на основе классического подхода

Можно отметить две отличительные стороны классического подхода: 1) наблюдается движение от частного к общему, создаваемая модель (система) образуется путем суммирования отдельных ее компонентов и 2) не учитывается возникновение нового системного эффекта.

Процесс синтеза модели (М) на базе системного подхода условно представлен на рис. 2. Система - целенаправленное множество взаимосвязанных элементов любой природы. Внешняя среда - множество существующих вне системы элементов любой природы, оказывающих влияние на систему или находящихся под ее воздействием.

На основе исходных данных (Д), которые известны из анализа внешней системы, тех ограничений, которые накладываются на систему сверху, либо исходя из возможностей ее реализации, и на основе цели функционирования формулируются исходные требования (Т) к модели системы. На базе этих требований формируются

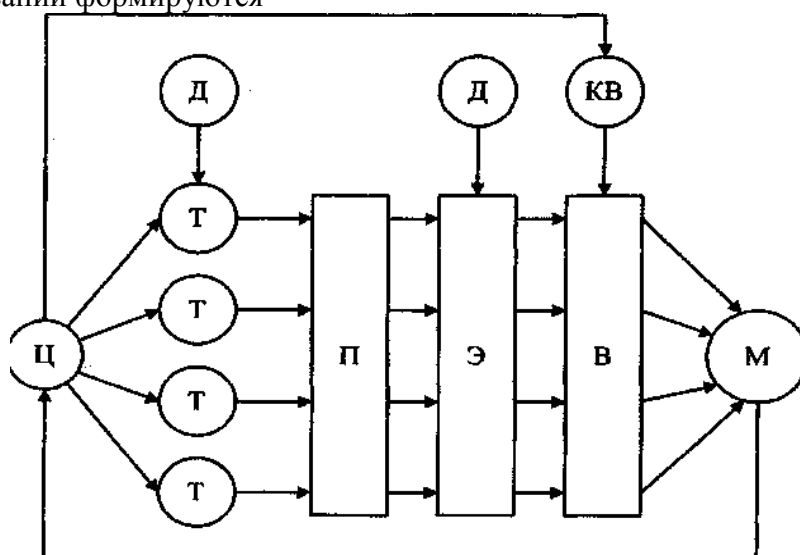


Рис. 2. Процесс синтеза модели на основе системного подхода

ориентировочно некоторые подсистемы (П), элементы (Э) и осуществляется наиболее сложный этап синтеза - выбор (В) составляющих системы, для чего используются специальные критерии выбора (КВ).

На базе системного подхода последовательность разработки моделей включает две основные стадии проектирования - макропроектирование и микропроектирование.

На стадии *макропроектирования* на основе данных о реальной системе и внешней среде строится модель внешней среды, выявляются ресурсы и ограничения для построения модели системы, выбирается модель системы и критерии, позволяющие оценить адекватность модели реальной системы.

Стадия *микропроектирования* в значительной степени зависит от конкретного типа выбранной модели. В случае имитационной модели необходимо обеспечить создание информационного, математического, технического и программного обеспечения систем моделирования.

Независимо от типа используемой модели при ее построении необходимо руководствоваться рядом принципов системного анализа.

*Принципы системного анализа* — это некоторые положения общего характера, являющиеся обобщением опыта работы человека со сложными системами.

Рассмотрим основные принципы системного анализа.

1. *Принцип конечной цели.* Это абсолютный приоритет конечной (глобальной) цели. Принцип имеет несколько правил:

- для проведения системного анализа необходимо в первую очередь сформулировать цель исследования; расплывчатые, не полностью определенные цели влекут за собой неверные выводы;
- анализ следует вести на базе первоочередного уяснения основной цели (функции, основного назначения) исследуемой системы, что позволит определить ее основные существенные свойства, показатели качества и критерии оценки;
- при синтезе систем любая попытка изменения или совершенствования должна оцениваться относительно того, помогает или мешает она достижению конечной цели;
- цель функционирования искусственной системы задается, как правило, системой, в которой исследуемая система является составной частью.

2. *Принцип измерения.* О качестве функционирования какой-либо системы можно судить только применительно к системе более высокого порядка. Другими словами, для определения эффективности функционирования системы надо представить ее как часть более общей и проводить оценку внешних свойств исследуемой системы относительно целей и задач суперсистемы.

3. *Принцип эквивиальности.* Система может достигнуть требуемого конечного состояния, не зависящего от времени и определяемого исключительно собственными характеристиками системы при различных начальных условиях и различными путями. Это форма устойчивости по отношению к начальным и граничным условиям.

*Принцип единства.* Это совместное рассмотрение системы как целого и как совокупности частей (элементов). Принцип ориентирован на «взгляд внутрь» системы, на расчленение ее с сохранением целостных представлений о системе.

4. *Принцип связности.* Рассмотрение любой части совместно с ее окружением подразумевает проведение процедуры выявления связей между элементами системы и выявление связей с внешней средой (учет внешней среды). В соответствии с этим принципом систему в первую очередь следует рассматривать как часть (элемент, подсистему) другой системы, называемой суперсистемой или старшей системой.

5. *Принцип модульного построения.* Полезно выделение модулей в системе и рассмотрение ее как совокупности модулей. Принцип указывает на возможность вместо части системы исследовать совокупность ее входных и выходных воздействий (абстрагирование от излишней детализации).

6. *Принцип иерархии.* Полезно введение иерархии частей и их ранжирование, что упрощает разработку системы и устанавливает порядок рассмотрения частей.

7. *Принцип функциональности.* Это совместное рассмотрение структуры и функции с приоритетом функции над структурой. Любая структура тесно связана с функцией системы и ее частей. В случае придания системе новых функций полезно пересматривать ее структуру, а не пытаться втиснуть новую функцию в старую схему. Поскольку выполняемые функции составляют процессы, то целесообразно рассматривать отдельно процессы, функции, структуры. Процессы сводятся

к анализу потоков различных видов: материальный поток; поток энергии; поток информации; смена состояний. С этой точки зрения структура есть множество ограничений на потоки в пространстве и во времени.

8. *Принцип развития.* Это учет изменяемости системы, ее способности к развитию, адаптации, расширению, замене частей, накапливанию информации. В основу синтезируемой системы требуется закладывать возможность развития, наращивания, усовершенствования.

Обычно расширение функций предусматривается за счет обеспечения возможности включения новых модулей, совместимых уже имеющимися. С другой стороны, при анализе принцип развития ориентирует на необходимость учета предыстории развития системы и тенденций, имеющих в настоящее время, для вскрытия закономерностей ее функционирования.

10. *Принцип децентрализации.* Это сочетание в сложных системах централизованного и децентрализованного управления, которое, как правило, заключается в том, что степень централизации должна быть минимальной, обеспечивающей выполнение поставленной цели.

Недостаток децентрализованного управления - увеличение времени адаптации системы. Он существенно влияет на функционирование системы в быстро меняющихся средах. То, что в централизованных системах можно сделать за короткое время, в децентрализованной системе будет осуществляться весьма медленно.

Недостатком централизованного управления является сложность управления из-за огромного потока информации, подлежащей переработке в старшей системе управления. Поэтому в сложной системе обычно присутствуют два уровня управления.

В медленно меняющейся обстановке децентрализованная часть системы успешно справляется с адаптацией поведения системы к среде и с достижением глобальной цели системы за счет оперативного управления, а при резких изменениях среды осуществляется централизованное управление по переводу системы в новое состояние.

11. *Принцип неопределенности.* Это учет неопределенностей и случайностей в системе. Принцип утверждает, что можно иметь дело с системой, в которой структура, функционирование или внешние воздействия не полностью определены. Сложные открытые системы не подчиняются вероятностным законам. В таких системах можно оценивать «наихудшие» ситуации и рассмотрение проводить для них.

Модель должна отвечать заданной цели ее создания, поэтому отдельные части должны компоноваться взаимно, исходя из единой системной задачи. Цель может быть сформулирована качественно, тогда она будет обладать большей содержательностью и длительное время может отображать объективные возможности данной системы моделирования. При количественной формулировке цели

возникает целевая функция, которая точно отображает наиболее существенные факторы, влияющие на достижение цели.

Построение модели относится к числу системных задач, при решении которых синтезируют решения на базе огромного числа исходных данных, на основе предложений больших коллективов специалистов. Использование системного подхода в этих условиях позволяет не только построить модель реального объекта, но и на базе этой модели выбрать необходимое количество управляющей информации в реальной системе, оценить показатели ее функционирования и тем самым на базе моделирования найти наиболее эффективный вариант построения и выгодный режим функционирования реальной системы.

### **Характеристика проблемы и классификация видов моделирования систем**

С развитием системных исследований, с расширением экспериментальных методов изучения реальных явлений все большее значение приобретают абстрактные методы, появляются новые научные дисциплины, автоматизируются элементы умственного труда. Но основным критерием любой теории является практика, и даже сугубо математические, отвлеченные науки базируются в своей основе на фундаменте практических знаний.

Одновременно с развитием теоретических методов анализа и синтеза совершенствуются и методы экспериментального изучения реальных объектов, появляются новые средства исследования. Однако эксперимент был и остается одним из основных и существенных инструментов познания.

Если раньше моделирование означало реальный физический эксперимент либо построение макета, имитирующего реальный процесс, то в настоящее время появились новые виды моделирования, в основе которых лежит постановка не только физических, но также и математических экспериментов.

Познание реальной действительности является длительным и сложным процессом. Определение качества функционирования большой системы, выбор оптимальной структуры и алгоритмов поведения, построение системы в соответствии с поставленной перед ней целью - *основная проблема* при проектировании современных систем, поэтому моделирование можно рассматривать как один из методов, используемых при проектировании и исследовании больших систем.

Одним из наиболее важных аспектов построения систем является проблема цели. Любую модель строят в зависимости от цели, поэтому одна из основных проблем при моделировании - это *проблема целевого назначения*. Подобие процесса, протекающего в модели, реальному процессу является не целью, а условием правильного функционирования модели, и поэтому в качестве цели должна быть поставлена задача изучения какой-либо стороны функционирования объекта.

Для упрощения модели цели делят на подцели и создают более эффективные виды моделей в зависимости от полученных подцелей моделирования. Например, для автоматизированных систем управления предприятием весьма существенно изучение процессов оперативного управления производством, оперативно-календарного планирования и т.д.

Если цель моделирования определена, то возникает проблема *построения модели*. Построение модели оказывается возможным, если имеется информация или выдвинуты гипотезы относительно структуры, алгоритмов и параметров исследуемого объекта. На основании их изучения осуществляется идентификация объекта.

Если модель построена, то следующей проблемой можно считать проблему работы с ней, то есть реализацию модели, основными задачами которой являются минимизация времени получения конечных результатов и обеспечение их достоверности.

Для правильно построенной модели характерным является то, что она выявляет лишь те закономерности, которые нужны исследователю, и не рассматривает свойства системы, не существенные для данного исследования. Оригинал и модель должны быть одновременно сходны по одним признакам и различны по другим, что позволяет выделить наиболее важные изучаемые свойства.

Таким образом, характеризуя *проблему моделирования в целом*, необходимо учитывать, что от постановки задачи моделирования до интерпретации полученных результатов существует большая группа сложных научно-технических проблем, к основным из которых можно отнести:

- идентификацию реальных объектов;
- выбор вида моделей;
- построение моделей и их машинную реализацию;
  - взаимодействие исследователя с моделью в ходе машинного эксперимента;
  - проверку правильности полученных в ходе моделирования результатов;
  - выявление основных закономерностей, исследованных в процессе моделирования.

В зависимости от объекта моделирования и вида используемой модели эти проблемы могут иметь разную значимость. В одних случаях наиболее сложной оказывается идентификация, в других - проблема построения формальной структуры объекта.

Возможны трудности и при реализации модели, особенно в случае имитационного моделирования больших систем. При этом следует подчеркнуть роль исследователя в процессе моделирования. Постановка задачи, построение содержательной модели реального объекта во многом представляют собой творческий процесс и базируются на эвристике. Выбор той или иной аналогии, выбор того или иного математического аппарата моделирования полностью основывается на имеющемся опыте исследователя, и ошибка исследователя может привести к ошибочным результатам моделирования.

Средства вычислительной техники, которые в настоящее время широко используются либо для вычислений при аналитическом моделировании, либо для реализации имитационной модели системы, могут лишь помочь с точки зрения эффективности реализации сложной модели, но не позволяют подтвердить правильность той или иной модели. Только на основе обработанных дан-

ных, опыта исследователя можно с достоверностью оценить адекватность модели по отношению к реальному процессу.

Если в ходе моделирования существенное место занимает реальный физический эксперимент, то здесь весьма важна и надежность используемых инструментальных средств, поскольку сбои и отказы программно-технических средств могут приводить к искаженным значениям выходных данных, отображающих протекание процесса.

При проведении физических экспериментов необходимы специальная аппаратура, специально разработанное математическое и информационное обеспечение, позволяющие реализовать диагностику средств моделирования.

В ходе машинного эксперимента могут иметь место и ошибочные действия человека-оператора. В этих условиях серьезные задачи стоят в области эргономического обеспечения процесса моделирования.

Классификация видов моделирования систем приведена на

*Детерминированное моделирование* отображает процессы, в которых предполагается отсутствие всяких случайных воздействий; *стохастическое моделирование* отображает вероятностные процессы и события. В этом случае анализируется ряд реализаций случайного процесса и оцениваются средние характеристики, то есть набор однородных реализаций.

*Статическое моделирование* служит для описания поведения объекта в какой-либо момент времени, а *динамическое моделирование* отражает поведение объекта во времени.

*Дискретное моделирование* служит для описания процессов, которые предполагаются дискретными, соответственно, *непрерывное моделирование* позволяет отразить непрерывные процессы в системах, а *дискретно-непрерывное моделирование* используется для случаев, когда хотят выделить наличие как дискретных, так и непрерывных процессов.

В зависимости от формы представления объекта (системы) можно выделить мысленное и реальное моделирование.

*Мысленное моделирование* часто является единственным способом моделирования объектов, которые либо практически нереализуемы в заданном интервале времени, либо существуют вне условий, возможных для их физического создания. Например, на базе мысленного моделирования могут быть проанализированы многие ситуации микромира, которые не поддаются физическому эксперименту. Мысленное моделирование может быть реализовано в виде наглядного, символического и математического.

При *наглядном моделировании* на базе представлений человека о реальных объектах создаются различные наглядные модели, отображающие явления и процессы, протекающие в объекте. В основу *гипотетического моделирования* исследователем закладывается некоторая гипотеза о закономерностях протекания процесса в реальном объекте, которая отражает уровень знаний исследователя об объекте и базируется на причинно-следственных связях между входом и выходом изучаемого объекта. Гипотетическое моделирование используется, когда знаний об объекте недостаточно для построения формальных моделей.

*Аналоговое моделирование* основывается на применении аналогий различных уровней. Наивысшим уровнем является полная аналогия, имеющая место только для достаточно простых объектов. С усложнением объекта используют аналогии последующих уровней, когда аналоговая модель отображает несколько либо только одну сторону функционирования объекта.

Существенное место при мысленном наглядном моделировании занимает *макетирование*. Мысленный макет может применяться в случаях, когда протекающие в реальном объекте процессы не поддаются физическому моделированию, либо может предшествовать проведению других видов моделирования.

*Символическое моделирование* представляет собой искусственный процесс создания логического объекта, который замещает реальный и выражает основные свойства его отношений с помощью определенной системы знаков или символов.

Если ввести условное обозначение отдельных понятий, то есть знаки, а также определенные операции между этими знаками, то можно реализовать *знаковое моделирование* и с помощью знаков отображать набор понятий - составлять отдельные цепочки из слов и предложений. Используя

операции объединения, пересечения дополнения теории множеств, можно в отдельных символах да описание какого-то реального объекта.

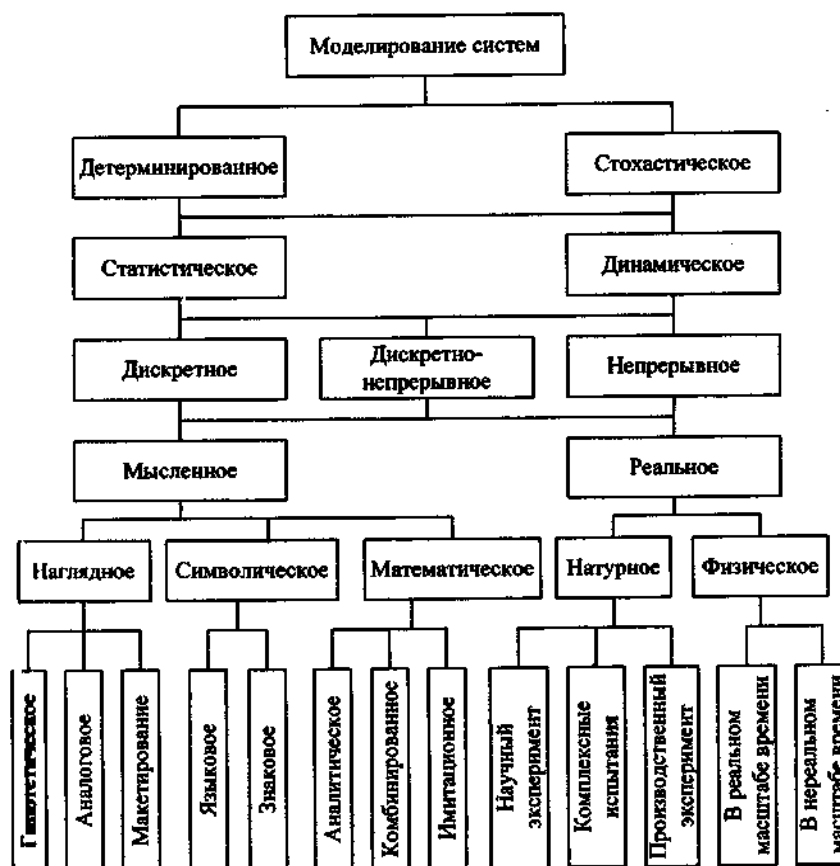


Рис. 3. Классификация видов моделирования систем

В основе *языкового моделирования* лежит некоторый тезаурус. Последний образуется из набора входящих понятий, причем этот набор должен быть фиксированным. Следует отметить, что между тезаурусом и обычным словарем имеются принципиальные различия. *Тезаурус* - словарь, в котором каждому слову может соответствовать лишь единственное понятие, хотя в обычном словаре одному слову могут соответствовать несколько понятий.

Под *математическим моделированием* будем понимать процесс установления соответствия данному реальному объекту некоторого математического объекта, называемого математической моделью, и исследование этой модели, позволяющее получать характеристики рассматриваемого реального объекта. Вид математической модели зависит как от природы реального объекта, так и от

задач исследования объекта и требуемой достоверности и точности решения этой задачи.

Для *аналитического моделирования* характерно то, что процессы функционирования элементов системы записываются в виде некоторых функциональных соотношений (алгебраических, конечно-разностных и т.п.) или логических условий.

*Имитационное моделирование* позволяет по исходным данным получить сведения о состоянии процесса в определенные моменты времени, дающие возможность оценить характеристики системы.

*Комбинированное* (аналитико-имитационное) *моделирование* при анализе и синтезе систем позволяет объединить достоинства аналитического и имитационного моделирования. При построении комбинированных моделей проводится предварительная декомпозиция процесса функционирования объекта на составляющие подпроцессы и для тех из них, где это возможно, используются аналитические модели, а для остальных подпроцессов строятся имитационные модели.

При *реальном моделировании* используется возможность исследования различных характеристик либо на реальном объекте целиком, либо на его части. Такие исследования могут проводить-



ся как на объектах, работающих в нормальных режимах, так и при организации специальных режимов для оценки интересующих исследователя характеристик (при других значениях переменных и параметров, в другом масштабе времени и т.д.). Реальное моделирование является наиболее адекватным, но при этом его возможности с учетом особенностей реальных объектов ограничены.

*Натурным моделированием* называют проведение исследования на реальном объекте с последующей обработкой результатов эксперимента на основе теории подобия. При функционировании объекта в соответствии с поставленной целью удается выявить закономерности протекания реального процесса. Надо отметить, что такие разновидности *натурного эксперимента*, как *производственный эксперимент* и *комплексные испытания*, обладают высокой степенью достоверности.

С развитием техники и проникновением в глубь процессов, протекающих в реальных системах, возрастает техническая оснащенность современного *научного эксперимента*. Он характеризуется широким использованием средств автоматизации проведения, применением весьма разнообразных средств обработки информации, возможностью вмешательства человека в процесс проведения эксперимента.

Другим видом реального моделирования является *физическое*, отличающееся от натурного тем, что исследование проводится на установках, которые сохраняют природу явлений и обладают физическим подобием. В процессе физического моделирования задаются некоторые характеристики внешней среды и исследуется поведение либо реального объекта, либо его модели при заданных или создаваемых искусственно воздействиях внешней среды. Физическое моделирование может протекать в *реальном и нереальном масштабах времени*, а также может рассматриваться без учета времени.

Особое место в моделировании занимает *кибернетическое моделирование*, в котором отсутствует непосредственное подобие физических процессов, происходящих в моделях, реальным процессам. В этом случае стремятся отобразить лишь некоторую функцию и рассматривают реальный объект как «черный ящик», имеющий ряд входов и выходов, и моделируют некоторые связи между выходами и входами. Чаще всего при использовании кибернетических моделей проводят анализ поведенческой стороны объекта при различных воздействиях внешней среды.

### Возможности и эффективность моделирования систем на ЭВМ

Наиболее конструктивным средством решения инженерных задач на базе моделирования в настоящее время стали ЭВМ. Современные ЭВМ можно разделить на две группы: универсальные, прежде всего предназначенные для выполнения расчетных работ, и управляющие, позволяющие проводить не только расчетные работы, но и приспособленные для управления объектами в реальном масштабе времени. Управляющие ЭВМ могут быть использованы как для управления технологическим процессом, экспериментом, так и для реализации различных имитационных моделей.

В зависимости от того, удастся ли построить достаточно точную математическую модель реального процесса или вследствие сложности объекта не удастся проникнуть в глубь функциональных связей реального объекта и описать их какими-то аналитическими соотношениями, можно рассматривать два основных пути использования ЭВМ: как средства расчета по полученным аналитическим моделям и как средства имитационного моделирования.

Анализ характеристик процессов функционирования больших систем с помощью только *аналитических методов* исследования приводит к необходимости существенного упрощения моделей либо на этапе их построения, либо в процессе работы с моделью, что может привести к получению недостоверных результатов.

Поэтому в настоящее время, наряду с построением аналитических моделей, большое внимание уделяется задачам оценки характеристик больших систем на основе *имитационных моделей*, реализованных на современных ЭВМ с высоким быстродействием и большим объемом оперативной памяти. Причем перспективность имитационного моделирования как метода исследования характеристик процесса функционирования больших систем возрастает с повышением быстродействия и оперативной памяти ЭВМ, с развитием математического обеспечения, совершенствованием банков данных и периферийных устройств для организации диалоговых систем моделирования. Ориентация на автоматизированные рабочие места на базе персональных ЭВМ для реализа-

ции экспериментов с имитационными моделями больших систем позволяет проводить не только анализ их характеристик, но и решать задачи структурного, алгоритмического и параметрического синтеза таких систем при заданных критериях оценки эффективности и ограничениях.

Вычислительные машины обладают возможностями, практически зависящими от порядка решаемого уравнения и от требований к скорости решения, причем могут быть использованы как ЭВМ, так и аналоговые вычислительные машины (АВМ). Последние значительно уступают ЭВМ по точности и логическим возможностям, но по быстродействию, схемной простоте реализации, сопрягаемости с датчиками внешней информации АВМ превосходят ЭВМ или, по крайней мере, не уступают им.

При использовании ЭВМ разрабатывается алгоритм расчета характеристик, в соответствии с которым составляются программы (либо генерируются с помощью пакета прикладных программ), дающие возможность осуществлять расчеты по требуемым аналитическим соотношениям. Основная задача исследователя заключается в том, чтобы попытаться описать поведение реального объекта одной из известных математических моделей.

Использование АВМ, с одной стороны, ускоряет для достаточно простых случаев процесс решения задачи, с другой стороны, могут возникать погрешности, обусловленные наличием дрейфа параметров отдельных блоков, входящих в АВМ, ограниченной точностью, с которой могут быть заданы параметры, вводимые в машину, а также неисправностями технических средств и т. д.

Перспективно сочетание ЭВМ и АВМ, то есть использование гибридных средств вычислительной техники - гибридных вычислительных комплексов (ГВК), что в ряде случаев значительно ускоряет процесс исследования.

В ГВК удастся сочетать высокую скорость функционирования аналоговых средств и высокую точность расчетов на базе цифровых средств вычислительной техники. Одновременно удастся за счет наличия цифровых устройств обеспечить контроль проведения операций.

В ряде случаев сложность объекта не позволяет не только построить математическую модель объекта, но и дать достаточно близкое кибернетическое описание, и перспективным здесь является выделение наиболее трудно поддающейся математическому описанию части объекта и включение этой реальной части физического объекта в имитационную модель. Тогда модель реализуется, с одной стороны, на базе средств вычислительной техники, а с другой - имеется реальная часть объекта. Это значительно расширяет возможности и повышает достоверность результатов моделирования.

Имитационная система реализуется на ЭВМ и позволяет исследовать имитационную модель, задаваемую в виде определенной совокупности отдельных блочных моделей и связей между ними, в их взаимодействии в пространстве и времени при реализации какого-либо процесса. Можно выделить три основные группы блоков: 1) блоки, характеризующие моделируемый процесс функционирования системы; 2) блоки, отображающие внешнюю среду и ее воздействие на реализуемый процесс; 3) блоки, играющие служебную, вспомогательную роль, обеспечивая взаимодействие первых двух, а также выполняющие дополнительные функции по получению и обработке результатов моделирования.

Имитационная система характеризуется наличием математического, программного, информационного, технического, эргономического и других видов обеспечения.

Математическое обеспечение имитационной системы включает в себя совокупность математических соотношений, описывающих поведение реального объекта, совокупность алгоритмов, обеспечивающих как подготовку, так и работу с моделью.

Программное обеспечение по своему содержанию включает в себя совокупность программ: планирования эксперимента, имитационной модели, проведения эксперимента, обработки и интерпретации результатов.

Информационное обеспечение включает в себя средства и технологию организации и реорганизации базы данных моделирования, методы логической и физической организации массивов, формы документов, описывающих процесс моделирования и его результаты.

Техническое обеспечение имитационной системы включает в себя, прежде всего, средства вычислительной техники, связи и обмена между оператором и сетью ЭВМ, ввода и вывода информации, управления проведением эксперимента. К техническому обеспечению предъявляются

весьма серьезные требования по надежности функционирования, так как сбои и отказы технических средств, ошибки оператора ЭВМ могут резко увеличить время работы с имитационной моделью и даже привести к неверным конечным результатам.

Эргономическое обеспечение имитационной системы представляет собой совокупность научных и прикладных методик и методов, а также нормативно-технических и организационно-методических документов, используемых на всех этапах взаимодействия человека-экспериментатора с инструментальными средствами (ЭВМ, гибридными комплексами и т. д.).

Таким образом, имитационная система может рассматриваться как машинный аналог сложного реального процесса и позволяет заменить эксперимент с реальным процессом функционирования системы экспериментом с математической моделью этого процесса в ЭВМ. В настоящее время имитационные эксперименты широко используют в практике проектирования сложных систем, когда реальный эксперимент невозможен.

Для большого класса задач исследования и проектирования систем метод имитационного моделирования наиболее приемлем.

В качестве основных критериев целесообразности применения метода имитационного моделирования на ЭВМ можно указать следующие:

отсутствие или неприемлемость аналитических, численных и качественных методов решения поставленной задачи;

- наличие достаточного количества исходной информации о моделируемой системе для обеспечения возможности построения адекватной имитационной модели;
- необходимость проведения на базе других возможных методов решения очень большого количества вычислений, трудно реализуемых даже с использованием ЭВМ;
- возможность поиска оптимального варианта системы при ее моделировании на ЭВМ.

К числу основных достоинств имитационного моделирования относятся следующие:

- машинный эксперимент с имитационной моделью дает возможность исследовать особенности процесса функционирования системы в любых условиях;
- применение ЭВМ в имитационном эксперименте существенно сокращает продолжительность испытаний по сравнению с натурным экспериментом;
- имитационная модель позволяет включать результаты натурных испытаний реальной системы или ее частей для проведения дальнейших исследований;
- имитационная модель обладает известной гибкостью варьирования структуры, алгоритмов и параметров моделируемой системы, что важно с точки зрения поиска оптимального варианта системы;
- имитационное моделирование сложных систем часто является единственным практически реализуемым методом исследования процесса функционирования таких систем на этапе их проектирования.

Основным недостатком, проявляющимся при машинной реализации метода имитационного моделирования, является то, что решение, полученное при анализе имитационной модели, всегда носит частный характер, так как оно соответствует фиксированным элементам структуры, алгоритмам поведения и значениям параметров системы, начальным условиям и воздействиям внешней среды.

Поэтому для полного анализа характеристик процесса функционирования систем, а не получения только отдельной точки, приходится многократно воспроизводить имитационный эксперимент, варьируя исходные данные задачи.

Эффективность имитационного моделирования может оцениваться рядом критериев: точностью и достоверностью результатов моделирования, временем построения и работы с моделью, затратами машинных ресурсов (времени и памяти), стоимостью разработки и эксплуатации модели. Наилучшей оценкой эффективности является сравнение получаемых результатов с реальным исследованием, то есть с моделированием на реальном объекте при проведении натурального эксперимента. Поскольку это не всегда удается сделать, статистический подход позволяет с определенной степенью точности при повторяемости машинного эксперимента получить какие-то усредненные характеристики поведения системы.

Существенное влияние на точность моделирования оказывает число реализаций, и в зависимости от требуемой достоверности можно оценить необходимое число реализаций воспроизводимого случайного процесса.

Существенным показателем эффективности являются затраты машинного времени. В связи с использованием ЭВМ различного типа суммарные затраты складываются из времени по вводу и выводу данных по каждому алгоритму моделирования, времени на проведение вычислительных операций, с учетом обращения к оперативной памяти и внешним устройствам, а также сложности каждого моделирующего алгоритма. Расчеты затрат машинного времени являются приближенными и могут уточняться по мере отладки программ и накопления опыта у исследователя при работе с имитационной моделью. Большое влияние на затраты машинного времени при проведении имитационных экспериментов оказывает рациональное планирование таких экспериментов. Определенное влияние на затраты машинного времени могут оказать процедуры обработки результатов моделирования, а также форма их представления.

При моделировании систем качества последние рассматриваются как сложные системы, элементами которых являются процессы, находящиеся в постоянном взаимодействии. Основная ответственность за качество возлагается на систему, для эффективного управления которой следует выполнять оценку степени взаимодействия между процессами путем построения различных видов моделей.

### Информационные модели и системы их реализующие

Корпоративные автоматизированные интеллектуальные и информационные системы на основе современных компьютерных и телекоммуникационных технологий являются эффективным инструментом управления предприятием. Современный рынок требует, чтобы вся продукция удовлетворяла стандартам качества, которые касаются не только качества конечного продукта, выставляемого на рынке, но и всего процесса производства этого продукта, начиная от выбора поставщиков и заканчивая сервисным обслуживанием.

В настоящее время широкое распространение получил комплекс стандартов на систему качества предприятия, разработанный ISO (International Standards Organization), техническим комитетом ISO/TC 176 (ИСО/ТК 176).

Внедрение и поддержание на предприятии системы качества в соответствии со стандартами семейства ИСО 9000 предполагает использование программных продуктов, по крайней мере, трех классов:

- 1) комплексные системы управления предприятием (автоматизированные информационные системы поддержки принятия управленческих решений), АИСППР;
- 2) системы электронного документооборота;
- 3) продукты, позволяющие создавать модели функционирования организации, проводить анализ и оптимизацию ее деятельности.

Это не значит, что любое предприятие, претендующее на соответствие системе качества ИСО 9000, должно обязательно иметь у себя корпоративную интеллектуальную информационную систему. Скорее, это значит, что управление огромными объемами данных, которые циркулируют на предприятии, без такой системы будет сопряжено с большими сложностями.

Как правило, корпоративная информационная система состоит из нескольких уровней.

1. Информационно-логический уровень. Представляет собой совокупность потоков данных и центров (узлов) возникновения, потребления и модификации информации.
2. Прикладной уровень. Представляет собой совокупность прикладных программ и программных комплексов, которые реализуют функционирование информационно-логической модели.
3. Системный уровень. Операционные системы и сетевые средства.
4. Аппаратный. Средства вычислительной техники.
5. Транспортный. Активное и пассивное сетевое оборудование, сетевые протоколы и технологии.

Центральное место среди задач управления с этой точки зрения занимает получение прибыли от результатов основной деятельности предприятия. Приобретение средств и орудий производ-

ства, производственные процессы, как правило, предшествуют доходам, получаемым в результате реализации произведенного товара (услуги). Поэтому важно суметь сопоставить производственные потребности с финансовыми возможностями предприятия.

Система, автоматизирующая процесс переработки информации, является лишь одним из необходимых компонентов, определяющих конечный успех предприятия. Очевидно, что преуспевающими в деловом мире в ближайшее время будут те фирмы и корпорации, которые в состоянии быстрее всех собрать информацию, обработать и проанализировать ее.

Задача планирования потребностей в материалах (Materials Requirements Planning, MRP) оказалась первой задачей, которая привела к созданию целой индустрии программного обеспечения для управления предприятием. Решение задачи планирования потребностей в материалах реализуется с помощью алгоритма, который также носит название MRP-алгоритма. MRP-алгоритм – это алгоритм оптимального управления заказами на готовую продукцию, производством и запасами сырья и материалов. MRP-методология – это реализация MRP-алгоритма с помощью компьютерной системы.

Реализация системы, работающей по этой методологии, представляет собой программную систему, позволяющую оптимально регулировать поставки комплектующих изделий в производственный процесс, контролируя запасы на складе и саму технологию производства. Главной задачей MRP является обеспечение гарантии наличия необходимого количества требуемых материалов и комплектующих в любой момент времени в рамках срока планирования, наряду с возможным уменьшением постоянных запасов, а, следовательно, разгрузкой склада.

В настоящее время MRP системы присутствуют практически во всех интегрированных информационно-системах управления предприятием.

Философия MRP, базируется на принципе, что все материалы-комплектующие, составные части и блоки готового изделия должны поступать в производство одновременно, в запланированное время, чтобы обеспечить создание конечного продукта без дополнительных задержек. MRP-система ускоряет доставку тех материалов, которые в данный момент нужны в первую очередь и задерживает преждевременные поступления. Таким образом, все комплектующие, представляющие собой полный список составляющих конечного продукта, поступают в производство одновременно. Это необходимо во избежание ситуации, когда задерживается поставка одного из материалов, и производство вынуждено приостановиться даже при наличии всех остальных комплектующих конечного продукта. Основная цель MRP-систем - формировать, контролировать и при необходимости изменять даты необходимого поступления заказов, чтобы все материалы, необходимые для производства поступали одновременно.

Таким образом, использование MRP-системы для планирования производственных потребностей позволяет оптимизировать время поступления каждого материала, тем самым значительно снижая складские издержки и облегчая ведение производственного учета

На следующем этапе появились системы класса MRP II, которые следуют двум важнейшим принципам:

- 1) логике «зависимого спроса», т.е. если есть потребность в конечном изделии, значит, есть потребность во всех его компонентах;
- 2) обеспечивать требуемые компоненты как можно позднее, чтобы уровень запасов был минимальным.

Чтобы следовать этим двум принципам, системе требуется большой объем информации. Для расчета потребностей в компонентах нижнего уровня требуется «спецификация» на каждое конечное изделие, по которой определяются компоненты, время начала и завершения работ, этапы производства и данные о «состоянии запасов». Это необходимо для того, чтобы определить, сколько требуемых компонентов имеется в запасе и в незавершенном производстве. В результате автоматизированных вычислений формируется план потребностей, который является стержнем в системах MRP II.

На планирование потребностей в материалах влияет точность спецификаций и записей о состоянии запасов. Допущенная ошибка может привести к тому, что будет вычислено неправильное количество или заказаны не те компоненты. Эта ошибка не может быть исправлена до тех пор, пока не будет обнаружена физически, и часто на это уходит несколько недель.

Ключевыми возможностями систем MRPII являются обратная связь по фактическому состоянию производства и заказов на закупку, более тщательная проверка выполнимости основного плана производства и внесение изменений в производственный план посредством приблизительного планирования мощности, анализа «что-если» и выполнения алгоритма MRP с учетом частых изменений. MRPII становится главной частью любой интегрированной вычислительной системы на производственных предприятиях.

MRPII системы объединяют процедуры обработки заказов на продажу, бухгалтерского учета, закупок и выписки счетов-фактур с производством на основе единой базы данных реального времени.

В то же время, MRPII системы не контролируют конструкторские разработки, составление сметы, кадры, сбыт и распределение продукции, обслуживание, т.е. подразделения не объединены в одну систему. Этот круг вопросов рассматривался разработчиками систем в 90-х годах, чтобы обеспечить полностью интегрированные системы для управления производственными предприятиями, в основе которых были заложены принципы MRPII, и был реализован переход в системы ERP.

Таким образом, предвидя возможные проблемы, и создавая руководству предприятия условия для предварительного их анализа, MRPII-системы является надежным средством прогнозирования и оценки последствий внесения тех или иных изменений в производственный цикл.

Использование систем MRPII позволяет:

- 1) улучшить обслуживание заказчиков - за счет своевременного исполнения поставок;
- 2) сократить цикл производства и цикл выполнения заказа - следовательно, бизнес будет более гибко реагировать на спрос;
- 3) сократить незавершенное производство - работа не будет выдаваться, пока не потребуется «точно ко времени» для удовлетворения конечного спроса;
- 4) значительно сократить запасы, что позволит более экономно использовать складские помещения и потребуются меньше средств на его хранение;
- 5) сбалансировать запасы - будет меньше дефицита и меньше устаревших запасов;
- 6) повысить производительность - людские ресурсы и материалы будут использоваться в соответствии с заказами с меньшими потерями; можно использовать анализ «что-если», чтобы проверить, соответствует ли производство задачам предприятия по получению прибыли;
- 7) создать скоординированную группу управления, которая сможет решать стратегические и оперативные вопросы и организовать работу в соответствии с выработанным основным планом производства.

Основные понятия производственного менеджмента (в том числе и термин «ERP») можно считать вполне устоявшимися. В этой области признанным «стандартом де-факто» служит терминология американской ассоциации по управлению запасами и производством (American Production and Inventory Control Society, APICS). Основные термины и определения приводятся в Словаре APICS, который регулярно обновляется по мере развития теории и практики управления. Именно в этом издании содержится наиболее полное и точное определение ERP-системы.

Термин «ERP-система» (Enterprise Resource Planning – управление ресурсами предприятия) может употребляться в двух значениях.

1. ERP-система - информационная система для идентификации и планирования всех ресурсов предприятия, которые необходимы для осуществления продаж, производства, закупок и учета в процессе выполнения клиентских заказов.
2. ERP методология – это методология эффективного планирования и управления всеми ресурсами предприятия, которые необходимы для осуществления продаж, производства, закупок и учета при исполнении заказов клиентов в сферах производства, дистрибьюции и оказания услуг.

Таким образом, термин ERP может означать не только информационную систему, но и соответствующую методологию управления, реализуемую и поддерживаемую этой информационной системой.

Следует отметить, что и для MRPII-систем, и для ERP-систем основным является производство. Они, безусловно, развиваются в связи с запросами рынка: добавляются новые функциональности, решения переносятся на новые технологические платформы. Однако производственные

подсистемы остаются центральными для рассматриваемых систем, и различия между MRPII-/ERP-системами лежат именно в области планирования производства. Связаны эти различия с глубиной реализации планирования, что обусловлено ориентацией этих систем на различные сегменты рынка.

ERP-системы создаются для больших многофункциональных и территориально распределенных производственных корпораций (например, холдингов, ТНК, ФПП и т. д.). MRPII-системы ориентированы на рынок средних предприятий, которым не требуется вся мощность ERP-систем.

Безусловными лидерами являются системы SAP R/3 немецкой компании SAP AG, Oracle Applications американской компании Oracle

Главная цель концепции ERP – распространить принципы MRPII (Manufactory Resource Planning, планирование производ-ственных ресурсов) на управление современными корпорациями. Концепция ERP представляет собой надстройку над методологией MRPII. Не внося никаких изменений в механизм планирования производственных ресурсов, она позволяет решить ряд дополнительных задач, связанных с усложнением структуры компании.

Системы класса ERP отличает набор следующих свойств:

- 1) универсальность с точки зрения типов производств;
- 2) поддержка многозвенного производственного планирования;
- 3) более широкая (по сравнению с MRPII) сфера интегрированного планирования ресурсов;
- 4) включение в систему мощного блока планирования и учета корпоративных финансов;
- 5) внедрение в систему средств поддержки принятия решений.

Несмотря на довольно широкую функциональность, ERP-системы не являются полностью интегрированными системами управления: на многих предприятиях существуют подразделения, деятельность которых хотя и связана с производственным процессом, однако не укладывается в существующую идеологию MRPII/ERP-систем. Для автоматизации работы таких подразделений используются свои системы. Речь идет, например, о системах автоматизированного проектирования (САПР), системах конструкторской и технологической подготовки производства (PDM-системы - Product Data Management). Поэтому реально ERP-системы (так же, как и MRP II-системы) практически всегда используются совместно с подобными подсистемами.

ERP-система не является инструментом для принятия управленческих решений, она лишь предоставляет необходимую для этого информацию. Реальную же поддержку принятия управленческих решений оказывают специальные аналитические средства, вводимые в ERP-системы (обычно эти средства называют OLAP - On-Line Analysis Processing).

Управление отношениями с клиентами (Customer Relations Management, CRM) - это стратегия, основанная на применении таких управленческих и информационных технологий, с помощью которых компании аккумулируют знания о клиентах для выстраивания взаимовыгодных отношений с ними [1, 3, 6]. Подобные отношения способствуют увеличению прибыли, т. к. привлекают новых клиентов и помогают удерживать старых.

CRM - это клиент-ориентированная стратегия, с одной стороны, формирования наценки «выше рыночной» за счет обеспечения индивидуального обслуживания каждого клиента, а с другой - ориентации на долгосрочные отношения, в том числе и в ущерб краткосрочным экономическим задачам. Целью CRM является не просто увеличение объема продаж, а прибыльное «увязывание» потребностей клиента с возможностями продавца, что и требует совместной коллективной работы на клиента различных функциональных подразделений организации.

Таким образом, CRM «в большом» – это стратегия «отличительного» ведения бизнеса. CRM «в малом» - собственно информационные технологии, позволяющие формализовать и автоматизировать различные аспекты взаимодействия с клиентами подразделений маркетинга, продаж и сервисного сопровождения на основе автоматических/автоматизированных процессов (в том числе сбытовых) и единого «информационного пространства» организации. То есть происходит консолидация всей информации о каждом клиенте путем обмена данными с другими информационными системами. Объединяя ключевые блоки информации о контактах, организациях, сделках, заказах/ проектах и связях между этими «сущностями», CRM-система позволяет, опираясь на факты,

узнать все о поведении клиентов и подобрать экономически целесообразный способ их обслуживания, ведя бизнес «проактивно».

## Практические аспекты эксплуатации корпоративных информационных систем Особенности перехода к единой корпоративной информационной системе

В 80-90-е годы прошлого века, когда компьютерная техника стала повсеместно распространяться на предприятиях в РФ, было разработано множество систем с использованием самых современных, на тот период времени инструментальных средств, таких как Clipper, FoxPro, которые позволяли автоматизировать различные виды деятельности и работать с массивами данных. Многие, из которых продолжают успешно эксплуатироваться и в настоящее время. Как правило, реинженеринг таких систем заключался в переходе на более новую версию средства разработки, на котором была написана система, например с FoxPro 2.0 на FoxPro 2.5 и далее на Visual FoxPro, с улучшением функциональности, которое позволяло реализовать новая версия инструментального средства. Однако по мере накопления больших объемов хранимой информации и переходу к централизации хранения данных, эксплуатация таких систем вызывает определенные трудности связанные с их сопровождением, синхронизацией данных, кроме того, данные системы не соответствуют требованиям по производительности, расширяемости и масштабируемости. Поэтому реинженеринг в этом случае, заключается в переходе на абсолютно новые архитектуры и соответственно средства разработки, которые позволяют решить описанные выше трудности.

Нами был выполнен комплекс работ по реинженерингу корпоративной системы при переходе от многопользовательских приложений, реализованных средствами FoxPro к единой распределенной многослойной архитектуре «клиент – сервер приложений – СУБД» базирующейся на Java - технологиях.

Изначально система состояла из 10 комплексов таких как АРМы «Банк», «Кадры», «Зарплата», «Склад» и т.д.), которые базировались на общем ядре АРМ "Баланс" для выполнения основных бухгалтерских операций, а также использовали свои специфичные модули бизнес логики, отчетов и таблицы для хранения данных. Весь комплекс эксплуатировался как в аппарате управления так и на значительно удаленных филиалах.

Очевидно, что мгновенный переход к новой архитектуре невозможен, так как во-первых система находится в промышленной эксплуатации и любое внеплановое время простоя приводит к большим убыткам, во-вторых за годы эксплуатации системы накоплены огромные массивы информации перенос и синхронизация которых на новую платформу требует больших временных затрат.

Поэтому на первом этапе было принято решение о переходе на архитектуру клиент-сервер, с использованием наиболее совместимой с FoxPro СУБД – MS SQL Server 2000 и реализации части бизнес-логики в виде хранимых процедур. Данный процесс по времени занял примерно около одного года.

На втором этапе была реализована возможность ведения единой базы нормативно справочной информации (НСИ) и реестра контрагентов. Первая задача была решена средствами СУБД MS SQL Server 2000 с помощью механизма репликации. В силу того, что НСИ является относительно статичной структурой, т.е. изменения и дополнения данных выполняются не часто, было принято решение делать репликацию снимка НСИ с сервера аппарата управления, которая выполняется раз в сутки в нерабочее время и передавать его подписчикам – SQL серверам филиалов. Для решения второй задачи, был задействован совершенно другой механизм, в силу того, что филиалам при заключении договоров необходимо оперативно заводить нового контрагента, в случае если он отсутствует в едином реестре, причем с данным контрагентом, договора могут заключать разные филиалы. Т.е. необходимо чтобы при вводе нового контрагента он автоматически попадал в единый реестр. Для этих целей стали использовать каркас (framework) менеджера транзакций Mule, реализовав с его помощью корпоративную шину для обмена информацией (enterprise service bus - ESB), следующим образом. При вводе нового контрагента, перед тем как выполнить внесение в единый реестр, выполняется его поиск на центральном сервере аппарата управления по ИНН и КПП, и если в другом филиале данный контрагент в этот день уже был заведен, то выдается его идентификатор, в противном случае он добавляется в реестр, а вечером реестр вместе с НСИ реп-



лицируется на все филиалы. Реализация, отладка и внедрение второго этапа по времени заняли примерно около полугода.

Параллельно первому и второму этапам выполнялось проектирование и реализация много-слойной архитектуры распределенной корпоративной системы. В качестве инструментальных средств выбрана Java – технология и решения базирующиеся на ней. Так как во–первых, де-факто Java– технологии – промышленный стандарт в области больших корпоративных распределенных систем, во–вторых сам язык Java и сопутствующие средства, такие как, например среда разработки NetBeans, Eclipse, средство ORM – отображения Hibernate, всевозможные менеджеры отчетов и пр. являются свободно распространяемыми. В-третьих, в силу открытости данной технологии, она легко интегрируется с любой СУБД. В-четвертых, Java является аппаратно- и платформенно-независимой, т.е. система реализованная с помощью Java – технологии может работать на компьютерах не только с архитектурой x86 и не только на операционной системе Windows.

Рассмотрены варианты смены платформы СУБД на более «дружественную» по отношению к Java. Был реализован ряд тестов для сравнительной оценки производительности при работе с СУБД, на основании которых в качестве альтернативы предполагается использование MySQL v.5, PostgreSQL 8.2, также отработана технология межплатформенного переноса данных.

Архитектура системы представлена на рисунке. Рассмотрим каждый из уровней. Уровень базы данных отвечает за реализацию физического хранения. Во время первого и второго этапов выполнялся постоянный мониторинг работы SQL-сервера в результате которого были выявлены «узкие» места в его работе, снижавшие его производительность.

Во-первых, была выполнена нормализация всех отношений и их приведение к третьей нормальной форме. Во-вторых, для снижения нагрузки на СУБД, было принято решение реализовать хранение небольших по объему, около сто записей, справочников в формате XML на Web-сервере, а при загрузке приложения выполнять анализ данного файла и заносить его содержимое в стековые структуры.

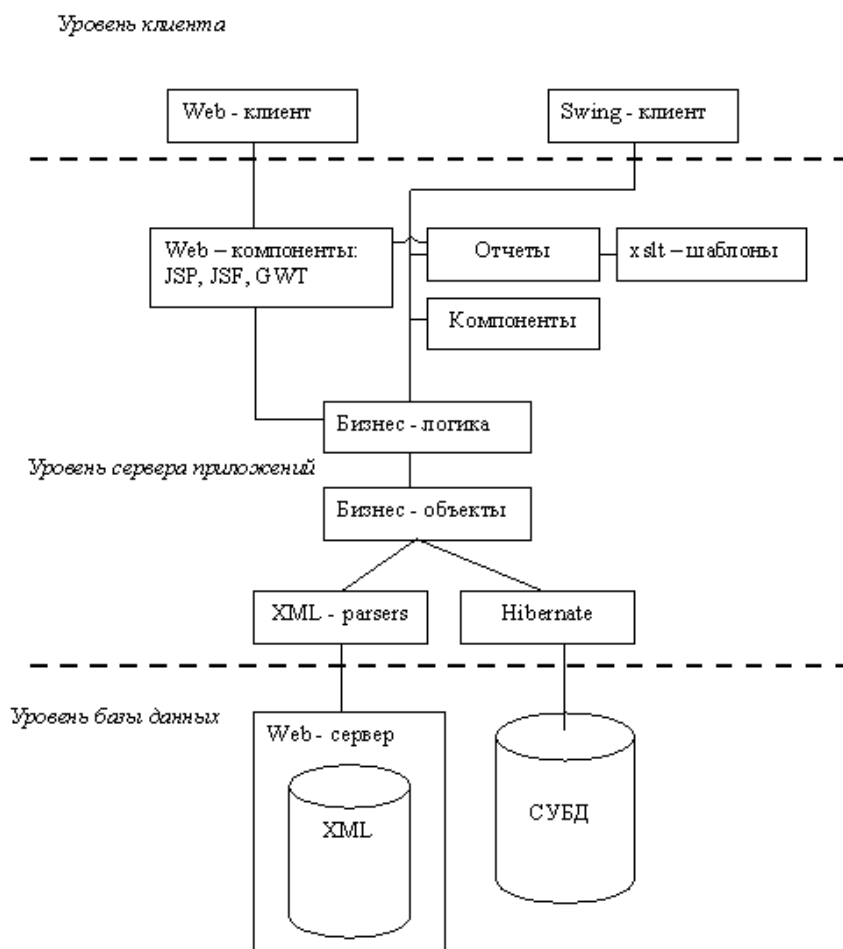


Рис.1. Архитектура корпоративной информационной системы

Такой подход позволяет добиться повышения производительности системы, как на стороне сервера, так и на стороне клиента. В-третьих, решили отказаться от хранимых процедур на стороне СУБД и вынести их на уровень сервера приложений, так как при значительном увеличении количества пользователей пытающихся их одновременно выполнять существенно снижает производительность. Кроме того, специфика синтаксиса языка T-SQL MS SQL Sever не позволяет переносить хранимые процедуры на другие серверные платформы без их глобальной переработки.

Уровень сервера приложения представляет собой множество слоев (см. рис 1.), которые при необходимости для повышения производительности могут быть разнесены между несколькими серверами. Взаимодействие между уровнем баз данных и уровнем сервера приложений реализуется ORM системой – Hibernate. Это обстоятельство является одной из важных частей позволяющей довольно просто переходить от одной платформы СУБД к другой, без изменений в слоях. Слой бизнес-логики в свою очередь условно можно разделить на несколько частей. Первая часть отвечает за работу с СУБД. Основная цель, которую пытались достичь при реализации данной части – снижение нагрузки на СУБД. Для чего все методы реализующие запросы к СУБД старались делать универсальными, т.е. не зависящими от реализации конкретного бизнес объекта, используя компонент библиотеки Hibernte – Criterion, а далее работать с полученными коллекциями, используя механизмы Java - RTTI и Reflection реализуя методы, позволяющие выбирать бизнес объекты из коллекции по различным критериям.

Кроме того, при работе с большим количеством бизнес объектов (десятки и сотни тысяч) использовали отсоединенные коллекции (DetachetCriterion). Описанное выше, является второй важной частью делающей приложение независимым от платформы СУБД. Следующая составная часть слоя бизнес-логики – специфичные методы, реализованные для конкретной задачи.

В слой компонент вынесены всевозможные модели и контроллеры (слушатели, адаптеры), которые могут использоваться как для разработки интерфейса пользователя, так и для бизнес-логики и отчетов.

Слой отчетов, это фактически набор шаблонов для их построения. Отчеты строятся с использованием библиотеки JasperReport, а при реализации Web-клиента с помощью xslt преобразования, для чего реализован подслой с xslt – шаблонами.

Последний слой – Web-компонент реализован для построения web-интерфейса.

На уровне клиента реализовано два типа интерфейсов – Web-клиент и полнофункциональный Swing – клиент. Так как ни одно web-приложение не обладает такими возможностями, которые предоставляет для пользователя библиотека Swing. Поэтому было принято решение для реализации относительно простых возможностей использовать Web-интерфейс, а взаимодействие посредством Swing-клиента, реализовать, используя технологию Java Web-start.

Кроме того, была разработана технология единой идентификации пользователей корпоративной системы на территории всей Ростовской области на базе быстрого и удобного механизма - LDAP сервера. На сервере, данные о пользователях разбиты в зависимости от их принадлежности к филиалу в иерархические структуры, которые повторяют организационные структуры предприятия. Сам объект «пользователь», хранящийся на сервере, содержит массу дополнительной информации, которая в том числе может использоваться для конфигурирования программ, например атрибут филиала – IP адрес сервера СУБД. Более того, извлекая информацию с LDAP сервера и используя для доступа Web-интерфейс, можно формировать справочную систему, например телефонный справочник предприятия, не затрачивая дополнительных усилий.

Таким образом, был выполнен реинженеринг существующей системы и разработана архитектура и каркас (framework) приложения. Однако процесс перехода на новую архитектуру был довольно сложным и долгим, так как одна из самых больших проблем связана с переносом данных в новые структуры, а малейшие изменения в существующих структурах данных могла привести к катастрофическим последствиям. Также для коллективной разработки, использовался Subversion server(SVN). Это обеспечивает качественную связь всех разработчиков системы. В качестве клиента SVN был выбран продукт TortiseSVN.

Следующим этапом стал переход на централизацию юридических задач. Таким образом, при заключении нового договора, номер ему присваивается на сервере аппарата управления, в результате чего все договора должны иметь сквозную нумерацию. В этой связи мы были вынуждены решать две большие задачи. Первая – реализация механизма двухфазной транзакции. Вторая –

сведение существующих договоров, находящихся на исполнении, в единый реестр. В этом случае возникают две больших проблемы, которые заключаются в том, что при простом объединении информации со всех филиалов в базе данных нарушается ограничение первичного ключа, а идентификатор договора фигурирует в бухгалтерских проводках и остатках, и поэтому при его изменении необходимо выполнять перекодировки в связанных с ним сущностях.

Уровень клиента. Приложение работает по технологии Web-start и имеет Swing- интерфейс.

Уровень сервера приложений. На основании полученного с LDAP сервера IP-адреса происходит настройка подключения к СУБД. В случае если указан IP-адрес филиала, то выполняется настройка Hibernate на создание двух фабрик сессий (SessionFactory), одна из которых создается с центральным сервером другая с локальным. При этом бизнес объект имеет два варианта мэппинга, основное отличие которых заключается в способе получения идентификатора, на сервере значение – native, а на филиале – assign. В случае если IP-адрес центрального сервера, то соответственно создается одна фабрика сессий. Исходя из этого, работают методы сохранения и обновления бизнес-объектов. При работе юристов филиала вначале изменения вносятся на центральный сервер, а затем на локальный, в случае их успешного занесения на оба сервера происходит принятие транзакций (commit) на обоих серверах, в случае возникновения каких либо проблем на одном из серверов происходит общий откат (rollback) транзакции. Следует заметить, что за время эксплуатации данной технологии в течение четырех месяцев, каких либо значимых сбоев или потери информации не было. Следующим шагом стал переход на технологию с использованием менеджера транзакции JOTM. В силу того, что отсутствует возможность установки и эксплуатации сервера приложений на каждом филиале, было принято решение использовать один сервер приложений Tomcat, на котором формировать источники данных для каждого филиала.

Уровень базы данных. Уровень базы данных на филиале и в аппарате управления отличается только способом задания уникальности идентификатора бизнес объекта. В аппарате управления установлено автоматическое увеличение на единицу.

Таким образом, переход к полноценной единой корпоративной системе может растянуться на годы. Несмотря на его трудоемкость очевидны его преимущества, особенно с использованием технологий от Sun Microsystems. Которые обеспечивают, оперативный доступ к любому сегменту информации на территории Ростовской области, являются свободно распространяемыми, в отличие, например от подобных технологий на базе SAP R/3.

Некоторые способы реализации механизма распределенной транзакции

В настоящее время, нет универсального архитектурного решения в области корпоративных информационных систем, которое бы позволило удовлетворить множество, порой противоречивых требований. Обычно то или иное решение, как правило, принимается на основании множества факторов, в частности, специфики деятельности предприятия, опыта эксплуатации предыдущих систем, характеристик коммуникационных линий между подразделениями и пр. Следует отметить, что во всех без исключения случаях, во главу угла ставится надежность хранения данных, которая решается, как аппаратными, так и программными средствами. Также одним из важных вопросов, является проблема синхронизации данных, которая может быть решена средствами СУБД, например различными типами репликации. Однако при таком подходе увеличивается нагрузка на СУБД, что влечет за собой снижение производительности всей системы.

Рассмотрим способы синхронизации данных с помощью механизма двухфазной транзакции, реализованных нами при разработке корпоративной системы. Данное решение повышает производительность всей системы в целом и позволяет пользователям взаимодействовать с системой в режиме реального времени. Механизм двухфазной транзакции реализован нами различными средствами в различных архитектурных решениях:

- на стороне клиента Swing приложения запускаемого по Web –start с помощью библиотеки Hibernate;
- на сервере приложений с помощью менеджера распределенных транзакций JOTM;
- в виде отдельной службы с планировщиком задач, для синхронизации данных с помощью Hibernate, JOTM и реестра rmiregistry.

Распределенная транзакция включает в себя несколько локальных транзакций, каждая из которых либо фиксируется, либо прерывается. Распределенная транзакция фиксируется только в том

случае, когда зафиксированы все локальные транзакции, ее составляющие. Если хотя бы одна из них была прервана, то должна быть прервана и распределенная транзакция. Данный механизм может быть реализован как на распределенной БД, так и на множестве СУБД, имеющих часть общей информации необходимой для функционирования единой корпоративной системы.

Для этого в СУБД предусмотрен так называемый протокол двухфазовой фиксации транзакций (two-phase commit protocol - 2PC). Название отражает то, что фиксация распределенной транзакции выполняется в две фазы. Управляет транзакцией - менеджер распределенных транзакций, например JBoss, JOTM, WebSphere 6. Они, как правило, интегрируются в сервер приложений, но могут также работать в режиме standalone.

Первая фаза начинается, когда клиент выполняет оператор COMMIT. Менеджер транзакции направляет уведомление PREPARE TO COMMIT – «подготовиться к фиксации» всем серверам БД, выполняющим транзакцию. Последние после подготовки к фиксации остаются в состоянии готовности и ожидают от менеджера команды фиксации.

Выполнение второй фазы заключается в том, что менеджер направляет команду COMMIT серверам на всех узлах, затронутых транзакцией. Выполняя команду, последние фиксируют изменения, достигнутые в процессе выполнения распределенной транзакции. В результате гарантируется одновременное синхронное завершение (удачное или неудачное) распределенной транзакции на всех участвующих в ней узлах.

Основной недостаток технологии физического распределения данных - жесткие требования к скорости и надежности каналов связи. Действительно, когда узлы с локальными БД соединены локальной вычислительной сетью, а распределенная транзакция затрагивает два-три узла сети, то, как правило, серьезных проблем с фиксацией распределенных транзакций не возникает.

Совершенно иная ситуация складывается в том случае, когда база данных распределена по нескольким территориально удаленным узлам, а число одновременно работающих в распределенной среде пользователей составляет сотни и тысячи.

Аналогичная проблема, была решена нами в рамках задачи централизации корпоративной системы. В единое корпоративное информационное пространство входит 7 серверов СУБД с MS SQL Server расположенных на филиалах и один в аппарате управления. Задача централизации была разбита на три подзадачи.

1. Ведение единого реестра, какого либо объекта, входящего в корпоративную информационную систему.
2. Синхронизация единого реестра со стороны центрального сервера.
3. Обеспечение доступа, в режиме реального времени, к распределенным сущностям.

Рассмотрим первую подзадачу на примере решения проблемы реализации механизма распределенной транзакции при ведении единого реестра договоров. Договора могут заключаться на всех филиалах.

Можно выделить три типа договоров:

1. Договор, заключенный филиалом или аппаратом управления в своих интересах, если он заключен филиалом, то он должен быть сохранен на сервере аппарата управления и на сервере филиала, если аппаратом управления в своих интересах – то только на сервере аппарата управления.
2. Договор, заключенный аппаратом управления в интересах филиала. Его необходимо сохранять на сервере филиала, в интересах которого он был заключен и на сервере аппарата управления.
3. Централизованный договор, заключенный аппаратом управления в интересах всех филиалов необходимо сохранять на сервере аппарата управления и на серверах всех филиалов.

На первом этапе был реализован механизм распределенной. Кроме того, любые изменения в договорах, должны быть реализованы по описанному выше взаимодействию.

При внесении нового договора, он, в начале, должен быть сохранен на сервере аппарата управления. Договору присваивается идентификатор, после чего, он при необходимости сохраняется на сервере филиала. Аналогичным образом должно выполняться и редактирование транзакции средствами Hibernate. При авторизации пользователя происходит его идентификация на LDAP-сервере, после чего клиентскому приложению передается информация о филиале, к которому принадлежит пользователь, и в зависимости от этого на стороне клиента создается либо две

Нibernateфабрики сессий, если он не принадлежит к аппарату управления либо одна в противном случае.

Далее при сохранении объекта, выполняется последовательное сохранение, вначале на центральном сервере, а потом на сервере филиала. При этом если COMMIT на центральном сервере прошел успешно, выполняется фиксация транзакции на сервере филиала. В случае неудачной фиксации на сервере филиала происходит последовательный «откат» транзакции. Несмотря на очевидное несовершенство данного механизма реализации двухфазной транзакции, который нами рассматривался как временный, переходный вариант, промышленная эксплуатация системы в течение четырех месяцев с его использованием показала неплохие результаты.

При вводе тысяч объектов с удаленных серверов имели место единичные случаи, когда объект сохранялся на центральном сервере, а на сервере филиала отсутствовал. Анализ данных случаев показал, что 90% из них произошло из-за загруженности каналов связи между аппаратом управления и филиалом. Оставшиеся 10% связаны с загруженностью локального сервера СУБД.

Каждый объект (договор) имеет атрибут принадлежности к филиалу, который является одним из параметров на основании, которого регламентируется доступ к нему. Таким образом, на центральном сервере хранится полный реестр договоров, а на филиале те, которые ему принадлежат, а так же централизованные и заключенные в его пользу. С учетом этого, необходимо было разработать механизм, который бы обеспечивал синхронизацию данных центрального сервера с филиальскими при вводе и редактировании централизованных и заключенных в пользу филиала договоров. На начальном этапе были разработаны SQL-скрипты, с помощью которых, используя технологию связанных серверов (Linked - Servers) выполнялась синхронизация. В дальнейшем, для решения данной задачи, был разработан инструмент, который запускается по расписанию в режиме standalone и выполняет рассылку. Работает инструмент следующим образом. При запуске формируется контекст JNDI имен серверов, на которые необходимо выполнять рассылку. Для этого на локальной машине инициализируется gmi-сервер, в котором устанавливается связь между менеджером транзакций - JOTM, источниками данных - SQL-серверов филиалов и Hibernate – фабрик сессий. После того как сформирован JNDI контекст и выполнено связывание фабрик сессий Hibernate, драйверов СУБД и менеджера транзакций, выполняется рассылка объектов. При этом в рамках одной транзакции выполняется предварительное сохранение на всех серверах.

## СИСТЕМЫ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ

### Концепция систем управления бизнес-процессами

Бизнес-процесс можно рассматривать как организованный комплекс взаимосвязанных действий, которые в совокупности дают ценный для клиента результат. В зависимости от контекста, термин бизнес-процесс может означать и метод, технологию в приведенном выше смысле, и конкретный, протекающий «здесь и сейчас» процесс, например, заказа продукции клиентом. Там, где необходимо подчеркнуть разницу между этими понятиями, для них используются соответственно термины схема или описание и экземпляр бизнес-процесса. Так же, бизнес-процесс иногда называют технологией получения коммерческого результата. При этом подчеркивается повторяемость: требуемый результат неукоснительно вытекает из заданных предусловий при соблюдении установленной процедуры. Однако бизнес-процесс – это больше, чем просто следование установленной процедуре. Строгое определение бизнес-процесса, данное выше, требует, чтобы процедура эта была оптимизирована на удовлетворение потребности клиента. Если работа на предприятии (включая систему стимулирования) организована так, что отдельные работники, менеджеры, подразделения преследуют свои узкие цели, и никто из них не беспокоится о достижении нужного клиенту результата, то это значит, что основная идея бизнес-процессов данным предприятием не усвоена.

Исторически концепция бизнес-процесса появилась как ответ на органические недостатки управления, организованного по функциональному признаку. Традиционно управление предприятием делится на функциональные области, за которые отвечают отделы: производственный, бухгалтерия, финансовый, снабжение, продажи, и т.д. Фундаментальная неэффективность такой системы обусловлена тем, что в ней каждый преследует цели личные или своего подразделения, и никто не нацелен на конечный результат – удовлетворение потребности клиента.

Иллюстрация проблем, возникающих в такой системе, рассмотрим на каноническом примере процесса продажи товара:

- отдел продаж стремится максимально снизить цену и сроки, чтобы убедить клиентов купить товар и выполнить свой план по продажам;
- технологи, поставленные перед фактом подписанного договора, говорят, что произвести товар с заданными в спецификации характеристиками или в установленные контрактом сроки невозможно;
- бухгалтерия извещает руководство, что данная сделка будет убыточной, потому что прописанная в контракте цена ниже себестоимости;

Подобные проблемы, конечно, решаются, но только непрекращающимися усилиями менеджмента. Управление на основе бизнес-процессов – это путь к решению их на системной основе. Бизнес-процессы призваны «сломать стены между отделами» и подчинить деятельность предприятия главным, а не локальным целям.

Схемы бизнес-процессов – это особо ценная форма интеллектуального капитала компании. Но, как показал опыт 90-х годов, накопление его идет слишком медленными темпами и обходится слишком дорого.

Теоретически внедрить бизнес-процесс можно двумя способами: либо разработав его «с чистого листа», либо критически переработав существующую «практику». Первому подходу соответствует английский термин *engineering* (конструирование), второму – *re-engineering* (повторное конструирование, перестройка).

Потребность в технологиях ведения бизнеса проявляется по мере взросления и роста компании; пока компания состоит из нескольких увлеченных основателей и производит уникальный продукт и услугу, бизнес может быть «кустарным». Поэтому интерес к бизнес-процессам в первую очередь проявляют зрелые компании, работающие в стабильных отраслях. А для таких компаний возможен только путь перестройки уже существующей практики, отсюда – устойчивое словосочетание «реинжиниринг бизнес-процессов».

Система управления бизнес-процессами (BPMS – Business Process Management System) эффективно решает перечисленные выше проблемы. BPM-проект нацелен на конечный результат – внедрение бизнес-процесса; в нем невозможна подмена конечного результата промежуточным – рисованием множества пусть правильных, но самих по себе бесполезных схем. Схема бизнес-процесса в BPM предельно конкретна. BPM нацелен на автоматизацию конкретных, критичных для бизнеса процессов, а не на составление всеобъемлющей картины жизни предприятия. Процессы, попадающие в фокус, моделируются не «примерно», а очень точно. BPM – средство изучения бизнес-процессов. BPM радикально сокращает время и затраты на внедрение бизнес-процессов. Благодаря этому появляется возможность оптимизировать бизнес-процесс не «большим скачком», а последовательно, в несколько этапов. Бизнес-процесс внедряется в кратчайшие сроки, так, как он видится вначале. Затем, по мере выявления и осознания расхождений со сложившейся практикой или требованиями бизнеса, в него вносятся изменения. Благодаря тому, что BPM – средство прямого управления бизнес-процессами, изменения не приходится проводить в жизнь через переобучение, изменение должностных инструкций и т.п. BPM позволяет постоянно поддерживать схему бизнес-процесса в актуальном состоянии. Если изменение бизнес-окружения или внутренних требований бизнеса диктует изменение схемы бизнес-процесса, то такие изменения оперативно вносятся в BPM-систему. Правила могут меняться и меняются, но никакие операции не выполняются в обход них. Важно, что изменения в схему бизнес-процесса под силу вносить собственным специалистам предприятия, для этого не требуется ни программистская квалификация, ни глубокие теоретические познания.

#### Формальное описание бизнес-процессов

Для формального описания бизнес-процессов и протоколов их взаимодействия между собой используется язык на основе XML – BPEL (Business Process Execution Language). BPEL расширяет модель взаимодействия веб-служб и включает в эту модель поддержку транзакций.

Интеграция разнородных систем и приложений в комплексные бизнес-процессы всегда была нелегкой задачей. Web-сервисы появились как средство доведения функциональности этих систем до всей организации, но сами по себе они не ориентированы на нужды интеграции и координации

бизнес-процессов. Язык BPEL (другие названия BPEL4WS и WS-BPEL) устраняет это слабое место, отображая модель того, как Web-сервисы оперируют бизнес-процессами.

BPEL – это язык, используемый для управления коммуникациями между системами или бизнес-партнерами. Поскольку он предназначен для описания процессов, то BPEL-процесс может представлять собой ИТ- или бизнес-сценарий. BPEL-процессы обычно исполняются на больших временных интервалах и используются для решения широкого круга задач – от интеграции систем до координирования процесса согласования финансовых отношений между деловыми партнерами.

Уникальные языковые конструкции, присущие BPEL, позволяют поместить соответствующую спецификацию на вершину иерархии Web-сервисов OASIS и делают ее хорошим средством для организации взаимодействия сервисов, которая получила название «оркестровка». Значение BPEL возросло вместе с широким распространением сервис-ориентированной архитектуры SOA (Service-Oriented Architecture), поскольку она требует оркестровки. Последняя позволяет организациям определять и переопределять бизнес-процессы в режиме реального времени, создает так необходимую организациям гибкость и является ключевой ценностью SOA.

### Архитектура BPMS

Типичная BPM-система состоит из стандартного набора компонент, соответствующих стадиям жизненного цикла бизнес-процесса: проектированию, исполнению, мониторингу.

Под проектированием понимается разработка схемы бизнес-процесса. В состав BPM-системы обычно входят:

- графический дизайнер для рисования схемы бизнес-процесса;
- репозиторий для ее хранения и организации совместного доступа.

Возможность моделировать бизнес-процесс при помощи графического редактора является принципиальной особенностью BPM-систем: проектирование бизнес-процесса должен выполнять бизнес-аналитик без участия программиста.

Процедура создания модели бизнес-процесса мало чем отличается от привычной для бизнес-аналитиков процедуры рисования схем. Нарисовать шаги, описать бизнес-логику, определить группы пользователей и перечень вводимых на каждом шаге реквизитов.

Результат сохраняется на сервере, после чего процесс может быть инициирован. При необходимости в схему можно вносить изменения, не прибегая к помощи программистов. Альтернативно, схема бизнес-процесса может разрабатываться традиционными средствами моделирования бизнес-процессов и переноситься в BPM-систему при помощи импорта-экспорта.

Ядром BPM-системы является его «движок» (BPM Engine) (рисунок).

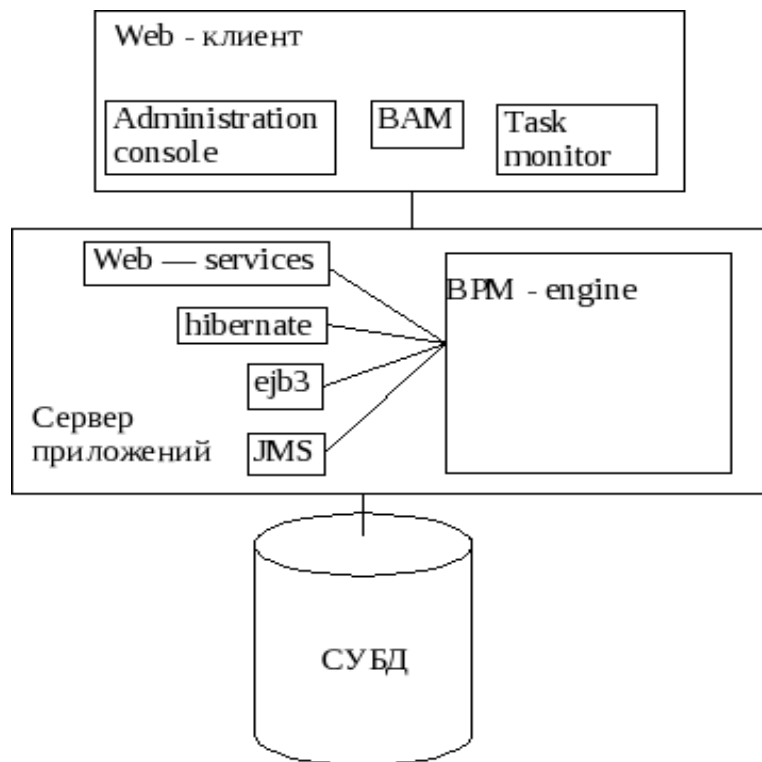


Рисунок – Архитектура BPM системы

Он запускает экземпляры бизнес-процессов, отслеживает смену их состояний, хранит значения реквизитов, выполняет бизнес-правила.

Ядро BPM-систем предоставляет также интерфейсы для стыковки с внешними приложениями – специальные адаптеры, веб-сервисы, драйверы для доступа к реляционным базам данных или к другим источникам данных. Использование этих интерфейсов зависит от типа бизнес-процесса.

Относительно небольшую долю составляют бизнес-процессы, которые BPM-система может выполнить полностью автоматически, запустив ряд специализированных программ. Например, интернет-провайдер при активации нового клиента должен создать для него учетную запись на сервере, добавить информацию в системную службу имен, в конфигурационные файлы веб-сервера и электронной почты и наконец, сформировать счет и отправить его пользователю вместе с уведомлением об активации сервиса. Каждая операция выполняется отдельной программой (в идеале через стандартизованный интерфейс – веб-сервис), а BPMS играет роль планировщика.

Наиболее распространен тип бизнес-процессов, предполагающий как стыковку со специализированными приложениями, так и участие персонала. Например, сотрудник финансового отдела должен зарегистрировать факт оплаты в ERP-системе как шаг бизнес-процесса реализации товара. Этот сценарий требует разработки интерфейсных программ, работающих и с контекстом бизнес-процесса (т.е. с его реквизитами), и с внешней прикладной программой или базой данных. В контексте бизнес-процесса сохраняются ссылки – номер платежки, код контрагента – по которым развернутую информацию можно извлечь из внешнего приложения или базы данных на следующих шагах бизнес-процесса.

Разработка таких комплексных приложений обычно – самая трудоемкая часть проекта BPM.

Наконец, существуют шаги бизнес-процессов, автоматизировать которые невозможно или очень сложно. Например, бизнес-процесс может включать рытье канавы – ясно, что это задача не для компьютера. В такой ситуации BPM-система подаст пользователю сигнал о том, что ему поручено определенное задание, и будет ждать от него подтверждения о выполнении.

Ключевой элемент интерфейса пользователя BPM-системы – «персональный список задач», перечень шагов запущенных экземпляров бизнес-процессов, назначенных данному конкретному пользователю или ролевой группе, к которой он принадлежит.

Благодаря такой организации работы исполнителю за компьютером не приходится думать, с какой функцией и какого именно внешнего приложения ему пора работать: он видит перечень назначенных ему заданий, и когда он берет очередное задание себе на исполнение, нужная программа запускается автоматически. BPM-системы предоставляют доступ через веб-интерфейс, что



позволяет максимально легко вовлекать в коллективную работу сотрудников территориально удаленных подразделений и организаций-контрагентов.

### Функции BPMS

BPMS-системы позволяют организациям управлять (определять, исполнять, контролировать, мониторить и совершенствовать) бизнес-процессами независимо от внедренных в ней бизнес-систем (ERP, SCM, CRM и т.д.).

BPMS-системы оркестрируют бизнес-процессы т.е. исполняют их от начала и до конца, пытаясь объединять действия в рамках процесса. Workflow-сервер BPMS-системы передает работу от одного исполнителя (конечного пользователя) другому. Таким образом, исполнитель знает, в какой момент времени какая работа ему назначена, какой приоритет присвоен этой работе и в какой срок она должна быть выполнена. Таким образом, BPMS-система управляет действиями и процессами. С другой стороны, ERP системы – это транзакционные процессные системы, которые автоматизируют транзакции и осуществляют интеграцию данных между различными функциями, но не справляются с оркестровкой бизнес-процессов от начала и до конца.

BPMS системы дают контроль над бизнес-процессами и повышают производительность. Исполнение бизнес-процесса в «не-BPMS» среде не обеспечивает наглядность ни исполнителям процессов, ни ответственным. Как сказано выше, исполнители процесса не знают, когда работа им назначена, им также неизвестны приоритет и срок исполнения назначенной работы. Подобным образом ответственный за процесс не подозревает о задержках, исключительных случаях и т.д. Тенденция такова, что ответственный исполнитель реагирует «вдогонку» уже произошедшим событиям – задержкам или внешним сигналам, например, звонкам заказчика или директора по продажам по поводу задержки отгрузки или звонкам от поставщика по поводу задержки платежа.

BPMS-системы передают работу от одного исполнителя к другому. Они могут распределять нагрузку между группами исполнителей. Они напоминают исполнителю, как только ему назначена работа, и предоставляют ему информацию о приоритете и сроке, в который она должна быть выполнена. Они также сигнализируют исполнителю о том, что задание не выполнено в отведенное для него время. BPMS-системы могут извещать ответственного и инициировать эскалацию задания ответственному за процесс, если оно не выполнено. Ответственный может легко проследить движение каждого экземпляра процесса и выполнить корректирующие действия или заблаговременно предупредить исключительную ситуацию. Подобная функциональность позволяет организациям исключать задержки и неэффективность, в итоге снижая издержки.

BPMS системы делают возможным управление сквозными процессами предприятия. Бизнес-процессы можно определить как серию шагов (транзакций) выполняемых с целью преобразования входов в выходы. Бизнес-процесс имеет дело с логикой потоков информации, логикой потоков управления и транзакционной логикой. Логика потоков управления определяет последовательность действий, логика потоков информации определяет входы и выходы информации, требуемой на каждом шаге, а транзакционная логика определяет что мы получим на выходе.

Возьмем в качестве примера процесс «Заказ на закупку», который состоит из следующих шагов:

- Оформление заявки продавцом в офисе.
- Оформление документов на отгрузку работником склада.
- Оформление счета бухгалтером.

Этот процесс требует дополнительного кредитного контроля от продавца в офисе. Например, если заказ на закупку превышает кредитный лимит, то требуется специальное одобрение менеджера по продажам, прежде чем на складе выпишут отгрузочные документы. Менеджер по продажам, для того, чтобы принять решение, нуждается в информации о деталях заказа, кредитном лимите и кредитной истории прежде, чем он это решение примет.

Как часть транзакции создания документа Заказ, продавец должен рассчитать стоимость заказа, исходя из продукта, количества, цены, скидок, и применяемых налогов и пошлин. Продавец также должен учитывать данные о наличии ассортимента, производственном графике и графике доставки. В процессе «Заказ на закупку» логика потоков управления определяет действие после шага Оформление заказа, которое может быть либо Оформление документа на отгрузку, либо Получение санкции менеджера по продажам, зависящей от результатов кредитного контроля. Информационная логика определяет, какая информация должна быть предоставлена и какая получе-

на от менеджера по продажам. И транзакционная логика определяет стоимость заказа и время доставки или решение менеджера по продажам.

С другой стороны BPM-системы позволяют пользователям легко построить логику потоков управления и информации. Таким образом, они предлагают слой автоматизации и управления процессами, независимый от информационных систем предприятия, таких как ERP, SCM или CRM. Это дает организациям возможность управлять сквозными процессами предприятия.

Однако BPMS системы имеют ряд недостатков. Во-первых, то, что называется процессом в BPMS, с точки зрения методологии процессного управления не более чем фрагмент сквозного кросс-функционального процесса. BPMS плохо умеют работать на уровне выше отдельного процесса: на каждой диаграмме моделируется только один процесс и связи его с другими процессами, а возможности показать совокупность процессов, составляющих единое логическое целое нет. Если сравнить это, например, со средствами разработки баз данных, то там обычно есть общий репозиторий таблиц (с полями, индексами и прочим), из которых можно складывать разные диаграммы, а в BPM-системах такая возможность отсутствует.

Во-вторых, нужен репозиторий не только процессов, но и информационных объектов. Взаимодействие процессов на самом верхнем уровне удобно (и правильно с точки зрения процессной методологии) описывать в терминах ресурсов, которые один процесс производит, а другой – потребляет. Однако в BPMS информационные объекты трактуются как малозначащие подробности реализации процесса.

Аналогичная картина еще с целым рядом артефактов: бизнес-правилами, KPI, организационной структурой, сервисами.

#### Сравнительный обзор BPM систем

Обзор рынка BPM-систем позволил выявить несколько основных разработчиков к которым можно отнести коммерческие реализации от фирм ActiveVOS, Intalio и Oracle.

Однако существуют не только коммерческие «движки исполнения бизнес-процессов», но и аналогичные продукты, распространяемые свободно, такие как JBPM от JBoss и BPEL SE от фирмы Sun. Это позволяет приблизить людей бизнеса к автоматизированным системам, сократить время и затраты на автоматизацию и т.п.

С точки зрения архитектуры системы, все продукты указанных производителей соответствуют архитектуре приведенной на рис 2.1. Основные различия касаются стоимости, поддержки стандартов BPEL, используемого дизайнера, а также других компонент на которые опирается продукт, например тип СУБД. И если BPM - движки от фирмы Oracle ориентированы естественно на свои и СУБД и сервер приложений, то в остальных случаях можно попытаться выбрать подходящую комбинацию СУБД – сервер приложений исходя из корпоративных интересов.

В архитектуре BPM-системы базовым элементом является сервер приложений и фирмы-производители стремятся продвигать его в стандартной комплектации, так для ActiveVOS это Tomcat, JBPM – JBoss, Sun – GlassFish. Аналогичным образом обстоят дела и с СУБД, которые, как правило, поддерживаются все основные типы.

Намного хуже обстоят дела при выборе дизайнера для разработки бизнес- процессов. Здесь у пользователей нет выбора, практически все в качестве базового дизайнера используют Eclipse и графическую нотацию для отображения процессов специально разработанную под этот дизайнер. Следует признать, что выбор дизайнера является делом субъективным, однако по нашему мнению, дизайнер от фирмы Sun на базе NetBeans является удобным и более предпочтительным по следующим причинам:

- Наглядная связь процесса с внешними сервисами (partnerLink).
- Инструментарий, использующий drag-and-drop для реализации операции связываний (assign), что делает этот процесс очень удобным и наглядным.
- Удобные механизмы импорта сервисов, что позволяет использовать довольно просто переменные находящиеся во внешних пространствах имен.
- Удобство WSDL дизайнера, что позволяет довольно просто добавлять элементы, атрибуты, комплексные типы, операции, связывания.

С точки зрения поддержки стандартов, сегодня ситуация такова, что бизнес процесс без участия человека мало интересен и довольно неплохо стандартизован, что отражено в стандартах спецификации BPEL 1 и BPEL 2. На передний план выходят процессы, в которых человек должен

принимать непосредственно управленческое решение, специально для таких процессов разработана спецификация BPEL4PEOPLE. На наш взгляд, в этом безусловным лидером является фирма ActiveVOS, которая также является одним из главных разработчиков и поставщиков этого стандарта.

В терминологии фирмы ActiveVOS, задача в которой решение принимает человек называется Human task. Эта задача имеет потребителей и поставщиков и соответственно ориентирована на определенную группу пользователей, которые могут выполнять различные роли, например исполнитель процесса, владелец, аналитик и т.п. Все это указывается в BPEL файле. После чего создается либо типовая пользовательская задача, либо специфичная для какого-то конкретного случая. Кроме того с процессом может быть связано несколько сервисов: оповещения, идентификации, почтовый. Это позволяет фиксировать события наступления определенных этапов задачи с оповещением всех заинтересованных лиц посредством, например отправки сообщения на e-mail.

Клиентский уровень BPM-системы должен отражать то, ради чего собственно и планировалось внедрение – мониторинг состояния процессов. Эту функциональную возможность принято называть – Business Activity Monitoring (BAM). BAM позволяет выполнять всесторонний статистический анализ по процессам, позволяя выявлять проблемные места процесса в целом, на основании чего можно выполнить реинжиниринг процесса оптимизируя его параметры. Как правило, BAM реализован в виде web-приложения в BPM системе. Однако предоставляемый многими BPM движками API-интерфейс системы позволяет реализовать его исходя из конкретных задач самим заказчиком системы. Также, некоторые системы допускают использование сторонних разработок. Так, например, в JBPM от JBoss проанонсировали возможность использования разновидности BAM системы - SAS от Google.

Реализацию остальных клиентских компонент нам удалось рассмотреть только на базе решения от ActiveVOS в котором удобно реализована возможность посредством Web-интерфейса выполнять обслуживание всей системы. К сожалению, в свободно распространяемом движке JBPM обслуживание системы в целом сводится к обслуживанию СУБД, используя для этого ее штатные средства.

Таким образом, достоинство модели бизнес-процессов по сравнению с «моделями компонентов Системы» (к которым нас приучил язык UML) состоит в том, что модель бизнес-процессов создается на другом, более высоком, уровне абстракции и позволяет бизнес-аналитикам и клиентам непосредственно участвовать в развитии Системы во время её промышленной эксплуатации, работая в команде на своём уровне понимания (на бизнес-уровне Системы). В данном случае, для внесения в Систему достаточно большой группы изменений (тех изменений, которые относятся к уровню бизнеса и его логики) Клиенту уже не нужно самому быть программистом или использовать программиста в качестве переводчика его мыслей на язык машины (и, наоборот: с языка машины на язык бизнеса).