

**Министерство образования Украины  
Запорожский государственный университет**

**К защите допущена  
Зав. кафедрой ММИТ**

**Борю С.Ю  
7.06.1999**

**Дипломная работа**

**на тему " Разработка функциональной модели подразделения  
университета в рамках Единого Информационного  
Пространства "**

<b>Выполнила ст. группы</b>	<b>8224-2 7.06.1999</b>	<b>Ермолаева Е. В.</b>
<b>Руководитель доцент</b>	<b>7.06.1999</b>	<b>Ермолаев В. А.</b>
<b>Нормоконтролер</b>		<b>Бакурова А. В.</b>

**Запорожье  
1999**

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>1 НЕОБХОДИМАЯ ИНФОРМАЦИЯ, ЗАМЕЧАНИЯ, ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ И ПОСТАНОВКА ЗАДАЧИ</b>	<b>10</b>
1.1 <b>Дополнительные сведения</b>	<b>11</b>
1.1.1 Корпоративные сети. Подходы к организации сетей Intranet	13
1.1.2 Концепция Виртуального Университета	18
1.1.3 Архитектура Единого Информационного Пространства	19
1.3 <b>Теоретические аспекты математического аппарата, используемого при построении данной функциональной модели</b>	<b>24</b>
1.3.1 Диакоптика	24
1.3.2 Конечные автоматы [20, !!!!!!!]	32
1.4 <b>Постановка задачи</b>	<b>37</b>
<b>2 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ВИРТУАЛЬНОГО ПРЕДПРИЯТИЯ НА ПРИМЕРЕ ИЗДАТЕЛЬСКОГО ЦЕНТРА С АКЦЕНТОМ НА ФУНКЦИИ "ЗАКАЗ НА ИЗГОТОВЛЕНИЕ ПОЛИГРАФИЧЕСКОЙ ПРОДУКЦИИ"</b>	<b>38</b>
2.1 Структура и функции издательского центра	39
2.2 Схема функции принятия заказа на изготовление полиграфической продукции	43
2.3 Агенты, их роли и политики в рассматриваемой модели процедуры принятия заказа	48
2.4 Типы воздействий и реакций	49
<b>3 ФОРМАЛЬНАЯ МОДЕЛЬ СООБЩЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ</b>	<b>51</b>
3.1 Модель агента как основного структурного элемента сообщества интеллектуальных агентов	51
3.2 Модель взаимодействия агентов	58
3.3 Состояния и эволюция агентов	60
3.4 Модель сообщества агентов	60
3.5 Анализ модели на примере конкретно взятого заказа	61
<b>ВЫВОДЫ</b>	<b>69</b>
<b>ПЕРЕЧЕНЬ ПРИНЯТЫХ В ДИПЛОМЕ СОКРАЩЕНИЙ</b>	<b>71</b>
<b>СПИСОК ССЫЛОК НА ЛИТЕРАТУРУ</b>	<b>72</b>

## ВВЕДЕНИЕ

XX век примечателен не только тем, что он кровавый, атомный, электрический и экологически загрязненный. Одной из самых ярких и не таких печальных его характеристик является то, что этот век стал мансардой бурного развития НТР. И, главным образом, наше столетие по праву может считаться родителем и опекуном развития электроники, кибернетики, ЭВМ, теории информации, моделирования и всего того, что так или иначе связано с автоматизацией и новыми технологиями. Можно с уверенностью сказать, что мы стали одними из первых непосредственных очевидцев, которым посчастливилось воочию увидеть и ощутить на себе практически полную автоматизацию окружающего мира. Конечно, еще рано говорить о 100-процентной автоматизации всего и вся на сегодняшний день в окружающем нас социуме. Но если делать хотя бы примерный сравнительный анализ уровня влияния машин и различного рода автоматов сегодня и, скажем, лет 80 назад, то можно с уверенностью сказать, что эта разница не просто видна, а она разительна.

Развитие ЭВМ с самого начала своего появления тесно сосуществовало с другими понятиями, окружающими и ЭВМ, и человека. Генезис окружающего мира стимулировал развитие ЭВМ и всех новых научных направлений, тесно с ним связанных, что в свою очередь способствовало общему развитию цивилизации.

Все это привело к тому, что на сегодняшний день в полной или хотя бы частичной автоматизации нуждаются не столько отдельные части подразделений, предприятий и корпораций (например, отдел бухгалтерии некоторого предприятия), сколько встала проблема моделирования всего целостного и объемно представленного предприятия, которое может включать в себя не одну подструктуру, и которое является огромной иерархической системой.

Сегодня наибольшую ценность имеет не золото, как это было, скажем в XIX веке. Самый большой вес в наше время имеет информация. Рядом с ней граничит не

менее важная ценность - время доступа к информации. Как говорится, кто первый владеет информацией, тот и правит миром.

Отсюда напрашивается вывод, что сегодня очень актуальна проблема **моделирования информационной системы, представляющей собой некое предприятие, являющееся собранием многих подразделений и подчиняющиеся единому центру. Все эти подразделения должны быть тесно связаны между собой и откликаться на запросы друг друга в предельно сжатые сроки.** Таким образом, появляются понятия корпоративной информационной сети или сети интранет, единого информационного пространства, распределенной информационной системы и виртуального предприятия, которое может быть смоделировано в информационном пространстве.

В настоящее время исследователи уделяют пристальное внимание проблемам и методам, связанным с моделированием виртуальных предприятий [1]. Повышенное внимание к этим проблемам объясняется как ростом процессов распределения и виртуализации производственных и бизнес процессов (хорошим примером может служить стремительное развитие интернет-магазинов, других средств оказания реальных услуг виртуальными методами, например, электронной коммерции и др.), так и необходимостью придания существующим моделям автоматизированного управления реальными предприятиями недостающего в них динамизма.

И для нас, живущих в конце тысячелетия и оптимистически смотрящих в будущее, очень важно правильно понимать и оценивать тенденции будущего развития цивилизации, найти свое место и внести свой профессиональный вклад в это развитие. Широкое использование информационных технологий, построение информационного общества на нашей планете, интенсивные процессы интеграции во многих гуманитарных, технологических и научных направлениях, глобализация информационных обменов - вот тот базис, который придает все большую актуальность и все большее значение исследованиям в области интеллектуальных информационных агентов и практическому применению результатов этих

исследований в деле интеграции и повышения мобильности человеческого сообщества.

Виртуальное предприятие является одной из сфер применения интеллектуальных информационных агентов. Оно представляет собой временный консорциум автономных, возможно, географически рассредоточенных организаций, которых объединяют их ресурсы и решение каких-либо общих задач. При этом понятно, что виртуальное предприятие обязано использовать быстро изменяющиеся рыночные тенденции, а составляющие виртуального предприятия компонуются с помощью электроники, гарантирующей быстрое, легкое и гибкое взаимодействие и сотрудничество отдельно взятых подразделений предприятия.

Этот вид предприятия получил право на жизнь благодаря свойствам рассредоточения объекта, развития технологий и массового внедрения интернет в мировую экономику и жизнь отдельно взятого человека.

Кроме проблемы моделирования виртуального предприятия, другой не менее важной проблемой является задача создания комфортабельной среды обитания лицам, действующим внутри моделируемой динамической системы - субъектам, выполняющим определенные функции.

В университете для моделирования реального или виртуального предприятия будем использовать способ представления сложной динамической системы виртуального предприятия в терминах интегрированной информационной системы [2], архитектура которой базируется на стандарте CORBA[3].

В современной библиографии по проблемам моделирования виртуальных предприятий можно обнаружить достаточно интересные идеи и методы создания подобных сред. Одним из направлений, например, является использование логики одновременно выполняющихся транзакций (Concurrent Transaction Logic), лежащей в основе Системы Управления Виртуальными Предприятиями (СУВП) (Virtual Enterprise Management Systems (VEMS)) для моделирования и формальных выводов взаимодействий в модели виртуального предприятия [4]. Этот способ моделирования ориентирован на создание формального аппарата описания и вывода взаимодействий во временных сообществах объектов, образующих динамическую

систему (виртуальное предприятие) для достижения какой-то цели. Заслуживают внимания и другие методы функционального моделирования реальных предприятий, построенные на базе аппарата (framework), базирующегося на понятиях роли (role), политики (policy) и взаимодействия (relationship). Одним из важных примеров реализации такого подхода является ICRF (Imperial College Role Framework) [5]. Интенсивно развивается и направление, связанное с построением формализмов для информационных потоков моделирующих взаимодействия и взаимоотношения в сообществах функциональных элементов виртуальных предприятий на базе парадигмы кооперирующихся агентов (co-operative agents). Примером построения такого формализма может служить [6].

Уже понятно, что данная работа будет ориентирована на моделирование предприятия. Однако, следует упомянуть, что абсолютно любое предприятие можно рассматривать с нескольких точек зрения, т.е., в нескольких проекциях: топологической, организационной (структурной), и функциональной.

В своей работе уделим внимание лишь одной из них – функциональной. Она призвана на решение следующих задач:

- a) Какие сервисы и информационные ресурсы предоставляются комнатой N, подразделением D или сервером S?
- b) Какие элементы предприятия дадут нам ответ на запрос R?
- c) Какую информацию мы можем, а какую - не можем рассчитывать получить?

Следует упомянуть тот факт, что отмеченные проекции модели очень тесно связаны друг с другом и образуют пересечения по многим направлениям.

В качестве предприятия для описания функциональной проекции мы избрали Запорожский Государственный университет. Общеуниверситетскую сеть является разнородной распределенной системой и включает в себя на разных уровнях различные типы аппаратных и программных компонентов. Для упрощения нашей задачи попробуем смоделировать функциональную проекцию лишь одного конкретно взятого подразделения Запорожского Государственного университета,

скажем, недавно созданного издательского центра. И, более того, исследуем одну какую-либо функцию, реально осуществляемую данным центром.

**Новизна данной работы** заключается в том, что никогда прежде и нигде ранее разрабатываемая нами модель не была применена по отношению к такому подразделению, которое избрали мы - издательскому центру. Естественно, не исследовалась ранее и избранная нами функция<sup>1</sup> для анализа данной модели.

Попробуем теперь ответить на вопрос, **зачем и кому нужно** то, чему посвящена эта дипломная работа.

Результаты исследований по данной теме могут уменьшить пробелы в наших знаниях и умении в деле реализации таких сложных и практически полезных систем, как виртуальные информационные пространства и виртуальные предприятия (это шаг вперед по пути к Информационному Сообществу).

Очень не маловажен тот факт, что данные результаты широко востребованы в областях, начиная с высшего образования и заканчивая индустрией и коммерцией. А если смотреть более масштабно на результаты моделирования Единого Информационного Пространства и его функциональной проекции в частности, то можно сказать, что некоторые выводы, сформулированные в этом дипломе будут содействовать дальнейшей интеграции Восточных (а также Западных) стран в объединенное Информационное Сообщество и, таким образом, способствовать делу снижения уровня напряженности, непредсказуемости и повышения уровня комфорта во взаимоотношениях между Восточными и Западными соседями.

Кроме всего прочего Интеллектуальные Виртуальные Информационные Пространства могут применяться как эволюционирующая самообучающаяся среда для:

- поддержки процессов подготовки и принятия решений (улучшая, таким образом, процедуры управления в широком спектре прикладных областей), использования в качестве персональных ассистентов для выполнения рутинных функций

---

<sup>1</sup> Для анализа избранной модели функциональной проекции мы выбрали такую часто реализуемую функцию, как принятие заказа на изготовление полиграфической продукции. О причинах выбора такой функции будет сказано ниже.

(интенсифицируя, таким образом, использование рабочей силы и снижая риск человеческих ошибок).

– использования в качестве персональных ассистентов для дистанционного обучения и руководства студентами (интенсифицируя, таким образом, процесс получения образования).

Разработанный нами аппарат в большей степени применим к моделированию функциональных элементов УИП, но не процессов - информационных потоков, координации работ, сетевого планирования и т.д., хотя, по нашему мнению, он и имеет все необходимые средства для моделирования процессов.

Еще одной отличительной чертой данной модели является учет динамического характера функциональной проекции УИП. Для создания адаптивных эволюционирующих моделей функциональных блоков УИП могут применяться диакоптика, концепция кооперирующихся интеллектуальных агентов, методы семантической кластеризации данных и элементы теории конечных автоматов.

Мы предлагаем применить подходы, базирующиеся на применении интеллектуальных кооперирующихся агентов, к построению и реализации функциональной проекции нашего Унифицированного Информационного Пространства.

Кроме этого, глобальными целями для нашего университета по использованию данной дипломной работы являются:

- a) После реализации виртуального информационного пространства для среды университета применить полученные результаты к улучшению и интенсификации процедур управления университетом, а также для поддержки процессов подготовки и принятия решений.
- b) А также применить полученные результаты для создания новой электронной среды обучения: дистанционное обучение, новые методы преподавания, новые формы учебных пособий и т.д.

- c) Распространить полученные результаты в нескольких университетах СНГ и, возможно, в некоторых правительственных учреждениях Украины<sup>2</sup>.
- d) Попробовать исследовать пути помощи странам СНГ в деле интеграции в Информационное Сообщество при помощи широкого использования парадигм и технологий Интеллектуальных Агентов и Виртуальных Информационных Пространств.

При этом понятно, что достижение данных целей будет способствовать:

- улучшению качества образования (средства дистанционного обучения, глобальный распределенный доступ к библиотечным ресурсам, разработка электронных учебников и других пособий, реализация активных процедур и средств обучения и тестирования) и, таким образом, подъему культурного уровня, увеличению профессиональной квалификации, мотивации поколения будущего;
- повышению эффективности и современности процедур управления предприятием/университетом (поддержке процессов подготовки и принятия решений, интенсификации выполнения рутинных функций) и, таким образом, помощи в придании экономике Украины черт большей предсказуемости, рыночности, постоянного роста и большей демократичности.

И в конце заметим, что **целью данной дипломной работы** является:

- формализовать функциональный элемент сообщества;
- определить правила взаимодействия функциональных элементов (агентов);
- описать всевозможные множества состояний и эволюцию элементов модели;
- разработать формальное представление сообщества интеллектуальных агентов моделирующего функциональное подразделение предприятия, используя уже построенную модель отдельно взятого агента, т.е. модель агента распространим на сообщество в целом.

---

<sup>2</sup> Подобные задачи были поставлены на всеукраинской конференции EDUNET 97.

# 1 НЕОБХОДИМАЯ ИНФОРМАЦИЯ, ЗАМЕЧАНИЯ, ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ И ПОСТАНОВКА ЗАДАЧИ

Как уже упоминалось во введении, университетская сеть является разнородной распределенной информационной системой. Поэтому в первой главе постараемся осветить вкратце понятие именно университетской корпоративной сети. Понятие распределенной информационной системы в том понимании, как это себе представляет отечественная библиография представлено в [7]. Следует упомянуть, что на сегодняшний день в нашей стране, как впрочем, и на территории всего бывшего Советского Союза, изданий по данной тематике практически не существует. А то, что есть не передается массовому распространению, поскольку данная тема представляет коммерческий интерес, как за границей, так и у нас. Однако, в 1998 году в периодическом издании журнала "Управляющие Системы и Машины" под рубрикой новые методы в информатике была напечатана статья В.И. Гриценко, А.А. Щипцова, А.В. Ищенко, Ю.Г. Авдюшенко, В.Н. Никулина, С.А.Тарасенко, А.А. Урсатьева "Современные подходы к созданию распределенных информационных систем". В этой статье рассмотрено применение Интернет-технологий для проектирования информационной системы Государственной гидрографической службы Украины.

Среди большого разнообразия современных информационных систем (ИС) можно выделить достаточно обширный класс ИС, характерной особенностью которых является интеграция в одной системе реляционных и гипертекстовых баз данных. Решение подобных задач стало возможным при тесном взаимодействии Интернет-технологий с новейшими высокопроизводительными реляционными СУБД (Informix, Oracle, Sybase), которые представляют свои во многом уникальные способы хранения и управления потоками данных. По существу речь идет о публикации информации из реляционных баз данных по запросу пользователя в виде динамически формируемых Web-страниц (Web-документы, расположенные на WWW-серверах, представляют собой тексты в стандарте ASCII, содержащие команды специального языка HTML. Они могут просматриваться различными

типами WEB-броузеров. Использование HTML позволяет форматировать документы для их представления с использованием шрифтов, линий и других графических элементов на любой системе, их просматривающей. HTML-документы могут быть созданы при помощи любого текстового редактора или специализированных HTML-редакторов и конвертеров. Например, HTML редакторы, такие, как "Netscape Navigator Gold" компании Netscape позволяют создавать документы графически с использованием технологии WYSIWYG (What You See Is What You Get). С другой стороны, большинство традиционных средств для создания документов имеют конвертеры, позволяющие преобразовывать документы к формату HTML. Если вас интересует более подробная информация, то рекомендую следующие адреса в Интернет, используя которые вы сможете получить необходимую информацию: <http://www.ucc.ie/~pflynn/books/htmlcard.html> – краткий справочник по html, <http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.htm> - базовая информация по html.). Однако, непосредственная реализация гипертекстовых структур в СУБД реляционного типа приводит к избыточности создаваемой базы и усложняет установки гиперсвязей между полями.

Одним из подходов к построению ИС такого класса, оптимизирующих показатель "возможности/затраты", может быть применение современных Интернет-технологий. Использование при проектировании апробированных компонентов в значительной степени сокращает сроки создания крупных ИС и их интеграции в мировое пространство.

### **1.1 *Дополнительные сведения***

Запорожский университет в настоящее время осуществляет программу модернизации вычислительных средств и перехода к современным вычислительным конфигурациям сетевым и системным решениям [8].

Сейчас в ЗГУ работают локальные сети, построенные на базе персональных рабочих станций IBM PC 486 и Pentium, которые обслуживаются WINDOWS, WINDOWS NT, NOVELL и UNIX серверами, предоставляющими распределенные

локальные ресурсы и выход в глобальную сеть **INTERNET** (это всемирная компьютерная сеть, объединяющая многие глобальные сети. Иначе говоря, Internet – сеть сетей, опутывающих весь земной шар. Электронная почта, телеконференции (сетевые новости), FTP, WWW, - виды услуг, предоставляемые всемирной компьютерной сетью.). Компьютерная сеть университета создается как корпоративная сеть, включающая в себя различные элементы оборудования и организационные компоненты. Она включает в себя несколько локальных сетей, построенных на базе Novell Netware 3.11, MS Windows NT 4.0, MS Windows for WorkGroups 3.11 и MS Windows 95.

При рассмотрении перспектив развития систем компьютерной обработки данных в Запорожском университете делается акцент на следующих важных направлениях:

- полное оснащение научно - исследовательских подразделений университета современной компьютерной техникой с учетом необходимых функциональных возможностей и интеграции в университетскую сеть;
- создание интегрированной общеуниверситетской вычислительной системы на базе современных архитектурных решений с использованием распределенных и интегрированных информационных и вычислительных ресурсов;
- дальнейшее расширение и развитие каналов связи и компьютерных сетей;
- охват полного спектра операционных систем для полного представления в учебном процессе классического университета всего спектра современных программных средств;
- внедрение технологий INTERNET/INTRANET в учебный процесс и управление ЗГУ.

Компьютерная сеть университета создается как корпоративная сеть включающая в себя различные элементы оборудования и организационные компоненты. Компьютерная сеть ЗГУ включает в себя несколько локальных сетей, построенных на базе Novell Netware 3.11, MS Windows NT 4.0, MS Windows for WorkGroups 3.11 и MS Windows 95.

Таким образом, необходимо более подробно остановиться на понятии корпоративной сети и всего, что с ней связано.

### 1.1.1 Корпоративные сети. Подходы к организации сетей Intranet

К середине 90-х годов все в компьютерном мире казалось совершенным, устоявшимся и определенным. Особую популярность приобрели локальные сети персональных компьютеров. Авторитеты ведущих производителей аппаратных и программных средств казались незыблемыми. И вдруг совершенная картина стала терять свою стройность: появились такие имена, как **WWW** (технология World Wide Web. WWW («всемирная паутина») – самый модный и перспективный сервис Internet. Вообще говоря, WWW – это глобальная гипертекстовая система. Под гипертекстом понимают текст с рядом слов (как правило, каким либо образом выделенных), которые выполняют роль рассылок и указывают либо на другие части этого же документа, либо вообще на другой документ. Дальнейшим развитием гипертекста является гипермедиа, т.е. документ, включающий в себя гипертекст и мультимедиа-информацию (графику, звук, видео). Практически любая информация, заслуживающая всеобщего внимания, представляется сегодня средствами WWW. Кроме специальных программ для установления связи с Internet, пользователь для навигации по WWW должен иметь одну из программ типа Mosaic, Netscape Navigator или Microsoft Internet Explorer.), Internet, **Java** (система программирования Java позволяет использовать WWW для распространения небольших интерактивных прикладных программ (апплетов), которые размещаются на серверах Internet, транспортируются клиенту по сети, автоматически устанавливаются и запускаются на месте, как часть документа WWW. При этом апплет имеет весьма ограниченный доступ к ресурсам компьютера клиента. Основным свойством апплетов является возможность выполнять их на различных платформах и в различных окружения, не оказывая вредного влияния на аппаратуру, программы и данные их пользователей. В Java-программу можно включать фрагменты, написанные на других языках и выполняемые в объектном коде). Неожиданно оказалось, что конец тысячелетия ознаменовался радикальной сменой компьютерной парадигмы даже не столько в

технологиях, сколько в умах. Количественные накопления знаний повлекли качественные изменения.

Сейчас технология Internet широко используется для реализации корпоративных информационных систем – Intranet. Различие между Internet и Intranet скорее семантическое, чем технологическое[9]. Обе эти технологии используют практически одни и те же протоколы, продукты и стандарты, однако между ними есть два принципиальных различия.

Во-первых, информация, циркулирующая в Intranet, обычно не предназначена для публичного использования. Поэтому применяются методы авторизации доступа и криптозащиты данных. Во-вторых, функциональные возможности Intranet могут расширяться и дополняться специализированными средствами по мере развития системы в сторону GroupWare в процессе эксплуатации

Еще одно отличие технологии Intranet заключается в том, что она не разрабатывалась специально, но оказалась готовой к моменту, когда в ней возникла потребность. По мнению [9] феномен Intranet может быть охарактеризован, одним словом – **СКОРОСТЬ**. Слово «скорость» применимо к оценке темпов распространения Intranet, к процессу внедрения Intranet в конкретном предприятии и всему, что его окружает.

Что же делает Intranet столь привлекательной, в чем секрет ее успеха? Прежде всего, в том, что в ее родоначальнике – Internet – воплотились лучшие идеи нескольких талантливых, если не гениальных людей. Эти идеи получили неформальное признание мировой компьютерной общественности, пройдя своего рода жестокий эволюционный отбор. То есть, Intranet – это не некий коммерческий законченный продукт, имеющий фиксированное авторство с присущими авторам определенными достоинствами и недостатками, а совокупность взглядов, плод развития в условиях жесткой конкуренции. Представляется возможным к эпитету «скорость» добавить второй **ЕСТЕСТВЕННОСТЬ**. Кроме того, технология Intranet замечательна тем, что привлекает внимание крупных корпоративных пользователей, гигантских транснациональных компаний и, следовательно, охватывает огромный потенциальный рынок. Фактически сегодня сложились два конкурирующих альянса

из крупнейших компьютерных компаний: первый включает AT&T, IBM, Netscape; второй, образованный совсем недавно, - Microsoft, MCI Communications, Digital Equipment Corp.

В двух словах, Intranet – это перенос технологий Internet во внутрикорпоративную деятельность предприятия. По сути своей, это не новая технология как таковая, а новое видение известной технологии.

В основе Intranet лежат идеи гипертекста, впервые высказанные Теодором Нельсоном в начале 60-х годов, а затем развитые в проекте Nanadu, профинансированном компанией Autodesk и окончательно завершившем, свое существование в начале 90-х годов. Однако эти идеи были восприняты в виде концепции WWW Тимом Бернсом Ли. Именно он создал протокол передачи гипертекстов **HTTP** [10] (Hyper Text Transfer Protocol – протокол прикладного уровня, предназначен для распределения и управления ИС, реализующими механизм гипертекстовых ссылок. Он является основным объектно-ориентированным протоколом, который может использоваться для решения задач управления обменом между серверами и объектами распределенных систем с помощью их методов запросов. Основное направление развития HTTP применяется в WWW с 1990 года. Этот протокол базируется на основе парадигмы запрос\ответ. Клиент посылает запросы серверу с указанием метода запроса URI, версии протокола; сообщения передаются в соответствии с MIME-спецификацией, содержащей информацию пользоавтеля, и поля, необходимые для установления соединения с сервером), язык разметки HTML и общую систему адресации – универсальный указатель ресурсов **URL** [10] (Universal Resource Name общая система адресации, универсальный указатель местоположения ресурсов. Частный случай URI – универсального идентификатора ресурсов.). Затем началась эра навигаторов, первым был Mosaіks Марка Андреессена. Навигаторы открыли людям возможность простого удобного доступа к Internet. В 1994 году Джимом Кларком, университетским профессором и основателем Silicon Grapfics, была создана компания Netscape Communications, первоначально названная Mosaic Communications, куда был приглашен Марк Андреессен – признанный «бог

Internet». Это и стало началом развивающейся с ураганной скоростью новой компьютерной индустрии навигаторов. Таковы были предпосылки успеха Internet. А это то, что нужно любому предприятию только под названием Intranet.

Использование технологии WWW естественным образом позволяет создать информационную среду (MIS – Management Information System), которая адекватно отражает все виды деятельности предприятия. Она настолько органична, что для человека, имеющего минимальные представления о средствах Internet не представляет никакого труда решить стоящие перед ним конкретные задачи средствами технологии Intranet. Радикальное новшество Intranet заключается в рационализации потребления информации [11].

Значительное, драматическое по силе воздействие технологии Internet на совершенствование внутрикорпоративного информационного обмена, приведшее к рождению Intranet имеет несколько причин. Среди них [9]:

- **универсальность методов коммуникации:** Intranet – это эффективное взаимодействие между отдельными сотрудниками и подразделениями предприятия и его внешними партнерами;
- **производительность:** Intranet позволяет полноценно использовать существующие сетевые ресурсы предприятий для передачи разнообразной информации, в том числе аудио клипов и видео изображений;
- **стоимость:** Intranet намного дешевле специализированных программных продуктов для автоматизации предприятий;
- **открытость стандартов:** стандартные протоколы и прикладные программные интерфейсы (HTML, **FTP** (протокол передачи файлов ftp (File Transfer Protocol) является популярным сервисом Internet. Он используется для доставки информации в файловых архивах. Как правило, для доступа в компьютер (называемый обычно ftp-сервером), где расположен необходимый вам архив, можно использовать негласное соглашение о пользователе с именем anonymous (аноним). В этом случае вместо имени вводится слово anonymous (или ftp), а вместо пароля – сетевой адрес пользователя), **MIME** [10] (стандарт, разработанный с целью пересылки по Интернет любых типов данных), Windows

Sockets, TCP/IP) гарантируют способность информационной системы к плавной эволюции.

Информационная система предприятия приобретает следующие качества [9]:

- вся информация выдается по запросу тогда, когда в ней есть потребность;
- гарантировано, что вся информация точна и актуальна по времени;
- вся информация поддерживается одним координирующим подразделением (источников может быть много).

Наиболее распространенным способом организации сетей Intranet является создание набора HTML страниц, представляющих те или иные узлы корпоративной системы. Такой подход является неэффективным, так как порождает следующие проблемы [2]:

- неэкономное использование системных ресурсов вследствие необходимости постоянного хранения статического набора HTML страниц;
- отсутствие удобного способа модификации элементов интерфейса, рабочая станция, группа или сотрудник.

В настоящее время при проектировании сетей Intranet применяются два подхода, которые в зависимости от организации способа получения пользователем HTML-документов условно можно назвать статическим и динамическим.

В первом случае источником интерфейса является HTML-документ, созданный в каком-либо текстовом или HTML-ориентированном редакторе. Данный документ храниться в готовом виде и остается неизменным в течение использования. Однако такой подход является весьма неэффективным, поскольку любое изменение в предметной области влечет за собой либо модификацию существующего HTML документа, либо создание нового. Организация же связи с ним представляет собой не самую простую задачу.

Во втором случае источником интерфейса является HTML-документ, сгенерированный CGI-модулем (CGI является стандартом интерфейса (связи) внешней прикладной программы с информационным сервером типа HTTP, Web сервером. Обычно гипертекстовые документы, извлекаемые из WWW, серверов, содержат статические данные. С помощью CGI можно создавать CGI-программы,

называемые шлюзами, которые во взаимодействии с такими прикладными системами, как система управления базой данных, электронная таблица, деловая графика и др., смогут выдать на экран пользователя динамическую информацию. Т.е., CGI – механизм передачи параметров при формировании запроса пользователя к специальным программам (называемым CGI-скриптами или сценариями), которые позволяют быстро и удобно создавать сетевые приложения с выводом в HTML-формате.), вследствие чего появляется гибкость в видоизменении интерфейса во время использования. Динамический подход используется, например, при создании поисковых серверов Internet (Yahoo, AltaVista и др.). В качестве результата на свой запрос пользователь получает динамически созданную HTML-страницу с набором ссылок на документы с интересующей его информацией.

Другим примером использования динамического подхода является организация доступа к существующим хранилищам информации (БД) с использованием WWW-интерфейса. Однако существующие на данный момент подобные разработки не предоставляют пользователю интерфейс, адекватно отражающий предметную область их корпоративной сети, а могут служить лишь средством получения информации справочного характера.

### 1.1.2 Концепция Виртуального Университета

Концепция Виртуального Университета Запорожского Государственного Университета формируется на основе Единого Информационного Пространства Университета (ЕИПУ), описанного в [12]. Единое Информационное Пространство разработано как логическая надстройка над Корпоративной Сетью Университета. Как и любая другая информационная система, Корпоративная Сеть Университета представляет собой сложную разнородную систему, состоящую из элементов различных типов аппаратного и программного обеспечения на разных уровнях. Основная задача ЕИПУ - играть роль однородного виртуального средства, предоставляющего пользователю стандартизованные процедуры доступа и средства

навигации, системному разработчику - необходимые свойства адаптируемости и гибкости, а системному администратору - эффективное средство контроля.

Основное внимание при разработке данного проекта уделяется созданию для пользователя интуитивно понятного интерфейса. Поэтому одним из основных элементов ВУ является Unified Visual Intranet Interface (UVII) – Унифицированный Визуальный Интерфейс Intranet. Современное информационное окружение настолько сложно, что попытка обзора всей доступной информации представляет собой весьма трудоемкую задачу. Решение этой задачи в рамках данного проекта основывается на использовании визуального интерфейса, имитирующего структуру и содержание реальных объектов – университет, корпус (здание), подразделение, аудитория, сервер, рабочая станция, группа или сотрудник. Так как набор этих объектов ограничен, то предлагается разработать объекты интерфейса, связав их с объектами реального мира.

В следующем подразделе будут описаны некоторые аспекты разработки архитектуры ЕИП. Так как главной идеей первого уровня ЕИП является автоматическое генерирование HTML кода для элементов интерфейса, то следующий шаг состоит в получении необходимых данных из БД ЕИП.

### 1.1.3 Архитектура Единого Информационного Пространства

Для описания архитектуры ЕИП рассмотрим его отдельные уровни и интерфейсы между ними. Первым уровнем является уровень Пользователя или **Внешний уровень**. Он был описан в предыдущем разделе. Элементами интерфейса данного уровня являются: запрос Пользователя к ЕИП и результат в форме HTML кода, возвращаемый ЕИП браузеру пользователя. Следующий уровень - **Корпоративная Информационная Система**. Задача этого уровня состоит в том, чтобы преобразовать запрос пользователя в виртуальный запрос к Корпоративной ИС. После обработки виртуального запроса его результат передается Обработчику Результата запроса, который генерирует соответствующий HTML код и возвращает его на Внешний уровень (Уровень Пользователя). Архитектура Внешнего уровня и

уровня Корпоративной ИС схематически показана на рисунке 1.1.

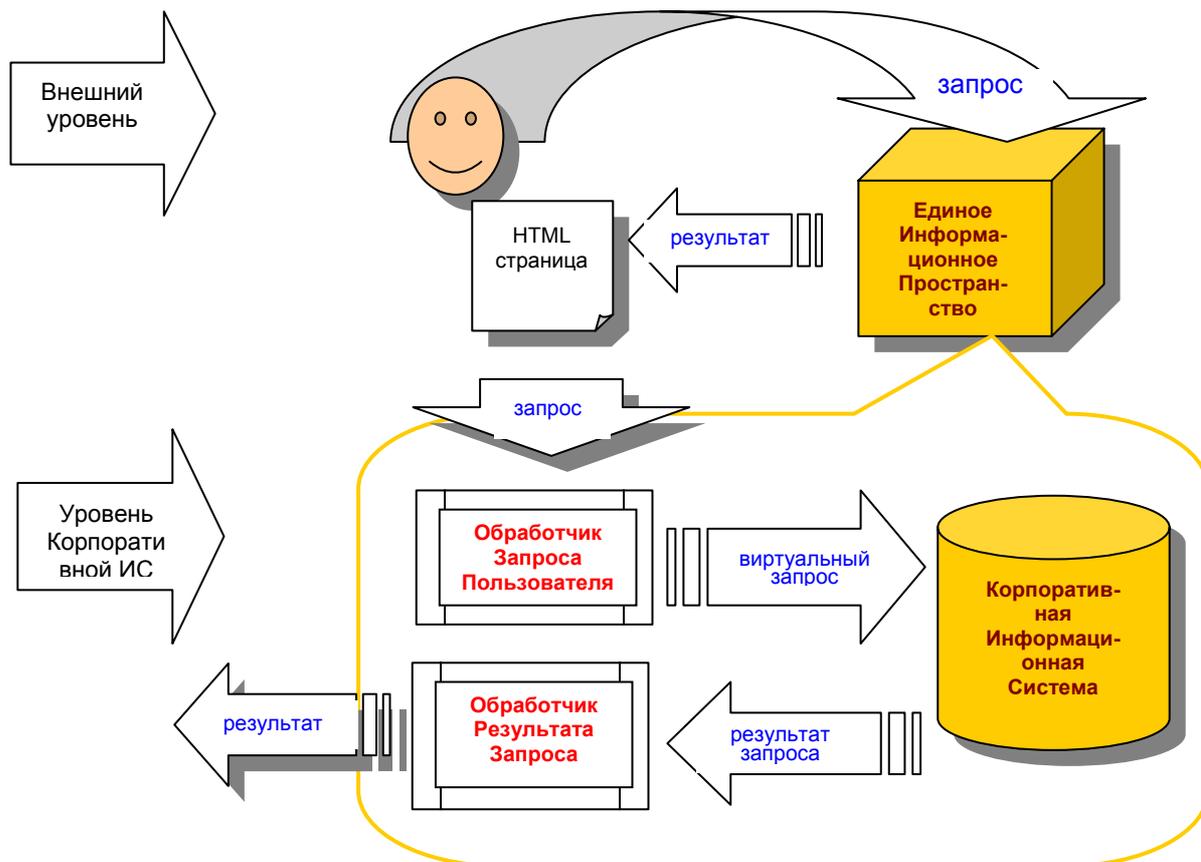


Рисунок 1.1. Внешний уровень и уровень Корпоративной ИС Архитектуры ЕИП.

После того, как запрос пользователя будут преобразован в запрос к КИС, возникает возможность описания UVII в терминах модели данных, объектов и отношений и рассматривать запрос как запрос в смысле СУБД. Для осуществления задач этого уровня архитектуры необходима модель данных для описания элементов интерфейса, СУБД, обработчик запроса пользователя и обработчик результата запроса к КИС.

Далее, при осуществлении виртуального запроса возникает ряд проблем. КИС представляет собой достаточно сложную структуру, включающую ряд Локальных Информационных Систем и Функциональных Серверов, распределенных по всему сетевому окружению и управляющих локальной информацией, приложениями и ресурсами. Данные и функциональные характеристики приложений и ресурсов зачастую имеют семантические пересечения. Одним из методов разрешения этой проблемы является использование Федеративных моделей данных и СУБД. В ЕИП планируется использование метода и модели, разработанных G. Saake и его коллегами [13], с использованием Активных Словарей Данных [14]. Таким образом, Федеративный уровень архитектуры ЕИП может быть представлен схемой, изображенной на рисунке 1.2.

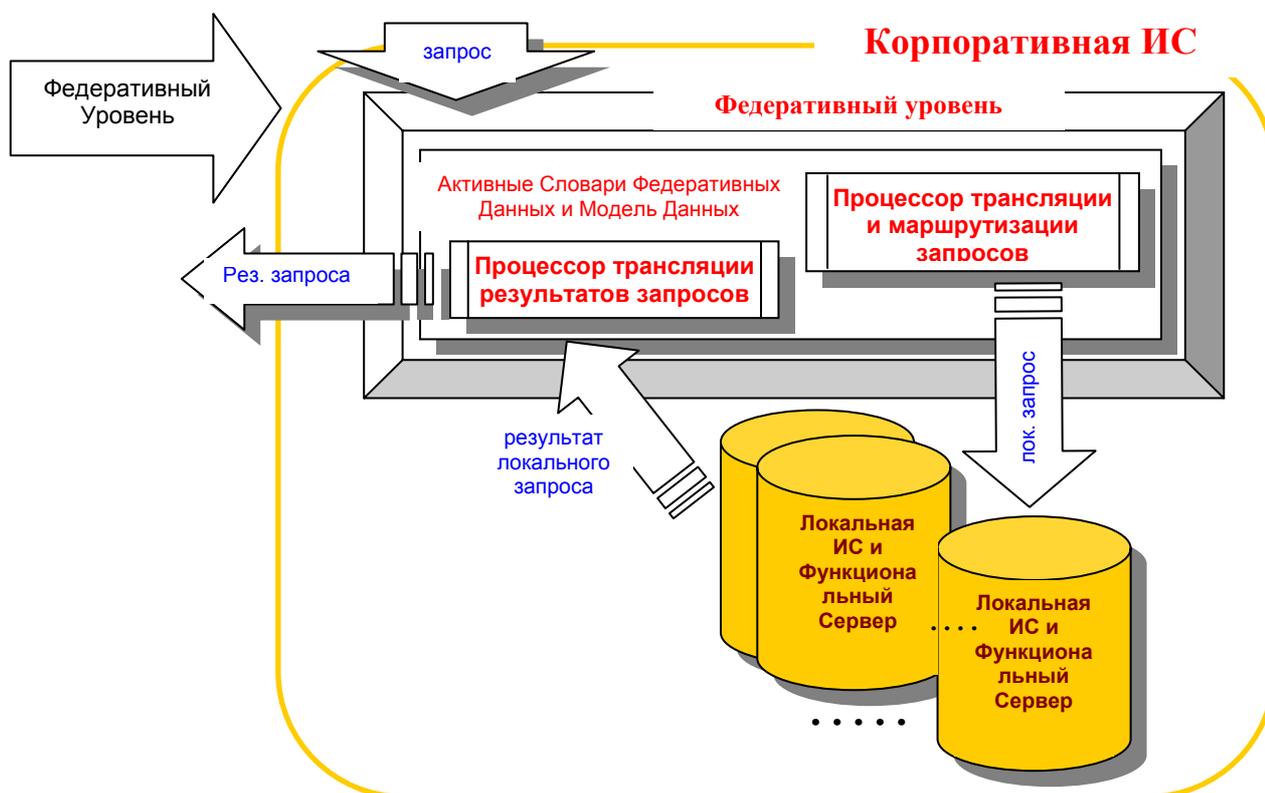


Рисунок 1.2. Федеративный уровень Единого Информационного Пространства.

Этот уровень разделяется на два подуровня с соответствующими интерфейсами – Федеративный сервер и набор серверов локальных информационных ресурсов. Федеративный уровень осуществляет преобразование поступающего виртуального запроса в набор локальных запросов и перенаправляет их к элементам локального

уровня архитектуры.

Главной архитектурной особенностью локального уровня является метод контроля Локальной ИС. Предполагается использование Активного словаря данных в качестве оболочки (shell) контроля ИС и Модели данных в качестве средства (engine) контроля. Таким образом, локальный уровень представлен тремя уровнями: Уровнем Программного Кода, Уровнем Модели Данных и Уровнем Данных (рисунок 1.3). Словарь Данных с его активными функциями, контролируемый Моделью Данных, предоставляет гибкий интерфейс между верхними уровнями и локальными данными и/или ресурсами.

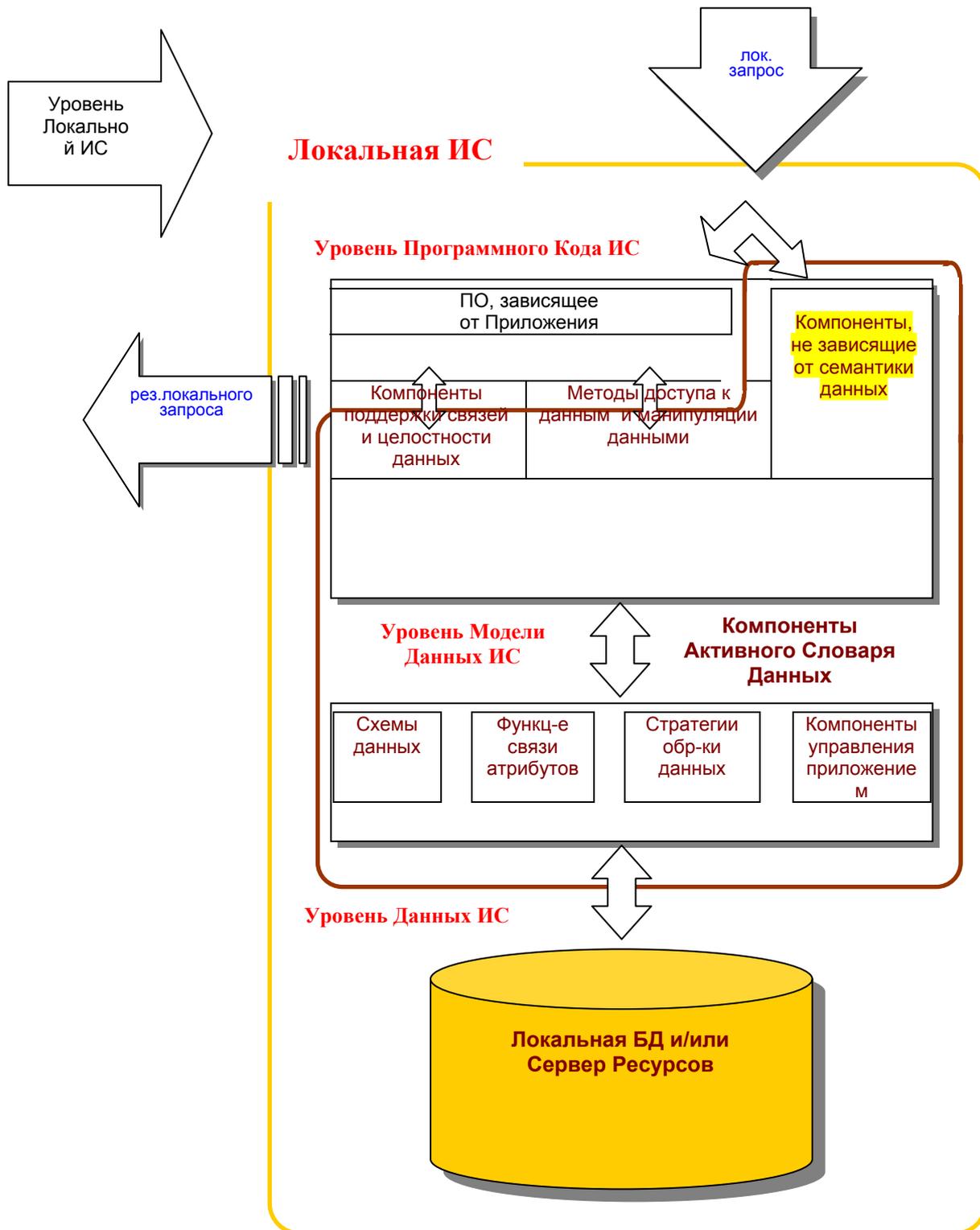


Рисунок 1.3 Уровень Локальной ИС Единого Информационного Пространства.

### **1.3 Теоретические аспекты математического аппарата, используемого при построении данной функциональной модели**

#### 1.3.1 Диакоптика

Для построение функциональной модели мы использовали один из достаточно новых методов исследования сложных систем, разработанный более 40-ка лет назад Г.Кроном [15]. Этот метод носит название диакоптики.

Слово "диакоптика" образовано от греческого "конто", означающего "расчленение" и "диа", которое усиливает слово, стоящее за ним, и может быть интерпретировано как "система". Следовательно, диакоптика означает расчленение как систематический метод.

Как известно, любая распределенная система, для которой возможно построение функциональной модели, несомненно, является сложной, структурированной и с первого взгляда не сразу поддающейся детальному изучению и исследованию. Усложнение систем со временем – одна из главных тенденций в развитии современной науке. Это связано с тем, что с одной стороны, развитие техники и производства требует использования все более совершенных средств, создание которых неразрывно связано с дальнейшим их усложнением, с другой стороны, логика развития науки для более полного познания объективных законов требует принимать во внимание те эффекты, которыми раньше пренебрегали, что также связано с весьма существенным усложнением рассматриваемых моделей реальных объектов и явлений.

Диакоптика же – это один из наиболее общих и эффективных путей исследования таких сложных систем. Она заключается в переходе от исходной сложной системы к исследованию более простых систем, по свойствам которых можно восстановить точно или приближенно свойства исходной системы. Причем, этот метод может быть применен к исследованию систем самой различной физической природы: электрических, механических, экономических, биологических. При этом сложная система расчленяется по определенным правилам на некоторое число малых подсистем, и на основе этого строятся топологические модели – графы исходной системы. Для каждой из подсистем в отдельности

проводится анализ и отыскивается решение. Общее решение получается путем сочленения полученных решений для подсистем. В данном методе используется аппарат тензорного и матричного исчисления.

В общем случае не только Крон занимался изучением и разработкой данного метода применительно к различного рода задачам. Хотя, конечно же, именно Крону принадлежит ветвь первенства по данной проблеме. Кроме того, в течение 35-ти лет он систематически публиковал работы, посвященные развитию предложенного им метода исследования систем ко все более сложным задачам в самых различных областях науки и техники (молекулярная и квантовая физика, механика упругих систем и сплошных сред, электротехника, линейное программирование и т.д. и т.п.).

В Советском Союзе одним из первых использовал разделение системы на части для определения собственных частот Ф.М. Диментберг [16]. Детальной разработке этого метода для цепных систем посвящены работы В.П. Терских, который широко использовал для итоговых выражений цепные дроби и поэтому свой метод назвал методом цепных дробей [17].

Метод цепных дробей обобщен на более широкий класс систем В.К.Дондошанским [18]. Матричные методы в исследовании свойств сложных систем по свойствам их частей использовал В.А. Троицкий [19].

Метод исследования по частям обобщен на системы с распределенными связями в [20]. Среди работ зарубежных авторов можно назвать [21], [22].

За последнее время метод разделения сложных систем на части нашел широкое применение в задачах оптимизации, где он получил название метода блочного программирования [23]. Эта же идея разрабатывается и в приложении к общим задачам управления (декомпозиционное управление) и в теории графов (декомпозиция графов) и в теории автоматов (декомпозиция автоматов).

Таким образом, решение задач по частям в различной форме и в приложении к различным задачам представляет собою широкую область исследований, в которой работают многие ученые в мире.

Остановимся чуть подробнее на данном методе. Диакоптику по-другому еще называют методом расчленений. Почти всегда данный метод безукоризненно

работает с системами, содержащими большое число переменных. Физическая или экономическая система (или ее схематическая топологическая модель) разделяется на соответствующее число малых подсистем, затем каждая подсистема рассчитывается и анализируется отдельно, как если бы остальные подсистемы не существовали, затем частные решения соединяются шаг за шагом до тех пор, пока не будет получено решение для всей системы

При этом описание топологической модели системы проводится на языке электротехники. Модель представляет собою просто вычерченный на бумаге линейный граф, некоторым ветвям которого приписаны значения импедансов.

В такой блок-схеме могут использоваться отрицательные нелинейные и другие произвольные алгебраические функции, так что модель не обязательно физически реализуема.

Расчленяемые системы могут быть линейными и нелинейными, статическими и динамическими.

Одной из целей метода расчленений является преодоление (или по крайней мере облегчение) трудностей, которые возникают при обдумывании задач, выводе уравнений состояния и программировании для вычислительных машин. Это достигается благодаря тому, что в каждый момент времени рассматривается только малая часть данной системы. Размер подсистем диктуется обычно возможностями имеющейся в распоряжении цифровой и аналоговой вычислительной техники или иными внешними условиями, такими как проектные требования или дальнейшее усложнение задачи.

Описываемый метод главным образом применим к решению задач трехмерных физических систем, обладающих такими свойствами, благодаря которым расчленение их оказывается удобным и целесообразным.

Однако Крон вовсе не ограничивает применение метода расчленений расчетом лишь физических систем. В своих работах он обобщил применение данного метода к биологическим, экономическим или каким-либо другим "системам", имеющим большее число переменных и представимых в виде блок-схемы или графа. Возможность построения графа системы является существенной,

так как расчленение проводится непосредственно на графе и только изредка на уравнениях.

Диакоптика полностью использует не только уравнения и матрицы, полученные для системы, но и граф-топологический портрет данной системы как новый источник информации, ускоряющий составление и решение уравнений.

Таким образом, диакоптика как метод расчленений, является теорией, объединяющей два источника информации, а именно: уравнения и графы или матрицы и графы, связанные с данной физической или экономической системой. Граф системы используется также для более эффективного применения вычислительных машин.

Данный метод представляет собою попытку объединить непрерывный анализ (уравнения) с дискретным (графы) для более быстрого и рационального решения сложных задач. Иначе говоря, метод решения по частям позволяет объединить возможности теоретико-множественной топологии (дифференциальных уравнений, теории функций и т.д.) с возможностями комбинаторной топологии (теории цепей) в единый инженерный метод анализа и расчета сложных систем, имеющих большое число степеней свободы. Такое объединение достигается с помощью четырех дисциплин:

- 1) тензорного анализа, поскольку он позволяет описывать сложные явления, исходя из небольшого числа зависимостей (уравнений);
- 2) матричного анализа, поскольку он позволяет при составлении программы для вычислительной машины вводить в нее в удобной форме большое количество информации;
- 3) теории электрических цепей, поскольку она дает наглядные физические представления для абстрактных уравнений, ставя им в соответствие большое разнообразие электрических, магнитных, диэлектрических и других систем;
- 4) теории расчленения и соединения электрических (или топологических) моделей, потому что с ее помощью можно подразделять процессы как анализа, так и решения на этапы, которые сравнительно легко могут быть выполнены.

Одной из практически наиболее важных особенностей тензорного анализа, применительно к данному случаю, является его способность разделять задачи на части (в дополнение к физическому разделению систем на части).

Тензор импедансов вращающейся электрической машины, например, может быть разделен на части (тензор сопротивления, тензор индуктивности и тензор кручения), отображающие действие электрических, магнитных и механических сил.

Для сложных систем, в которых протекают физические процессы различной природы или назначения, необходимо дополнить физическое разделение функциональным.

Следует подчеркнуть, что метод расчленений не предполагает конкурировать ни с каким из существующих методов решения, не использующих расчленения (итеративные методы и т.д.), ни с разделением матриц. Когда задача может быть решена без расчленения, бессмысленно делить ее на части и решать с помощью метода расчленений, если при этом не обнаруживаются преимущества. Аналогично, если известны, уравнения задачи, которые можно экономно решить с помощью разделения матриц на блоки, то их именно так и следует решать, вместо того чтобы расчленить систему. Расчленение следует применять особенно в тех случаях, когда система постепенно усложняется в ходе расчета, когда необходимо делать частые изменения в размерах или в используемых материалах, или когда делаются существенные дополнения или удаления, или когда система является лишь частью еще большей системы, и решения следует объединить, или когда одни части системы заменяются другими и т.д. Вообще, считается, что число переменных, в рассматриваемой задаче так велико, что неэкономно записывать уравнения всей системы, или что имеющаяся в распоряжении цифровая или аналоговая вычислительная машина не может оперировать одновременно с таким большим числом переменных, а требует поэтапного решения.

Предлагаемый метод исследования также основывается на предпосылке, что после того, как для некоторой системы получено решение методом расчленений, решение для еще более сложной метасистемы может быть получено комбинированием решений, полученных тем же методом. Некоторые из этих

решений, в свою очередь, могут быть в любое время снова соединены в виде пирамиды для решения систем, размеры которых ранее считались необозримыми. Нет необходимости анализ и решение задачи начинать каждый раз снова и снова, если в системе произошло изменение или она соединяется с другими системами. Постепенное усложнение решений вместе с ростом самих систем – основной принцип метода расчленений.

Описываемый метод исследования отличается от обычных чисто математических методов тем, что он начинает исследование с первоначальной физической системы или ее топологической модели, а не с уравнений всей системы, которые даже никогда и не записываются. Необходимо вывести лишь уравнения состояния каждой подсистемы, которые, в свою очередь, могут быть также выведены с помощью метода расчленений. Так что фактически вначале необходимо записать только уравнения "элементарных ячеек" системы. А далее нужно лишь сочетать надлежащим образом более простые понятия анализа по частям с более тонкими понятиями решения методом расчленений.

Идея, лежащая в основе решения сложных задач путем рассмотрения ее меньших изолированных частей, не нова. Отметим хотя бы следующие три метода, использующие эту идею:

- 1) получив собственные решения, разделяют систему на несвязанные части по числу ее степеней свободы;
- 2) симметричные компоненты в трехфазных электрических системах позволяют разделить цепь на три частные цепи;
- 3) теория групп использует определенную симметрию для разделения данной системы на несколько независимых частей.

В то время, как вышеупомянутые способы диагонализации определяются природой рассматриваемых систем, диакоптика выполняет процесс диагонализации принудительным образом. Способ расчленений мало или почти совсем не учитывает свойств самой системы и определяется главным образом внешними соображениями, такими, как размеры имеющейся в распоряжении вычислительной техники, или

требованиями расчета, или постепенным увеличением системы и т.д. Что касается свойств системы, то они-то учитываются лишь как вторичные факторы.

В работах, опубликованных Кроном на протяжении последней четверти века, он использовал аппарат уравнений и графов главным образом для вывода уравнений состояния сложных физических систем по частям. И только в последние годы он обнаружил, что путем небольших изменений этот же самый метод, использующий уравнения и графы, может быть также успешно применен для решения по частям сложных задач.

Применение понятий физического и функционального разделения наряду с понятиями конечных и бесконечно малых элементов побуждает автора расширить область задач, решаемых методом расчленений.

Процесс разделения при решении сложных физических и экономических задач аналогичен децентрализации огромной экономической системы (такой, например, как компания Дженерал Электрик), включающей в себя большое число действующих подразделений. Каждое из этих подразделений функционирует как независимая деловая организация. Тем не менее все они связаны в одно экономическое целое вспомогательными организациями, которые аналогичны цепи пересечений. Вспомогательная организация сама по себе является миниатюрной копией всей компании, и ее финансовое положение зависит от соотношения затрат и прибылей, также как и любых других действующих подразделений. Функциональное разделение каждой подсистемы с использованием тензоров аналогично разделению функций каждого независимого подразделения на закупочные, технические, обрабатывающие, сбывающие и т.д. Вспомогательные организации должны быть также подвергнуты функциональному разделению того же типа. Более детальное описание этой аналогии можно найти в работе [24].

Диакоптический подход к моделированию сложных электротехнических систем предложен школой Демирчана и носит название «макромоделирование» [25]. Метод макромоделирования в применении к моделированию сложных электрических цепей с вентильными элементами развивался в работах Борю [26].

Применительно к исследуемой нами области применения общий метод расчленения системы может быть сформулирован следующим образом:

1. Каждый из агентов, членов сообщества, является замкнутой системой. Эта замкнутая система определяется ролью агента - т. е. набором политик, которые агент выполняет; множеством системных параметров, т.е. тех параметров, которые воспринимаются агентом в составе внешних воздействий; множеством состояний агента.

2. Взаимодействие агентов внутри сообщества определяется при помощи понятий воздействие и реакция на базе уравнений (2.2) и (2.5), описанных в следующей главе.

3. Моделирование реакции сообщества на внешнее воздействие осуществляется при помощи подстановки модели агента выделенного функционального элемента вместо более сложной модели, включающей в себя все агенты сообщества.

4. Моделирование выполнения политики базируется на выделении подграфа взаимодействия необходимых для выполнения данной политики агентов. Топология подмножества таких агентов переменна, поскольку зависит от как политики, так и от состояния взаимодействующих агентов.

5. Модель сообщества агентов (исходя из 2.1. – 2.4.) эквивалентна модели агента - члена сообщества. Таким образом, модель сообщества (подразделения в рассмотренном нами примере) является моделью агента в другом сообществе (вычислительного центра университета).

6. Сообщество агентов является стабильной системой. Это означает, что не существует внешнего воздействия, воспринимаемого сообществом и приводящего это сообщество к разрушению.

Любопытным представляется тот факт, что перечисленные принципы расчленения системы не противоречат известной парадигме объектно-ориентированного программирования. Это косвенным образом подтверждает, что излагаемый подход имеет право на жизнь при моделировании таких сложных эволюционирующих программных систем, какими являются сообщества

интеллектуальных информационных агентов в функциональной проекции модели виртуального информационного

### 1.3.2 Конечные автоматы [27, 28]

Теория автоматов представляет собой раздел теории управляющих систем [29], изучающий математические модели преобразователей дискретной информации, называемые автоматами. С определенной точки зрения такими преобразователями являются как реальные устройства (вычислительные машины, автоматы, живые организмы и т. п.), как и абстрактные системы (математические машины, аксиоматические теории и т. д.). Характерной особенностью этих преобразователей является дискретность функционирования и конечность областей значений параметров, описывающих их. Теория автоматов возникла в середине 20 столетия в связи с изучением свойств конечных автоматов. Со временем предмет ее исследований расширился за счет рассмотрения также различных обобщений конечных автоматов. Содержательно конечный автомат можно охарактеризовать как устройство, имеющее входной и выходной каналы и находящееся в каждый из дискретных моментов времени в одном из конечного числа состояний. По входному каналу в каждый момент в устройство поступают входные сигналы (из некоторого конечного множества сигналов); указывается закон изменения состояний к следующему моменту в зависимости от входного сигнала и состояния устройства в предыдущий момент, а также значение выходного сигнала (из некоторого конечного множества сигналов). Существуют различные подходы к определению понятия конечного автомата, которые могут быть разбиты на группы макроподхода и микроподхода. При макроподходе интересуются внешним поведением устройства, тем, как оно осуществляет переработку входной информации в выходную информацию и в последовательность состояний, отвлекаясь от внутреннего его строения. На этом пути приходят к понятию абстрактного автомата. Тем самым абстрактный конечный автомат может быть задан с помощью набора отображений, описывающих его «внешнее» функционирование. При микроподходе учитывается структура устройства, функционирования и связь между собой его частей. На этом

пути приходят к понятию структурного конечного автомата, называемого также автоматной схемой или логической сетью. Структурный конечный автомат задается конечным множеством абстрактных автоматов, конечной схемой их соединения и указанием влияния частей схемы друг на друга. Понятия конечного абстрактного и конечного структурного автоматов можно считать составляющими понятие автомата.

Абстрактный автомат получается на пути допущения рассмотрения произвольных (не обязательно конечных) множеств входных и выходных сигналов, множества состояний, а также путем расширения понятия зависимости состояния и выходного сигнала и состояния, которые фигурируют в описании конечного абстрактного автомата. Структурный автомат получается на пути допущения рассмотрения произвольных множеств автоматов и схем их соединения.

В соответствии с двумя подходами к понятию автомата вся теория автоматов может быть разделена на теорию абстрактных и теорию структурных автоматов. В первом случае изучаются свойства различного рода поведений автоматов, во втором случае изучаются свойства композиций автоматов относительно заданного класса операций, а также алгебр автоматов.

Теория автоматов имеет широкий круг применений как внутри математики в различных ее областях (в алгебре, математической логике и др.), так и в решении практических задач (в анализе и синтезе ЭВМ, в распознавании образов и т. п.). понятие автомата может служить модельным объектом в самых разнообразных задачах, благодаря чему возможно применение теории автоматов в различных научных и прикладных исследованиях<sup>3</sup>.

Таким образом, алфавитные преобразования, реализуемые дискретными преобразователями информации, выдающими некоторый выходной сигнал (букву выходного алфавита) в ответ на каждый входной сигнал (букву входного алфавита). принято называть **абстрактными автоматами**. Для задания абстрактного автомата

---

<sup>3</sup> *Что собственно мы и сделали, применив теорию автоматов к построению функциональной проекции такого подразделения университета, как его издательский центр. Как будет показано ниже, в данной модели весь интерфейс между клиентом и виртуальным издательским центром построен на концепции конечных автоатов.*

должны быть заданы три множества: **входной алфавит**, **выходной алфавит** и **множество внутренних состояний** автомата. Автомат работает в дискретном времени, последовательные моменты которого удобно отождествлять с последовательными натуральными числами  $t=0,1,2, \dots$  (что всегда можно сделать, выбирая должным образом единицу измерения времени). В каждый данный момент дискретного автоматного времени  $t=0,1, \dots$  автомат находится в некотором определенном состоянии  $a = a(t)$  из множества его внутренних состояний

В каждый момент автоматного времени, начиная с  $t=1$ , на вход автомата поступает в качестве входного сигнала одна из букв входного алфавита  $x=x(t)$ . Конечные упорядоченные последовательности входных сигналов  $x(1)x(2) \dots x(k)$  автомата называются **входными словами** этого автомата. На вход автомата может подаваться любое входное слово из некоторого фиксированного заранее множества допустимых входных слов.

Любое допустимое слово  $p= x(1)x(2)\dots x(k)$ , поданное на вход данного автомата, вызывает появление на выходе автомата выходного слова  $q=y(1)y(2)\dots y(k)$ , представляющего собой некоторую упорядоченную конечную последовательность выходных сигналов автомата (букв его выходного алфавита), имеющего ту же самую длину, что и соответствующее ему и однозначно им определяемое входное слово  $p$ . Получаемое соответствие  $\varphi$  между допустимыми входными словами  $p$  и соответствующими им выходными словами  $q$  называется (алфавитным) **отображением, индуцируемым, рассматриваемым автоматом**.

Указанное отображение  $\varphi$  однозначно определяется заданием двух функций  $\delta$  и  $\lambda$ , называемых соответственно **функцией переходов** и **функцией выходов** рассматриваемого автомата.

Функция переходов определяет состояние автомата в любой момент дискретного автоматного времени по входному сигналу в тот же самый момент и состоянию в предыдущий момент автоматного времени:

$$a(t)=\delta(a(t-1),x(t)). \quad (1.1)$$

Функция выходов определяет зависимость выходного сигнала  $y(t)$  автомата от тех же самых переменных:

$$y(t) = \lambda(a(t-1), x(t)). \quad (1.2)$$

Другим способом задания конечных автоматов, обеспечивающим большую наглядность, является задание автоматов с помощью *направленных графов*. Вершины графа (изображаемые на рисунках в виде кружочков) отождествляются с различными состояниями автомата. Стрелка, соединяющая вершину  $i$  с вершиной  $j$ , означает, что существует входной сигнал  $x$ , переводящий автомат из состояния  $i$  в состояние  $j$ , т. е. удовлетворяющий соотношению

$$j = \delta(i, x).$$

Чтобы отличать, какими именно входными сигналами вызывается данный переход автомата из состояния  $i$  в состояние  $j$ , стрелка, соединяющая соответствующие этим состояниям вершины графа, отмечается символами этих входных сигналов. Выходной сигнал  $y$ , определяемый парой  $(i, x)$ , обычно ставится на графе рядом с входным сигналом  $x$  и для отличия от входных сигналов заключается в скобки.

Рассмотрим пример задания конечного автомата с помощью таблиц переходов и выходов направленного графа. Выберем для этой цели относительно простой автомат с тремя внутренними состояниями 1, 2, 3, двумя входными сигналами  $x, y$  и двумя выходными сигналами  $u, v$ . Предположим, что указанный автомат задается таблицами переходов и выходов

	1	2	3
X	2	3	3
Y	3	2	2

	1	2	3
X	U	U	V

Y | V | U | U

Этим таблицам соответствует направленный граф, изображенный на рис. 1.4

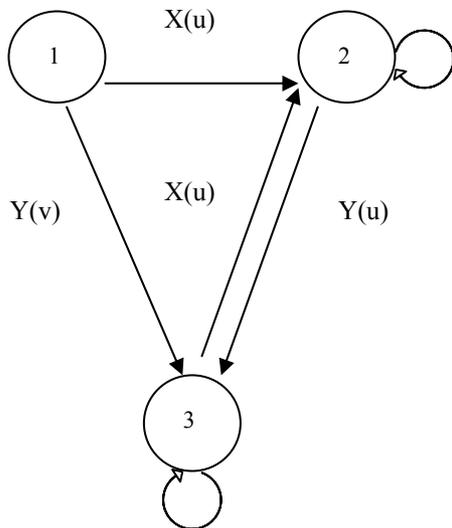


Рис. 1.4. представление автомата в виде графа

Чтобы отличать, какими именно входными сигналами вызывается данный переход автомата из состояния  $i$  в состояние  $j$ , стрелка, соединяющая соответствующие этим состояниям вершины графа, отмечается символами этих входных сигналов. Выходной сигнал  $u$ , определяемый парой  $(i, x)$ , обычно ставится на графе рядом с входным сигналом  $x$  и для отличия от входных сигналов заключается в скобки.

Широко применяются конечные автоматы и при построении функциональной проекции. Данная модель содержит три конечных автомата.

Первый из них

$$F_a = F_a(a, F, s_i) = \{\mathfrak{K}_a, \wp_a(s_i), \mathfrak{R}_a, \delta_a(s_i)\} \quad (1.3)$$

осуществляет авторизацию политики внешнего воздействия  $a$  в определенном состоянии  $s_i \in S$ . Входная последовательность  $a$  записывается в «символах» входного алфавита  $\mathfrak{K}_a$ . Множество состояний конечного автомата  $\wp_a$  есть функция от состояния  $s_i$  агента  $A$  и элементы множества состояний зависят от состояния  $s_i$  агента  $A$ . Функция переходов  $\delta$  также зависит от состояния  $s_i$  агента  $A$ .

Второй конечный автомат можно представить в виде:

$$F_X = F_X(X, X_A, s_i) = \{\mathfrak{K}_X, \wp_X(X_A, s_i), \mathfrak{R}_X, \delta_X(X_A, s_i)\} \quad (1.4)$$

он осуществляющий проверку соответствия параметров  $X$  полученного воздействия  $a$  на соответствие множеству системных параметров  $X_A$  и ограничениям состояния  $s_i$ .

И, наконец, последний автомат

$$F_Y = F_Y(Y, f, s_i) = \{\mathfrak{K}_Y(f), \wp_Y(f, s_i), \mathfrak{R}_Y(f), \delta_Y(f, s_i)\} \quad (1.5)$$

осуществляет проверку соответствия описания ожидаемых результатов  $Y$  тем результатам, которые могут быть получены при выполнении агентом  $A$  в состоянии  $s_i$  политики  $f$ .

#### **1.4 Постановка задачи**

Теперь, исходя из всего выше сказанного, можно сделать вывод, что основными **задачами диплома** являются:

- формализовать функциональный элемент сообщества;
- определить правила взаимодействия функциональных элементов (агентов);
- описать всевозможные множества состояний и эволюцию элементов модели;
- разработать формальное представление сообщества интеллектуальных агентов моделирующего функциональное подразделение предприятия, используя уже построенную модель отдельно взятого агента, т.е. модель агента распространим на сообщество.

## **2 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ВИРТУАЛЬНОГО ПРЕДПРИЯТИЯ НА ПРИМЕРЕ ИЗДАТЕЛЬСКОГО ЦЕНТРА С АКЦЕНТОМ НА ФУНКЦИИ "ЗАКАЗ НА ИЗГОТОВЛЕНИЕ ПОЛИГРАФИЧЕСКОЙ ПРОДУКЦИИ"**

Таким образом, сегодня, как никогда ранее актуальна проблема моделирования виртуальных предприятий в различных разрезах и с поддержкой новейших технологий сегодняшнего дня. Интерес к ним увеличивается с каждым днем все больше и больше. Отчасти в этом "виновна" всемирная сеть Интернет, которая стремительно ворвалась в жизнь миллионов людей и навряд ли собирается уступать уже завоеванные рубежи кому-либо или чему-либо. Плюс к этому четкая и стремительная тенденция к распределению и виртуализации производственных и бизнес процессов.

Как уже упоминалось выше, среди современной литературы, охватывающей своим содержанием проблемы моделирования виртуальных предприятий, существует несколько идей построения такого рода сред. Одним из направлений является использование логики одновременно выполняющихся транзакций (Concurrent Transaction Logic). Данный подход представляет собой основные принципы Системы Управления Виртуальными Предприятиями для моделирования (Virtual Enterprise Management Systems (VEMS)) и формальных выводов взаимодействий в модели виртуального предприятия [4]. Этот способ моделирования ориентирован на создание формального аппарата описания и динамическую систему (виртуальное предприятие), которая обязана решать поставленные перед этим предприятием четко определенные задачи. Существуют и другие не менее интересные методы функционального моделирования реальных предприятий, построенные на базе аппарата (framework), базирующегося на понятиях роли (role), политики (policy) и взаимодействия (relationship). Одним из важных примеров реализации такого подхода является ICRF (Imperial College Role Framework) [5]. Интенсивно развивается и направление, связанное с построением формализмов для информационных потоков моделирующих взаимодействия и

взаимоотношения в сообществах функциональных элементов виртуальных предприятий на базе парадигмы кооперирующихся агентов (cooperative agents). Примером построения такого формализма может служить [6].

Как уже упоминалось выше, в рассматриваемом нами контексте парадигма виртуального информационного пространства является способом моделирования реального или виртуального предприятия при помощи представления этой сложной динамической системы в терминах интегрированной информационной системы [2], архитектура которой базируется на стандарте CORBA [3].

В этом дипломе мы сделаем попытку смоделировать реальный процесс и функцию, применяя разработанные нами архитектуру и подход к построению Унифицированного Информационного Пространства (УИП) [2]. Это позволит нам в дополнение к функциональным моделям учитывать организационную и геометрическую топологию моделируемого предприятия и таким образом интегрировать уже работающие на предприятии Информационные Системы (ИС) (Legacy Information Systems). Для построения адаптивных эволюционирующих моделей функциональных блоков этого виртуального информационного пространства мы применяем диакоптический подход [15], концепцию взаимодействующих (кооперирующихся, cooperative) интеллектуальных агентов [30], методы семантического анализа данных [31] и элементы теории конечных автоматов [27].

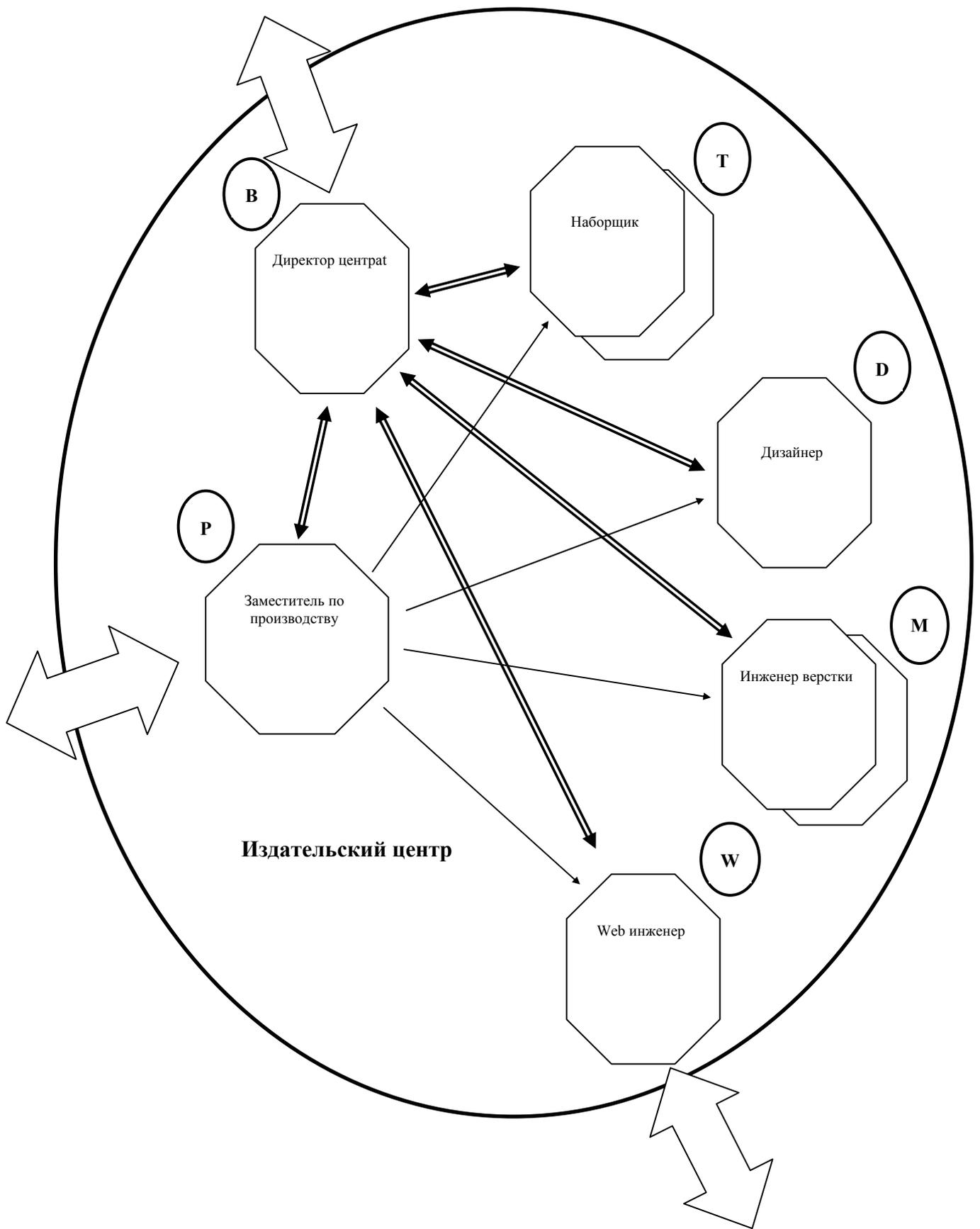
## **2.1 Структура и функции издательского центра**

В дипломе для демонстрации применяемых подходов к моделированию функциональной проекции виртуального предприятия рассмотрим типичный для анализа виртуального информационного пространства пример подразделения, функциональное описание которого даст нам отправную точку для разработки более общих принципов моделирования. Интересным с функциональной точки зрения является анализ Издательского Центра. Специфика этого университетского подразделения удачно сочетает выполнение как хорошо формализованных

технологических функций, так и участие в исследовательских проектах, что предполагает использование процедур анализа, оптимизации, а также отсутствие детального планирования и четко определенных статических связей. Кроме того, специфика подразделения такова, что в нем сосуществуют как реальные штатные сотрудники, имеющие статические функциональные обязанности, привязанные к технологическим цепочкам, так и виртуальные компоненты - группы, участвующие в выполнении того или иного исследовательского проекта.

В рамках диплома будем рассматривать только реальные (долгоживущие) функциональные элементы, имея, однако в виду, что эти объекты эволюционируют в рамках своего функционального поля. Моделирование виртуальных функциональных компонент будет реализовано в дальнейшем на базе полученных в данной работе подходов. Реальная составляющая Издательского центра (рис.2.1) включает в себя следующие функциональные элементы:

- директор центра,
- зам. по производству,
- наборщик, дизайнер,
- инженер верстки,
- Web - инженер.



**Рис. 2.1. Функциональные элементы и их взаимодействие в Издательском Центре**

Стрелками на рисунке на рис.2.1 частично показаны различные типы связей между функциональными элементами подразделения: равноправные, связи подчинения.

В общем случае, граф функциональных связей между функциональными элементами является NP - полным. Нам, однако, будут интересовать взаимодействия не в контексте подразделения, а в контексте выполнения подразделением некоторой функции. Важно также отметить, что выполнение подразделением той или иной функции вызывается воздействием на подразделение извне. Такие воздействия показаны на рис.2.1 широкими двунаправленными стрелками. Для восприятия внешних воздействий мы будем выделять некоторые функциональные элементы. В схеме рассматриваемого подразделения такими элементами являются директор центра, заместитель по производству и Web - инженер. Далее мы покажем, что топология взаимодействия функциональных элементов подразделения будет изменяться как в связи с выбором выполняемой функции (что очевидно), так и в процессе выполнения выбранной функции.

Перечислим **основные функции**, выполняемые **Издательским Центром**:

- ☐ принять и обработать у клиента заказ на выпуск полиграфической продукции;
- ☐ предоставить клиенту электронную публикацию;
- ☐ дать справку о состоянии процесса принятия статьи к публикации в журнале;
- ☐ принять заказ на размещение рекламы;
- ☐ показать образцы продукции с указанием стоимости, сроков исполнения, полиграфических параметров;
- ☐ подготовить комплект документации на выполнение тиража;
- ☐ обработать учетные данные по выполненной работе.

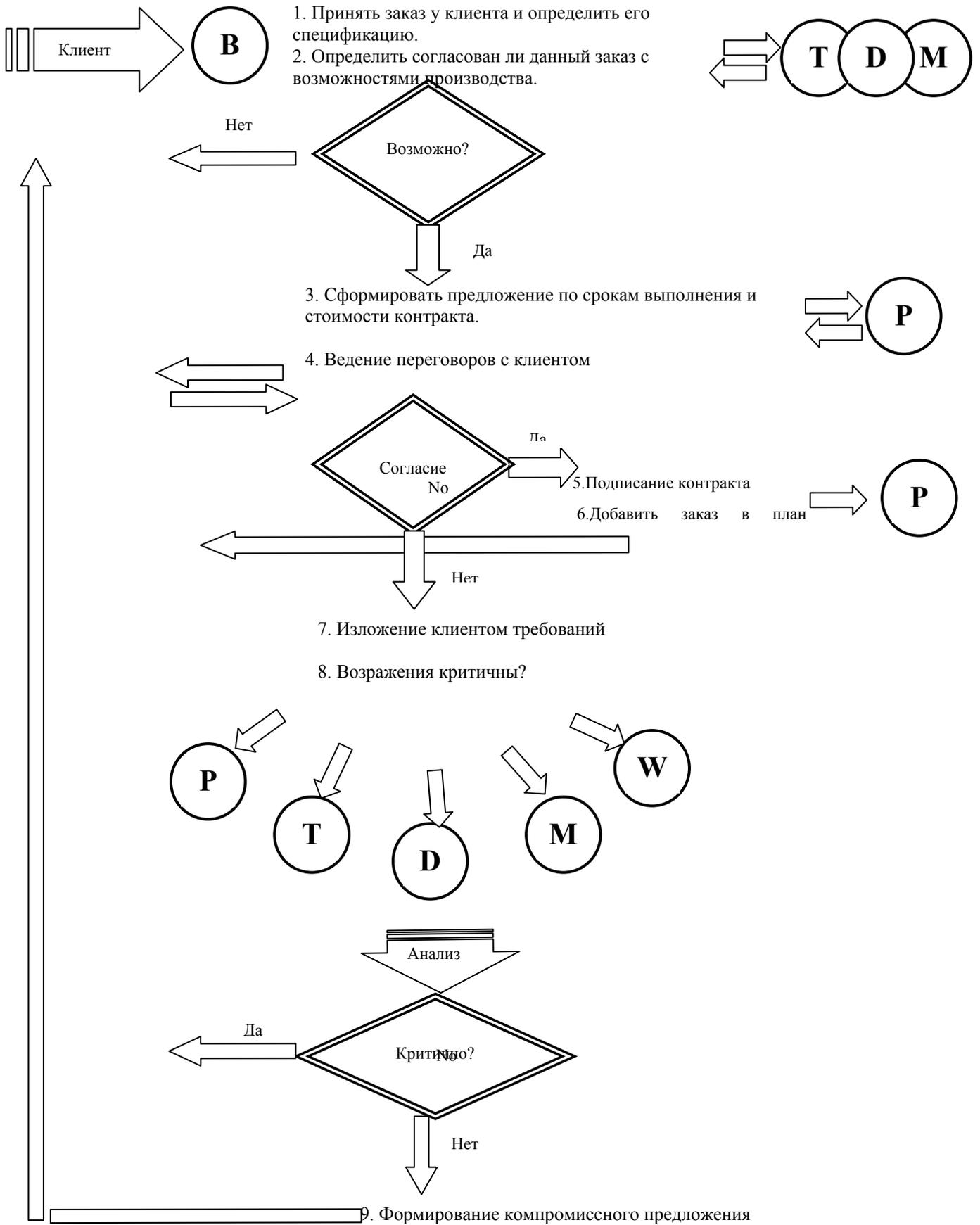
Среди этого разнообразия выполняемых полиграфическим центром функций для разработки функциональной модели исследуемой структурной единицы университета выберем лишь одну, причем, наиболее интересную с функциональной точки зрения.

Возьмем, к примеру, **функцию принятия заказа на изготовление полиграфической продукции** и проанализируем тщательным образом полный

процесс реализации этой функции от момента поступления заказа от клиента до момента подписания договора о принятии заказа между клиентом и руководством полиграфического центра.

## **2.2 *Схема функции принятия заказа на изготовление полиграфической продукции***

Выбранная нами процедура выполнения функции принятия заказа на печать полиграфической продукции интересна тем, что является процессом с итерационным уточнением и включает в себя разнообразные виды взаимодействий между функциональными объектами подразделения (директива, детерминированный, недетерминированный запросы). Схема этой процедуры приведена на рисунке 2.2. Внешним воздействием, возбуждающим выполнение функции принятия заказа является поступление заявки и спецификации заказа от клиента. Функциональным объектом, назначенным для восприятия такого воздействия на подразделение, является директор центра.



**Рис. 2.2. Схема процедуры принятия заказа**

Процедура, выполняемая директором центра, является достаточно стандартной последовательностью действий по согласованию различных параметров контракта. Нас, однако, будет интересовать не сама последовательность действий, а различные виды взаимодействий между функциональными элементами подразделения.

Итак, в самом начале клиент приходит к непосредственному руководителю издательства (будем называть его боссом или директором центра) и излагает суть своей просьбы с предоставлением необходимой информации и дачей множества требований, которые должны быть удовлетворены издательством. Эти требования могут включать в себя срок выполнения заказа, необходимые полиграфические атрибуты, стоимость и т.д. Босс со своей стороны в первую очередь **определяет спецификацию данного заказа** и согласует совместно с наборщиками (**T**), дизайнером (**D**) и инженерами верстки (**M**) **соответствует ли данный заказ возможностям производства**. Все это отображено на схеме рис. 1 в виде шага 1.

Шаг 2. рассматриваемой процедуры предполагает распределение воздействия **«проанализировать заявку на соответствие возможностям производства»** одновременно на несколько функциональных элементов: **T** (наборщик), **D** (дизайнер), **M** (инженер верстки), два из которых представляют собой группы (**T** и **M**), с последующим объединением и оптимизационным анализом их ответов для получения наиболее пессимистичной оценки.

Шаг 3. процедуры предполагает воздействие **«сформировать предложение по срокам выполнения и стоимости контракта»** на элемент **P** (заместитель по производству) и получение его реакции в виде такого **«предложения»**.

Шаг 4 предусматривает **ведение переговоров с клиентом**, причем в случае согласия клиента предложенным вариантом контракта приходим к шагу 5 и шагу 6.

Шаг 5 – **подписание контракта**.

Шаг 6. предполагает воздействие **«добавить заказ в план производства»** на элемент **P** без ожидания его реакции.

Шаг 7 – **изложение клиентом требований**, т.е. того, с чем он не согласен.

Шаг **8.** одновременно распространяет воздействие «**Возражения критичны?**» на несколько элементов (**P, T, D, M, W**) с целью последующей селекции из набора реакций наиболее оптимистичного результата.

Проанализируем одно из воздействий более подробно. Воздействие на шаге **8.** представляет собой попытку руководителя издательского центра выяснить на каком участке подразделения поступившие от клиента ограничения наиболее критичны. Целью элемента **B** в этом случае является - определить возможность формирования конструктивной реакции подразделения в целом на внешнее воздействие, полученное от клиента. Предположим, ограничения, наложенные клиентом на шаге **7.**, сформулированы следующим образом:

- срок выполнения заказа не должен превышать двух недель, начиная с завтрашнего дня;
- оптическое разрешение при печати не должно быть менее 600 dpi;
- стоимость одного экземпляра не должна превышать \$1.25

Сценарий выполнения шага **8.** может выглядеть следующим образом:

Элемент **B:**

**1. Сформировать воздействие** (Оценить возможность выполнения заказа по шкале 0 - невозможно; 0.25 - возможно, но за счет приостановки выполнения других работ; 0.5 - возможно, но необходимо оплатить сверхурочную работу, приобрести материалы; 0.75 - возможно, но потребует уточнения и несущественной коррекции плана работы; 1 - безусловно возможно. Параметры заказа: спецификация заказа в файле SPEXXX.XLS, ограничения заказчика (срок выполнения заказа не должен превышать двух недель, начиная с завтрашнего дня, оптическое разрешение при печати не должно быть менее 600 dpi, стоимость одного экземпляра не должна превышать \$1.25)

**2. Воздействовать** на элементы **P, T, D, M, W;** ждать результатов.

Элементы **P, T, D, M, W:**

- 1.** Получить воздействие.
- 2.** Определить, входит ли реакция на полученное воздействие в сферу собственных полномочий.

3. Определить параметры воздействия, которые относятся к сфере собственных полномочий (для элемента **P** такими параметрами будут: спецификация заказа в файле SPEXXX.XLS, ограничения заказчика - срок выполнения заказа не должен превышать двух недель, начиная с завтрашнего дня, стоимость одного экземпляра не должна превышать \$1.25; для элемента **T**: спецификация заказа в файле SPEXXX.XLS, ограничения заказчика - срок выполнения заказа не должен превышать двух недель, начиная с завтрашнего дня; и т. д.).

4. Определить, какие результаты из запрошенных этот элемент может сформировать (Для данного воздействия запрашиваемый результат - это число, принимающее значение из множества  $\{0, 0.25, 0.5, 0.75, 1\}$ , указав, какой/какие из параметров влияют на сформированное значение. Элемент, не имеющий отношения к выполнению оцениваемого заказа - таким элементом в данном случае может оказаться **W** - должен, естественно, дать ответ 1 - безусловно возможно). Возможные примеры ответов:

- а) элемент **P** - 0.25 (срок выполнения, стоимость);
- б) элемент **M** - 0.5 (срок выполнения, спецификация заказа).

5. Сформировать результат и передать его элементу **B**.

Элемент **B**:

1. **Проанализировать полученные результаты:** определить интегральную пессимистическую оценку и критические параметры, оказавшие влияние на формирование этой оценки. Для приведенных примеров ответов элементов **P** и **M** интегральной оценкой очевидно будет 0,25 (срок выполнения, стоимость). Полученный результат, таким образом, вероятнее всего приведет либо к дальнейшему итерационному уточнению с клиентом критических параметров, либо к отказу от подписания контракта.

### **2.3 *Агенты, их роли и политики в рассматриваемой модели процедуры принятия заказа***

Модель рассмотренной процедуры может быть построена с использованием известных подходов (например [4,5,6,30]), базирующихся на понятиях интеллектуального агента, роли, политики и взаимодействия (кооперации).

Каждый из агентов в рассмотренном примере фактически выполняет роль персонального ассистента того функционального элемента (мастера), реакции которого он представляет. Так агент **В** моделирует в данной процедуре поведение директора издательского центра, агенты **Р**, **Т**, **Д**, **М**, **W** - поведение его подчиненных:

- Заместитель по производству (**Р**)
- Наборщик (**Т**)
- Дизайнер (**Д**)
- Инженер верстки (**М**)
- Web-инженер (**W**)

Анализ примера показывает, что реакция агента на детерминированное воздействие может зависеть от состояния агента. Действительно, ответы агентов **Р**, **Т**, **Д**, **М**, **W** в пункте 8. зависит от их загрузки другой работой (по другим контрактам), наличия или отсутствия необходимых материалов, программного обеспечения, технологических ограничений. Очевидно, реакция агента на воздействие может привести к переходу его в другое состояние. Состояние агента, в свою очередь может накладывать определенные ограничения на воздействия или параметры воздействий, которые принимаются агентами к исполнению.

Таким образом, при разработке модели взаимодействующих агентов необходимо учитывать, что роль агента - есть функция состояния, изменяющая состояние. Следует также отметить, что состояние агента может накладывать ограничения как на область значений входящих параметров, так и на получаемые на базе этих параметров результаты.

## 2.4 Типы воздействий и реакций

В рассмотренном примере мы выделили несколько видов воздействий, оказываемых на агенты сообщества, которые отличаются друг от друга реакцией на это воздействие. Если представить **внешнее воздействие** на выделенный функциональный элемент (агент **B**) и его реакцию, в виде следующего уравнения:

$$\tilde{Y} = a(f, X, Y) \quad (2.1)$$

где:

$a(f, X, Y)$  - воздействие,  $f$  - политика,

$X = (x_1, \dots, x_n)$  - параметры,

$Y = (y_1(X), \dots, y_m(X))$  - описание запрашиваемых результатов,

$\tilde{Y} = (\tilde{y}_1(X), \dots, \tilde{y}_k(X))$  - реакция (результаты)

и взаимодействие агентов внутри сообщества, в виде:

$$\tilde{Y}^* = a^*(f^*, X^*, Y^*) \quad (2.2)$$

то:

- для воздействия вида «директива» (шаг 6. нашего примера) будет справедливо:

$$X^* \subseteq X, \quad Y^* = \tilde{Y}^* = \emptyset, \quad f^* \subseteq f; \quad (2.3)$$

- для воздействия вида «детерминированный запрос с детерминированной реакцией» (шаг 3. нашего примера) будет справедливо:

$$X^* \subseteq X, \quad Y^* = \tilde{Y}^*, \quad f^* \subseteq f; \quad (2.4)$$

- для воздействия вида «детерминированный запрос с оптимизацией результата» (шаг **2.** нашего примера) будет справедливо:

$$, X^* \subseteq X \quad Y^* \subseteq \tilde{Y}^* , \quad f^* \subseteq f ; \quad (2.5)$$

- для воздействия вида «недетерминированный запрос с оптимизацией результата» (шаг **8.** нашего примера) будет справедливо:

$$X^* = X , \quad Y^* = \tilde{Y}^* , \quad f^* \equiv f ; \quad (2.6)$$

Очевидно, что (2.5) дает нам соотношения для общего случая связи компонентов уравнений (2.1) и (2.2).

### 3 ФОРМАЛЬНАЯ МОДЕЛЬ СООБЩЕСТВА ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ

Попробуем на базе выработанного подхода разработать формальное представление сообщества интеллектуальных агентов моделирующего функциональное подразделение предприятия. Будем придерживаться принципов, предложенных в подразделе 2.5 и, поэтому, начнем разработку модели сообщества с формализации его функционального элемента (агента). Затем, определим правила взаимодействия этих элементов, опишем множества состояний и эволюцию элементов модели и в конце вернемся к модели агента, распространив ее на сообщество в целом.

#### 3.1 Модель агента как основного структурного элемента сообщества интеллектуальных агентов

Основываясь на анализе примера, который мы выбрали для моделирования функциональной модели, можно видеть, что каждый функциональный элемент, действует по следующему алгоритму:

```
DO While .T.  
    A = Wait(Action)           Среагировать на входное воздействие  
     $\tilde{Y}$  = Process(ActionEvent)  Обработать полученное воздействие  
    IF (Requested)  
        SendResults(A,  $\tilde{Y}$ )      Возвратить реакцию на воздействие  
    END IF  
LOOP
```

где:

- Wait() - процедура ожидания события воздействия.

- Process() - процедура обработки полученного воздействия, которая выполняет следующую последовательность действий:

Проверить свои полномочия (есть ли право реагировать на полученное воздействие  $a(f, X, Y)$  в текущем состоянии  $s_i$ ).

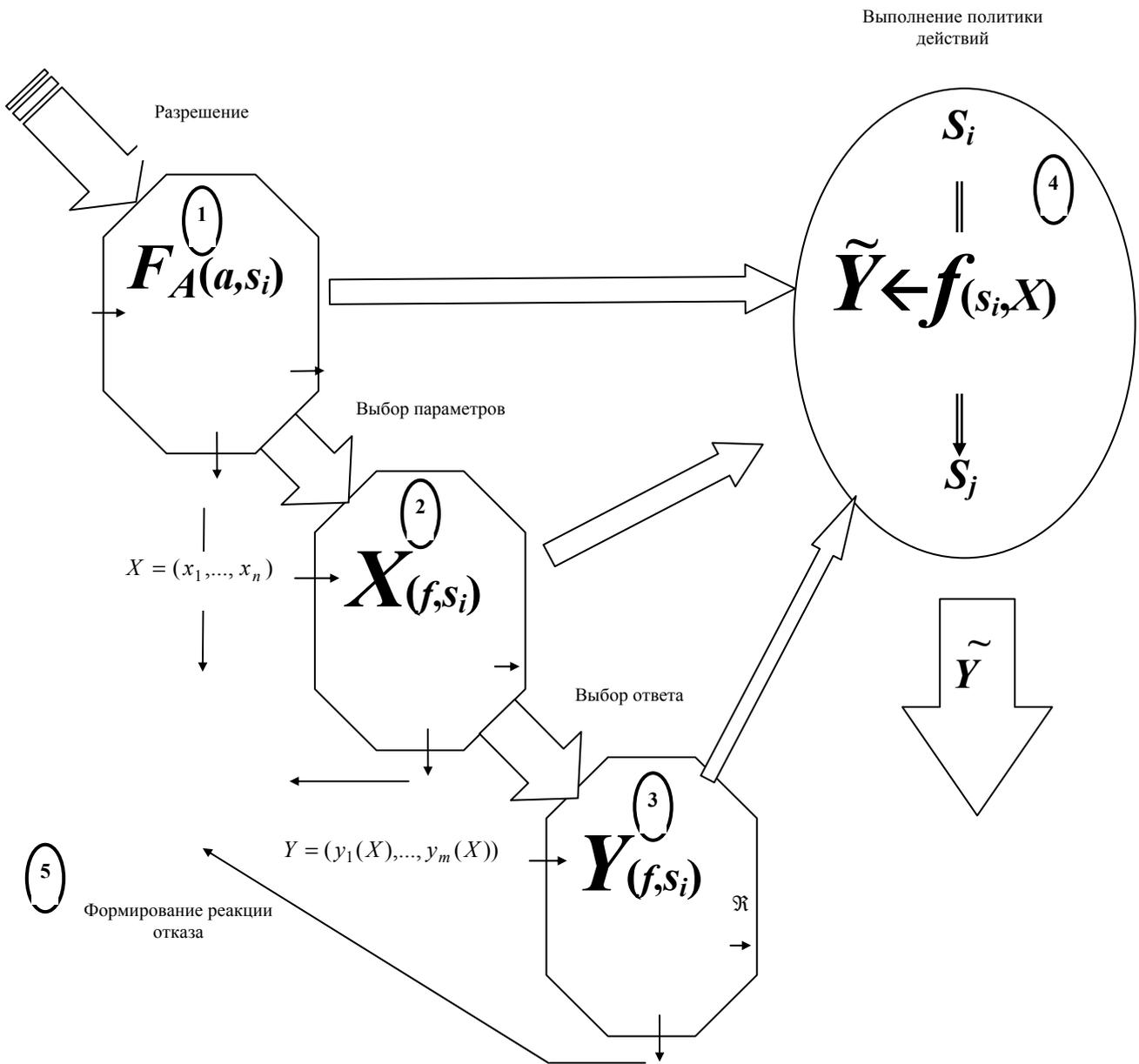
Проверить соответствие полученных параметров  $X = (x_1, \dots, x_n)$  политике  $f$  и граничным условиям состояния  $s_i$ .

Проверить формальное соответствие вектора результата  $Y = (y_1(X), \dots, y_m(X))$  политике  $f$  и состоянию  $s_i$ .

Выполнить политику  $f$ , сформировать вектор результатов  $\tilde{Y} = (\tilde{y}_1(X), \dots, \tilde{y}_k(X))$ , изменить состояние  $s_i \rightarrow s_j$ .

- SendResults() - процедура возврата реакции на обработанное воздействие.

Структурная схема интеллектуального агента, моделирующего поведение функционального элемента сообщества, приведена на Рис. 3.1. Каждый из блоков **1** - **3** представляет собой конечный автомат [27], на вход которого подаются:



**Рис. 3.1. Структурная схема агента, обрабатывающего воздействие.**

- для блока **1** - цепочка «символов»\*, описывающая внешнее воздействие;
- для блока **2** - цепочка «символов», описывающая входные параметры внешнего воздействия;

\* - под «символом» мы будем понимать элемент входного алфавита  $\aleph$  соответствующего конечного автомата

- для блока **3** - цепочка «символов», описывающая ожидаемые результаты, которые объект, оказывающий воздействие на наш агент, рассчитывает получить в результате выполнения запрошенной политики  $f$ .

В зависимости от того, в какое из состояний переходит каждый из конечных автоматов в результате разбора входной цепочки, агентом принимается та или иная стратегия поведения. Так при переходе одним из автоматов 1-3 состояние, отвергающее входную цепочку, формируется результат, говорящий о том, что данный агент  $A$  в данном состоянии  $s_i$  не может выполнить политику  $f$  по одной из следующих причин:

- агент  $A$  в данном состоянии  $s_i$  не авторизован для выполнения политики  $f$  - отвергающее состояние блока **1**;

- входные параметры  $X = (x_1, \dots, x_n)$  не соответствуют системным параметрам агента  $A$ , либо не соответствуют ограничениям на системные параметры агента  $A$ , накладываемым состоянием  $s_i$  - отвергающее состояние блока **2**;

- описания ожидаемых результатов  $Y = (y_1(X), \dots, y_m(X))$  не соответствуют тем результатам, которые может получить агент  $A$  выполняя политику  $f$  (например,  $y_1(X)$  содержит имя файла, а  $\tilde{y}_1(X)$  в результате выполнения политики  $f$  получит значение, представляющее собой вектор  $(0, 0.25, 0.5, 0.75, 1)$ ) - отвергающее состояние блока **3**.

Блок **4** представляет собой исполнитель, в функцию которого входит выполнение политики  $f$ , запрошенной внешним воздействием  $a$ . Из схемы, приведенной на рис.3.1, следует, что этот блок будет выполняться только тогда, когда каждый из конечных автоматов 1- 3 в результате анализа своих входных цепочек перейдет в разрешающее состояние. Задачей блока 4 является выполнение политики  $f$ , формирование результата  $\tilde{Y} = (\tilde{y}_1(X), \dots, \tilde{y}_k(X))$  и перевод агента  $A$ , в случае необходимости, из состояния  $s_i$  в состояние  $s_j$ .

Блок **5** является той частью модели, которая формирует отвергающую реакцию агента  $A$  на воздействие  $a$ . Хорошим примером такой отвергающей

реакции является результат (1 - безусловно возможно), возвращаемый элементом **W** в ответ на воздействие «Оценить возможность выполнения заказа» (см. подраздел 2.2.). Этот пример подчеркивает, что отвергающая реакция агента зависит от политики  $f$ , запрошенной воздействием  $a$ , состояния  $s_i$ , параметров  $X$ ,  $Y$  и далеко не всегда является **негативной**.

Исходя из вышеизложенного, формальная модель агента может быть представлена следующим образом:

$$A = \{F_a, F_X, F_Y, S, X_A, F, F_O\}, \quad (3.1)$$

где:

$S_A = \{s_1, \dots, s_n\}$  - множество состояний агента  $A$ .

$X_A = \{x_1, \dots, x_p\}$  - множество системных параметров агента  $A$ .

$$F_a = F_a(a, F, s_i) = \{\mathfrak{K}_a, \wp_a(s_i), \mathfrak{R}_a, \delta_a(s_i)\}$$

- конечный автомат, осуществляющий авторизацию политики внешнего воздействия  $a$  в состоянии  $s_i \in S$ . Входная последовательность  $a$  записывается в «символах» входного алфавита  $\mathfrak{K}_a$ . Множество состояний конечного автомата  $\wp_a$  есть функция от состояния  $s_i$  агента  $A$  и элементы множества состояний зависят от состояния  $s_i$  агента  $A$ . Множество разрешающих состояний  $\mathfrak{R}_a$  состоит из одного элемента - состояния помеченного на Рис 3.1. символом  $\mathfrak{K}$ . Функция переходов  $\delta$  также зависит от состояния  $s_i$  агента  $A$ .

$$F_X = F_X(X, X_A, s_i) = \{\mathfrak{K}_X, \wp_X(X_A, s_i), \mathfrak{R}_X, \delta_X(X_A, s_i)\}$$

- конечный автомат, осуществляющий проверку соответствия параметров  $X$  полученного воздействия  $a$  на соответствие множеству системных параметров  $X_A$  и ограничениям состояния  $s_i$ .

$$F_Y = F_Y(Y, f, s_i) = \{\mathfrak{N}_Y(f), \mathfrak{P}_Y(f, s_i), \mathfrak{R}_Y(f), \mathfrak{D}_Y(f, s_i)\}$$

- конечный автомат, осуществляющий проверку соответствия описания ожидаемых результатов  $Y$  тем результатам, которые могут быть получены при выполнении агентом  $A$  в состоянии  $s_i$  политики  $f$ .

$F = \{f_1, \dots, f_i, \dots, f_m\}$  - роль агента - множество политик, выполняемых агентом  $A$ .

$F_o$  - программная компонента, выполняющая политику  $f$ .

$f \in F$  - политика действия (Obligation policy) - набор атомарных инструкций, представленный в виде программы, содержащей ветвления. Подобные программы описаны, например в [6].

Структура политики действия по нашему представлению выглядит следующим образом:

DO CASE

CASE State =  $s_1$

**StartThread**(<Id<sub>1</sub>>)

BEGIN

**Execute** <Instruction> on  $(\tilde{x}_i)$  results to  $(\tilde{y}_i)$

**Broadcast** <Action> with  $(\tilde{x}_i, \tilde{y}_i)$  and wait for  $(\tilde{y}_i)$  from each

party

**OptimiseResults**()

**Require**  $A_j$  with (< Action >,  $\tilde{x}_i, \tilde{y}_i$ ) and wait for  $(\tilde{y}_i)$

**Invoke**  $A_j$  with (< Action >,  $\tilde{x}_i, \tilde{y}_i$ )

**ChangeState**( $s_1$ )

END

**CommitThread**(<Id<sub>1</sub>>)

.....

**StartThread**(<Id<sub>k</sub>>)

BEGIN

.....  
END  
**CommitThread**(<Id<sub>k</sub>>)

.....  
CASE State = s<sub>n</sub>

.....  
OTHERWISE

.....

END CASE

Где:

<Id<sub>i</sub>> - идентификатор потока

<Instruction> - инструкция, выполняющая атомарное действие в рамках данной политики без привлечения других агентов

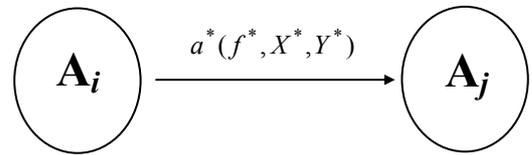
<Action> - идентификатор воздействия, которое определяет часть политики действия, которая не может быть выполнена данным агентом. Для воздействия <Action> может быть предпринят один из следующих методов исполнения:

**Broadcast** <Action> для случая, когда авторизованный для <Action> агент не известен. В этом случае воздействие <Action> производится на всех членов сообщества.

**Require** A<sub>j</sub> with (<Action>,  $\tilde{X}_i, \tilde{Y}_i$ ) для случая, когда авторизованный для <Action> агент известен. Этот способ моделирует случай синхронного взаимодействия с агентом A<sub>j</sub>. Результаты, возвращенные агентом A<sub>j</sub>, составят часть результатов выполняемой политики.

**Invoke** A<sub>j</sub> with (<Action>,  $\tilde{X}_i, \tilde{Y}_i$ ) для случая, когда авторизованный для <Action> агент известен и нам не важна его реакция на воздействие <Action>. В этом случае вектор возвращаемых значений вероятнее всего будет пустым. Интерфейс примитива **Invoke** тем не менее содержит параметр  $\tilde{Y}_i$  на тот случай, если в будущем будет реализован механизм направления результатов третьей стороне A<sub>m</sub>.

$$X = \bigcup_i \tilde{X}_i, \quad Y = \bigcup_i \tilde{Y}_i$$

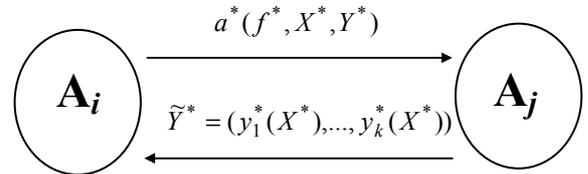


а) Взаимодействие путем выдачи директивы

### 3.2 Модель взаимодействия агентов

Взаимодействие агентов в

рассматриваемой модели, как уже было показано ранее в п. 2.4 и п. 3.1, включает в себя передачу агентом  $A_i$  воздействия  $a^*(f^*, X^*, Y^*)$  агенту  $A_j$  и, в случае



в) Взаимодействие вида воздействие - реакция.

Рис.3.2 Виды взаимодействий агентов в сообществе

необходимости, ожидание агентом  $A_i$  реакции (результатов воздействия)  $\tilde{Y}$  от агента  $A_j$ . Как мы уже отмечали, случай воздействия без ожидания результата представляет собой модель передачи директивы. Для агента  $A_i$ , издавшего директиву, важно только то, что директива получена исполнителем (агентом  $A_j$ ) и этого достаточно, чтобы агент  $A_i$ , безусловно продолжил выполнение своей политики. Случай взаимодействия путем выдачи директивы представлен на Рис. 3.2а, а соотношениями для параметров такого воздействия являются (2.3).

Взаимодействие вида воздействие - реакция, в отличие от случая передачи директивы, предполагает синхронизацию выполнения агентами  $A_i$  и  $A_j$  своих политик (см. Рис. 3.2в). Такой вид взаимодействия агентов, как легко видеть, покрывает рассмотренные нами ранее в п. 2.4. случаи, описываемые соотношениями (2.4)-(2.6).

Таким образом, модель взаимодействия агентов внутри сообщества может быть описана уравнением (2.2) с учетом соотношений (2.3) - (2.6), характеризующих специфику различных видов взаимодействия.

Для завершения формального описания модели взаимодействия интеллектуальных агентов в сообществе виртуального информационного пространства нам необходимо сформулировать способ анализа возвращаемых результатов. Необходимость в таком анализе, как показано в п. 2.2., возникает,

например, в том случае, когда агент  $A_i$ , инициирующий воздействие  $a^*$ , не знает точно, какой из агентов  $A_j$  в состоянии выполнить политику  $f^*$  наиболее полно и наиболее конструктивным образом.

Предположим, что агент  $A_i$  инициировал воздействие  $a^*(f^*, X^*, Y^*)$  одновременно на несколько агентов  $A_j^1, \dots, A_j^q$ . Пусть также, каждый из агентов  $A_j^1, \dots, A_j^q$  вернул агенту  $A_i$  следующие результаты:

		$Y^* =$	$y_1^*$	$y_2^*$	$y_3^*$	.....	$y_l^*$	.....	$y_m^*$
$A_j^1 \rightarrow$	<b><math>\Phi =</math></b>	$\tilde{Y}_1 =$	--	$\tilde{y}_2^1(X^*)$	$\tilde{y}_3^1(X^*)$	.....	--	.....	$\tilde{y}_m^1(X^*)$
$A_j^2 \rightarrow$		$\tilde{Y}_2 =$	$\tilde{y}_1^2(X^*)$	--	$\tilde{y}_3^2(X^*)$	.....	$\tilde{y}_l^2(X^*)$		--
$A_j^3 \rightarrow$		$\tilde{Y}_3 =$	$\tilde{y}_1^3(X^*)$	$\tilde{y}_2^3(X^*)$	--	.....			
....									
$A_j^i \rightarrow$		$\tilde{Y}_i =$	--	--	$\tilde{y}_2^i(X^*)$		--		$\tilde{y}_m^i(X^*)$
...									
$A_j^q \rightarrow$		$\tilde{Y}_q =$	$\tilde{y}_1^q(X^*)$				$\tilde{y}_l^q(X^*)$		$\tilde{y}_m^q(X^*)$

Пусть, наконец, для каждого столбца полученной таким образом матрицы  $\Phi$  известно отображение  $\lambda_l$ , переводящее элементы этого столбца в числовое значение из промежутка  $[0,1]$ .

Тогда задача анализа результатов и выбора агента  $A_j^i$ , наиболее подходящего для выполнения политики  $f^*$ , может, например, сводиться к задаче поиска строки матрицы  $\Phi$ , для которой сумма  $\lambda_l(\tilde{y}_l^i(X^*))$  максимальна:

$$A_j < - - A_j^i : \sum_{l=1,m} \tilde{\lambda}_l(\tilde{y}_l^i(X^*)) = \max \quad (3.1)$$

Для построения отображений  $\lambda_l$  для результатов нечисловой природы рационально воспользоваться методом кластеризации (clustering method), предложенным в [22].

### 3.3 Состояния и эволюция агентов

Эволюция интеллектуальных агентов в сообществе, исходя из того, что предложенная нами модель дискретна, - есть процесс переходов каждого из агентов  $A$  из одного состояния  $s_i$  в другое состояние  $s_j$ . Очень важно отметить, и мы уже прикасались к этому ранее в п. 2.3, что от состояния агента  $A$  зависит то, как он выполняет политику  $f$ , а также ограничения на входные параметры воздействия  $X^*$ . Таким образом, эволюция агента - есть эволюция его роли. Эти соображения приводят нас к следующей формулировке множества состояний интеллектуального агента  $A$ :

Множество состояний агента  $A$ :  $S_A = \{s_1, \dots, s_n\}$  - есть множество элементов  $s_i, i = 1, \dots, n$ , каждый из которых представляет собой кортеж арности 3 следующего вида:

$$s_i = \{r(X_A), q(F_a), t(F)\} \quad (3.2)$$

где:

$r(X_A)$  - функция ограничений на системные параметры агента  $A$ , накладываемых в состоянии  $s_i$ ;

$q(F_a)$  - функция ограничений на авторизацию политик, выполняемых агентом  $A$ ;

$t(F)$  - функция, определяющая переходы из состояния  $s_i$ , осуществляемые политиками  $f_1, \dots, f_i, \dots, f_m$  агента  $A$ .

$t(F)$

### 3.4 Модель сообщества агентов

Представляется очевидным, что модель сообщества агентов с одним выделенным элементом для восприятия внешних воздействий эквивалентна модели агента, рассмотренной в п.3.1. Модели сообществ с несколькими точками

восприятия внешних воздействий получаются простым соединением моделей для каждого выделенного элемента.

### **3.5 Анализ модели на примере конкретно взятого заказа**

Идея появления издательского центра в ЗГУ зародилась давно. Главным образом это связано с теми экономическими условиями, в которые был поставлен университет после распада СССР (нехватка литературы по ранее изучаемым предметам, а также в связи с резкой сменой идеологии по нововведенным предметам) и острой необходимостью печатать свои собственные (разработанные в самом университете) методические, учебные пособия, различные сборники. Кроме того, каждый уважающий себя вуз имеет, по крайней мере свою газету, которую также необходимо где-то печатать. Ну и, безусловно, престижно иметь свою, пусть небольшую, издательскую типографию. Университет к моменту появления издательства имел свою газету, свой журнал, т. е. он остро ощущал необходимость наличия в своей типографии. Раньше университет пользовался услугами других издательств города. И вот сейчас, начиная с сентября 1998 года, у нас есть свой издательский центр, деятельность которого в первую очередь направлена на своевременный выпуск наших университетских газет и журнала, плюс подготовка и изготовление различного рода объявлений, билетов на всевозможные мероприятия в рамках университета, а также многочисленные буклеты (например, во время пребывания у нас в университете министра иностранных дел Виктора Пинзеника в тот же самый день, после его выступления перед общественностью в актовом зале 2-го корпуса ЗГУ, ему был вручен на память буклет с информацией о нем самом). Естественно, что должна быть реализована не в последнюю очередь такая функция издательского центра ЗГУ, как печать методичек, пособий и другой научной литературы. И только после выполнения своих первоочередных функций центр имеет право брать внешние заказы.

Теперь представим себе, что клиент, посредством Интернет хочет узнать всю необходимую ему информацию, касающуюся реализации его желаний. Скажем, клиенту необходимо напечатать некий рекламный буклет в определенном тираже и с определенным текстовым и графическим сопровождением в предельно краткие сроки.

С помощью функциональной модели издательского центра он (клиент) может, не имея непосредственного контакта с работниками центра, узнать базовую информацию по интересующим его вопросам, используя лишь тот интерфейс, сосредоточием которого являются интеллектуальные агенты, наделенные обязанностями реальных служащих издательства.

Итак, пусть его заказ, как уже упоминалось выше, - это рекламная брошюра, объемом до 25-ти страниц. Предположим, что данная реклама касается фирмы, изготавливающей мебельные изделия, поэтому для этого буклета важно не только наполнение важной и всеохватывающей информацией, но наиболее важно создать визуальный интерфейс между людьми, получающими эту информацию при помощи данной рекламы и самой рекламной брошюрой (даже можно сказать «лицом» самой фирмы). Заказчик предоставляет необходимую текстовую информацию (не в электронном виде, т. е. в работу над данным заказом подключается группа наборщиков) и фотографии.

Вниманию клиента предоставляется интерфейс, с помощью которого он попадает в атмосферу виртуального издательского центра и общается с интеллектуальными агентами, формируя запросы, что соответствуют его требованиям, накладываемым на данный заказ.

Интерфейс в данной функциональной модели представлен в виде совокупности трех конечных автоматов. За каждым из них закреплены определенные функции. Первый осуществляет авторизацию политики внешнего воздействия в определенном состоянии. Второй конечный автомат осуществляет проверку соответствия параметров полученного воздействия на соответствие множеству системных параметров и ограничениям состояния. И, наконец, последний автомат осуществляет проверку соответствия описания ожидаемых

результатов тем результатам, которые могут быть получены при выполнении агентом в определенном состоянии своей политики.

Пусть его требования клиента таковы:

- тираж 2000 экземпляров;
- объем - 25-30 страниц;
- размещение всех предоставленных фотоснимков в цветном виде;
- применение дизайна, что соответствует специфике "лица" фирмы;
- срок выполнения заказа не должен превышать двух недель, начиная с завтрашнего дня;
- оптическое разрешение при печати не должно быть менее 600 dpi;
- стоимость одного экземпляра не должна превышать \$1.

Теперь проведем анализ данной ситуации в соответствии с нашей моделью. «Пройдемся» по той технологической цепочке, которая представлена на рис.2.2 во второй главе. Интеллектуальный агент, выполняющий роль руководителя издательского центра в самом начале определяет спецификацию заказа и оценивает возможность его выполнения в данных условиях. Этот заказ, например, может быть выполнен лишь при помощи офсетной печати или шелкографии, потому как на ризографе нет возможности реализации наличия в рекламном буклете высококачественных цветных фотоснимков. Отсюда – цена печати в среднем при таком тираже, если считать на экземпляр буклета, будет выше, но зато будет достигнута цель качественного представления информации. Если говорить о нашей типографии, то данный заказ может быть выполнен при помощи недавно приобретенного офсета.

При таком количестве страниц возможно скрепление экземпляра продукции при помощи специального оборудования металлическими скрепками, а значит, возможно снижение средней цены на единицу экземпляра за счет того, что не будет применен достаточно дорогой (но очень качественный) клей и специальное оборудование (клеевая машина производства Бельгии) для другого вида переплета, который применяется в случае больших объемов (более 30 страниц).

Предположим, что клиент хочет сделать свой заказ в то время, когда типография загружена первоочередной работой (верстка и печать газеты ЗГУ, которая по плану должна выходить один раз в две недели).

В соответствии со всем выше сказанным и со схемой принятия заказа на рис. 2.2, а так же теми ролями интеллектуальных агентов, совокупность которых представляет наше виртуальное сообщество – издательский центр, можно сказать следующее. Модель сообщества строится, исходя из модели функционального элемента сообщества – агента. Т. е., первое, что делает клиент – это формирует запрос, где указывает свои пожелания, все остальное «делает» модель (обработка и анализ этого запроса в соответствии возможностями и сложившейся ситуации к данному моменту). В конечном итоге клиент получает ответ от информационной среды, с которой он ведет диалог. Этот ответ может быть следующим:

- Принятие всех условий клиента и подписание контракта.
- Заказ принимается, но с учетом того, что оплатится сверхурочная работа.
- Заказ принимается, но с корректировкой некоторых требований клиента.
- Заказ не принимается.

В соответствии со схемой рис. 2.2 политики ролей каждого из интеллектуальных агентов будут следующими (напомним, что здесь приняты следующие обозначения: директор центра (**B**), заместитель по производству (**P**), наборщик (**T**), дизайнер (**D**), инженер верстки (**M**), web-инженер (**W**))

Директор центра.

1. Принять заказ от клиента, определив его спецификацию.
2. Сформировать запрос (описать внешнее воздействие).
3. Определить и предоставить на рассмотрение для T, D, M входные параметры внешнего воздействия.
4. Описать и предоставить T, D, M ожидаемые результаты.
5. Получить ответы от T, D, M.

В случае негативного ответа со стороны одного из агентов T, D, M отказ от заказа, как невыполнимого в рамках данного сообщества в текущем состоянии.

В случае позитивных ответов со стороны агентов T, D, M:

сформировать запрос Р о сроках выполнения заказа и его стоимости

сформировать и предоставить для Р вектор параметров, описывающий данный заказ с тем, чтобы Р мог определиться со сроками выполнения данного заказа и его стоимостью

получить ответ от Р

предоставить полученную от Р информацию клиенту.

Когда клиент полностью согласен со всей предоставленной информацией, то руководитель делает запрос к заместителю по поводу подписания контракта и внесения его в общий план производства. Когда же клиент не согласен с какой-либо предоставленной информацией (ценой, сроками, видом уже изготовленной в будущем продукции (возражения критичны)), то передаются запрос, а на сколько данные возражения критичны, и после ответов каждого из агентов моделью производится анализ сложившейся ситуации, и в результате выдается ответ

Сформировать запрос к Р, Т, М, D, W о виде несогласия (о критических моментах).

Сформировать и предоставить параметры, описывающие критичность, описать ожидаемые от Р, Т, М, D, W результаты.

Получить ответы от Р, Т, М, D, W

Провести детальный анализ полученных ответов и оформить ответ для клиента

В случае критичного ответа, когда ответ не критичен:

Отказ от заказа сформировать компромиссное предложение клиенту

Заместитель по производству.

1. Получить запрос от В в случае полного соответствия спецификации заказа работе издательского центра о сроках и цене.

2. Определить сроки и цену заказа.

3. Дать результативный ответ В об определенных сроках выполнения заказа и его цене.

4. Если со стороны клиента возникли критические возражения о том, как будет выглядеть готовая продукция, о ее сроках, цене, получить запрос от В

5. Определить входит ли реакция на данный запрос в сферу собственных полномочий.

6. Определить параметры воздействия, которые относятся к сфере собственных полномочий.

7. Определить, какие из параметров влияют на сформированное значение.

8. Сформировать результат и отправить его В.

9. Если В и клиент пришли к обоюдному согласию относительно заказа и подписали контракт, то добавить этот заказ в план производства.

Наборщик.

1. Получить запрос от В при поступлении заказа от клиента об удовлетворении данным заказом возможностям производства.

2. Определить входит ли реакция на данный запрос в сферу собственных полномочий.

3. Определить параметры воздействия, которые относятся к сфере собственных полномочий.

4. Сформировать ответную реакцию в терминах ожидаемого результата.

5. Если со стороны клиента возникли критические возражения, то получить соответствующее воздействие то В.

6. Определить входит ли реакция на полученное воздействие в сферу собственных полномочий.

7. Определить параметры воздействия, которые относятся к сфере собственных полномочий.

8. Определить, какие результаты из запрошенных он может сформировать, указав какие из параметров влияют на сформированное значение.

9. Сформировать результат и предоставить его В.

Дизайнер.

1. Получить запрос от В при поступлении заказа от клиента об удовлетворении данным заказом возможностям производства.

2. Определить входит ли реакция на данный запрос в сферу собственных полномочий.

3. Определить параметры воздействия, которые относятся к сфере собственных полномочий.

4. Сформировать ответную реакцию в терминах ожидаемого результата.

5. Если со стороны клиента возникли критические возражения, то получить соответствующее воздействие то В.

6. Определить входит ли реакция на полученное воздействие в сферу собственных полномочий.

7. Определить параметры воздействия, которые относятся к сфере собственных полномочий.

8. Определить, какие результаты из запрошенных он может сформировать, указав какие из параметров влияют на сформированное значение.

9. Сформировать результат и предоставить его В.

Инженер верстки.

1. Получить запрос от В при поступлении заказа от клиента об удовлетворении данным заказом возможностям производства.

2. Определить входит ли реакция на данный запрос в сферу собственных полномочий.

3. Определить параметры воздействия, которые относятся к сфере собственных полномочий.

4. Сформировать ответную реакцию в терминах ожидаемого результата.

5. Если со стороны клиента возникли критические возражения, то получить соответствующее воздействие то В.

6. Определить входит ли реакция на полученное воздействие в сферу собственных полномочий.

7. Определить параметры воздействия, которые относятся к сфере собственных полномочий.

8. Определить, какие результаты из запрошенных он может сформировать, указав какие из параметров влияют на сформированное значение.

9. Сформировать результат и предоставить его В.

Web-инженер.

1. Если со стороны клиента возникли критические возражения, то получить соответствующее воздействие то В.

2. Определить входит ли реакция на полученное воздействие в сферу собственных полномочий.

3. Определить параметры воздействия, которые относятся к сфере собственных полномочий.

4. Определить, какие результаты из запрошенных он может сформировать, указав какие из параметров влияют на сформированное значение..Сформировать результат и предоставить его В.

## ВЫВОДЫ

В процессе дипломной работы были решены следующие задачи и получены следующие результаты: исследованы существующие подходы к решению задачи моделирования функциональной проекции подразделения виртуального предприятия для корпоративных информационных сетей; в результате анализа было выяснено, что одним из лучших вариантов реализации функциональной проекции есть способ комбинации таких математических аппаратов, как конечные автоматы и метод расчленения сложной системы на более мелкие и легко реализуемые. И естественно, один из плюсов данного подхода реализации функциональной проекции подразделения предприятия это применение интеллектуальных информационных агентов, их ролей, политик и взаимодействия.

В результате мы формализовали функциональный элемент сообщества – основной строительный блок модели. Также определили правила взаимодействия функциональных элементов (агентов) и описали всевозможные множества состояний и эволюцию этих элементов модели.

В конце концов, на базе построенной модели отдельно взятого интеллектуального агента, применения диакоптического подхода и анализа взаимодействия всех агентов мы смогли построить формальное представление сообщества интеллектуальных агентов, на основе которого моделируется функциональное подразделение предприятия. Т. е., мы распространили модель агента на сообщество в целом.

.Широкое использование информационных технологий, построение информационного общества на нашей планете, интенсивные процессы интеграции во многих гуманитарных, технологических и научных направлениях, глобализация информационных обменов - вот тот базис, который придает все большую актуальность и все большее значение исследованиям в области интеллектуальных информационных агентов и практическому применению результатов этих

исследований в деле интеграции и повышения мобильности человеческого сообщества.

- Результаты исследований по данной теме уменьшат пробелы в наших знаниях и умении в деле реализации таких сложных и практически полезных систем, как виртуальные информационные пространства и виртуальные предприятия, данные результаты широко востребованы в областях, начиная с высшего образования и заканчивая индустрией и коммерцией.

Разработанный нами аппарат может быть применен к моделированию не только функциональных элементов УИП, но и процессов - информационных потоков, координации работ, сетевого планирования и т.д. а положительным качеством данной модели является учет динамического характера функциональной проекции УИП.

После реализации виртуального информационного пространства для среды университета применить полученные результаты можно применить к улучшению и интенсификации процедур управления университетом, а также для поддержки процессов подготовки и принятия решений. А также применить эти результаты для создания новой электронной среды обучения: дистанционное обучение, новые методы преподавания, новые формы учебных пособий и т.д.

Понятно, что все это способствует:

- улучшению качества образования (средства дистанционного обучения, глобальный распределенный доступ к библиотечным ресурсам, разработка электронных учебников и других пособий, реализация активных процедур и средств обучения и тестирования) и, таким образом, подъему культурного уровня, увеличению профессиональной квалификации, мотивации поколения будущего;
- повышению эффективности и современности процедур управления предприятием/университетом и, таким образом, помощи в придании экономике Украины черт большей предсказуемости, рыночности, постоянного роста и большей демократичности.

## ПЕРЕЧЕНЬ ПРИНЯТЫХ В ДИПЛОМЕ СОКРАЩЕНИЙ

CGI - Common Gateway Interface Универсальный шлюзовый интерфейс

FTP - File Transfer Protocol протокол передачи данных

HTML – Hyper Text Markup Language язык разметки гипертекстовых документов

HTTP - Hyper Text Transfer Protocol протокол передачи гипертекста

ICRF - Imperial College Role Framework

MIME -

SQL – язык запросов

TCP/IP -

URL – Universal Resource Name общая система адресации, универсальный указатель местоположения ресурсов

UVII - Unified Visual Intranet Interface Унифицированный Визуальный Интерфейс Intranet

VEMS - Virtual Enterprise Management Systems Система Управления Виртуальными  
Предприятиями

WWW - World Wide Web. «всемирная паутина»

WYSIWYG - What You See Is What You Get

БД – база данных

ВУ – Виртуальный Университет

ЕИП – Единое Информационное Пространство

ЕИПУ - Единое Информационное Пространство Университета

ЗГУ – Запорожский Государственный университет

ИС – Информационная Система

КИС – Корпоративная Информационная Система

НТР – научно-техническая революция

СУБД – Система Управления Базами Данных

СУВП - Система Управления Виртуальным Предприятием

УИП – Унифицированное Информационное Пространство

ЭВМ – электронно-вычислительная машина

## СПИСОК ССЫЛОК НА ЛИТЕРАТУРУ

1. Dewey et. al. The Impact of NIIP Virtual Enterprise Technology on Next Generation Manufacturing. In Proc. of Conference on Agile and Intelligent Manufacturing Systems, 1996
2. Ermolayev, V. A., Pletsky, S. U., Tolok, V. A., Architecture of Unified Information Space of a Virtual University. Lecture Notes of Zaporozhye State University, Vol. 1, No 2, 1998, pp. 44-53. (in russian).
3. R. Otte, P. Patrick, M. Roy. Understanding Corba, 1/e. - Prentice Hall Professional Technical Reference, ISBN 0-13-459884-9, 1996, 288 pp.
4. Davulcu, H., Kifer, M., Pokorny, L. R., Ramakrishnan, C. R., Ramakrishnan I. V., Modeling and Analysis of Interactions in Virtual Enterprises. Submitted to 9-th International Workshop on Research Issues on Data Engineering, March, 1999
5. Lupu, E., Milosevic, Z., Sloman, M., Use of Roles and Policies for Specifying and Managing a Virtual Enterprise. Submitted to 9-th International Workshop on Research Issues on Data Engineering, March, 1999
6. Baral, C., Lobo, J., Formalizing Workflows as Cooperative Agents In Proc. of DYNAMICS 97 (a workshop in ILPS 97).
7. В.И. Гриценко, А.А. Щипцов, А.В. Ищенко, Ю.Г. Авдюшенко, В.Н. Никулин, С.А.Тарасенко, А.А. Урсатьев Современные подходы к созданию распределенных информационных систем, Управляющие Системы и Машины, №3, май – июнь 1998, с.3-9.
8. EDUNET '97 Сборник материалов международной конференции, Киев, 1997 г
9. Компьютер Пресс
10. Компьютер Пресс
11. Компьютер Пресс

12. V. A. Tolok, S. U. Borue, V. A. Ermolayev, A. I. Kubushkaite, Development of the Concept and Implementation of the First Line of the Integrated Network at ZSU, Research Work Intermediate Report, State Reg. No 0197y012776, Ministry of Education of Ukraine, Zaporozhye State Univ., Zaporozhye, 1997, 28 p.
13. Schmitt, G. Saake, Merging Inheritance Hierarches for Schema Integration Based on Concept Lattices, Fakultät für Informatik, Otto-von-Guericke-Universität, Magdeburg, Preprint Nr. 2, 1997
14. V. A. Ermolayev, Object Oriented Dynamic Data Modelling and Active Data Dictionaries - Some Crosspoints . - to appear in "Journal of Metrology and Certification" Vol 1, No 1, (Jul.-Dec. 1997)
15. Kron, G., Diacoptics. Macdonald, London, 1963
16. Диментберг Ф. М., применение метода динамических жесткостей для расчета связанных колебаний. Сб. "Динамика и прочность коленчатых валов", Изд-во АН СССР, М., 1948.
17. Терских В. П., Метод цепных дробей, тт. I, II, III. Судпромгиз, М., 1955.
18. Дондошанский В. К., К теории колебаний цепных систем. Труды Лесотехнической академии, вып. 94, 1962.
19. Троицкий В. А., О расчете свободных колебаний пространственных стержневых систем методом динамической жесткости. Инф. Бюлл. ЛПИ им. Калинина, №5, 1957.
20. Баранов А. В., Элементы теории систем с распределенными связями. Вестник МГУ, серия III, №5, 1965.
21. Bishop R. E. D., The mechanics of vibration. Cambridge University Press, Cambridge, 1960.
22. Okumura A., On a method of analysis for vibration and stability problems of linear mechanical systems or structures. Mem. School Sci. And Engng., Wasenda Univ., v. 21, 1957.

23. Dantzig G. B., Wolfe P., Decomposition principle for linear programs. Oper. Res. V. 8, №1, 1959.
24. Kron G., Diacoptics – a gateway into universal engineering/ The Electrical Journal, v. 156, Dec. 28, 1956.
25. Demirchan, K. S., Rakitsky, U. V., Butuirin, P. A., Kartashev, E. N., Korovkin, I. V., Problems of Numerical Modeling of the Processes in Electric Circuits. Lecture Notes of the Academy of Sciences of the USSR. Energetics and Transport. No 2, 1982, pp. 94-114. (in russian).
26. Butuirin, P. A., Borue, S. U., Analytical Transformations of Characteristic Equations of Electrical Machines. Lecture Notes of the Academy of Sciences of the USSR. Energetics and Transport. No 2,
27. Glushkov, V. M., Elementary Cybernetics, Ukrainian Academy of Sciences, Kiev, 1964, 323 pp. (in russian).
28. В. Б. Кудрявцев, С. В. Алешин, А. С. Подколзин, Элементы теории автоматов, МГУ, механико-математический факультет, 1978г.
29. Яблонский С.В. Основные понятия кибернетики. Сб. Проблемы кибернетики, вып. 2, М. 1959, стр. 23.
30. Deen, S. M., An Architectural Framework for CKBS Applications, IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 3, June 1996, pp. 663 - 671
31. Storey, V. C., Dey, D., Sundaresan, S., Ullrich, H. and Yamakawa, S., Learning Accross Application Domains for Database Design Systems. Proc. of the 6-th Workshop on Information technologies and Systems (WITS'96), Cleveland, Ohio, 1996.

1.