

### **ГЛАВА 3.**

## **СИТУАЦИОННАЯ МОДЕЛЬ РЫНКА ПОЛИГРАФИЧЕСКИХ УСЛУГ**

Во второй главе предложена методика интеграции систем ситуационного, имитационного и экспертного моделирования. Ее третьим этапом является создание семиотической (ситуационной) модели предметной области и разработка языка взаимодействия с ней.

В первой главе рассмотрены различные способы описания и моделирования ситуаций. Наиболее наглядными и удобными для использования в ЭС являются такие ЯПЗ, как семантические сети и логика предикатов. Семиотический подход часто используют при создании гибридных языков и моделей, к числу которых можно отнести описываемую в следующей главе SIE-модель.

В этой главе приведен пример создания ситуационной модели рынка полиграфических услуг (СМ РПУ). Описание модели выполнено с помощью логики предикатов первого порядка с ситуационными расширениями. Такое описание может быть использовано на экспертном уровне SIE-модели.

В описании приводятся основные объекты полиграфического рынка, определяются возможные микроситуации и классы макроситуаций. СМ РПУ предназначена для использования в системе ситуационного моделирования (ССМ), которая включает две основные подсистемы — систему ситуационной поддержки руководителя (ССПР) и систему ситуационной поддержки оператора (ССПО).

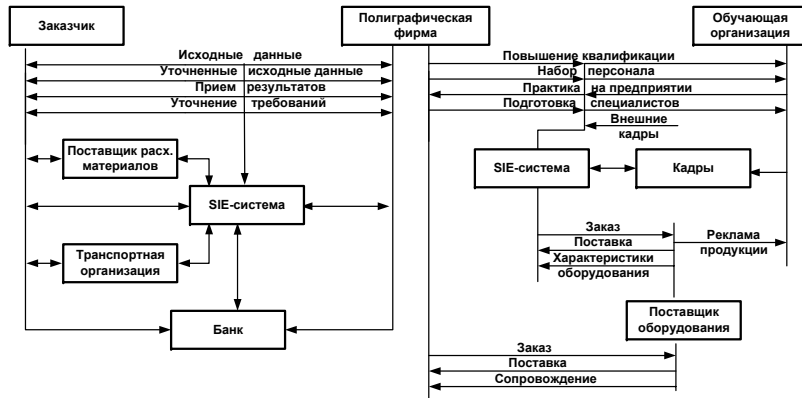
ССПР предназначена для наблюдения за обстановкой на полиграфическом рынке руководителями высшего ранга. Если использовать классификацию, то подсистема относится к классу стратегического или оперативного ситуационного центра. В ней возможно наблюдение и прогнозирование ситуации в целом, отдельном срезе или выбранном фрагменте.

ССПО предназначена для взаимодействия операторов с клиентами. Она работает с конкретными объектами, с ее помощью устанавливаются связи между заказчиками, исполнителями, обучающими институтами, поставщиками оборудования и др.

Операторы могут пользоваться вспомогательными средствами поддержки принятия решения (экспертными системами, системами имитационного моделирования, контактными звонками и т.д.), но основным источником информации служит ситуационная модель (СМ). Операторы имеют возможность изменять характеристики отображения в СМ (ус-

танавливать приоритеты, вносить новые объекты, изменять и удалять старые).

Рис.3.1. Структура рынка полиграфических услуг.



### 3.1. Общие принципы построения ситуационной модели РПУ

Рассмотрим составляющие полиграфического рынка услуг как компоненты ситуационной модели, используемые в работе ССМ РПУ. На рис. 3.1 представлена структура полиграфического рынка.

Ситуационная модель (СМ) включает следующие классы основных объектов, которые могут находиться в различных состояниях: *Исполнитель; Заказчик; Обучающий институт; Поставщик оборудования; Поставщик расходных материалов; Транспортная организация; Банк расчетов; Банк кадров; Управляющая система; Физическое лицо; Юридическое лицо; Форс-мажорные обстоятельства.*

**Основные объекты.** Каждый объект полиграфического рынка необходимо представить в виде элемента ситуационной модели. Рассмотрим пример описания объекта *Клиент (заказчик)*.

Пусть имеется определенное количество клиентов (заказчиков услуг или продукции). Для их представления в ситуационной модели введем массив объектов типа *Заказчик*. Для реализации множественности будем использовать предикат вида:

*Заказчик(x)*, где *x* — любой объект ситуационной модели.

Формально предикату соответствует следующее математическое выражение:

$$Z^k = \{x_1^{zk}, x_2^{zk}, \dots, x_{n_{zk}}^{zk}\} = \bigcup_1^{n_{zk}} x_i^{zk} \wedge \left( x_i^{zk} \in \bigcup_1^{\infty} x \right) \wedge (i = \overline{1, n_{zk}})$$

$Z^k$  – класс заказчиков;  
 $x_i^{zk}$  – конкретный заказчик;  
 $n_{zk}$  – количество заказчиков;  
 $\bigcup_1^{\infty} x$  – множество всех объектов СМ.

Предикат истинен, если объектом является заказчик продукции или услуг, и ложен, если это не так.

$Заказчик(x) = True$	$x \in \{Заказчик\}$
$Заказчик(x) = False$	$x \notin \{Заказчик\}$

$$(\forall x_j \in \bigcup_1^{\infty} x)$$

$$((Z^k(x_j) = True \Leftrightarrow x_j \cap Z^k \neq \emptyset) \vee (Z^k(x_j) = False \Leftrightarrow x_j \cap Z^k = \emptyset))$$

Представление объектов в СМ является типовым. Каждому объекту соответствует одноместный предикат, который можно применить для любого элемента СМ. Обобщенное описание всех основных объектов представлено в табл. 3.1.

<b>Предикат(x) = True</b>	<b>Если x ∈</b>
Заказчик	{Заказчик}
Исполнитель	{Исполнитель}
Поставщик Оборудования	{Поставщик Оборудования}
Обучающий Институт	{Обучающий Институт}
Поставщик Расходных Материалов	{Поставщик Расходных Материалов}
Транспортная Организация	{Транспортная Организация}
Банк Расчетов	{Банк Расчетов}
Банк Кадров	{Банк Кадров}
Управляющая Система	{Управляющая Система}

Табл. 3.1. Основные объекты СМ.

Банк кадров используется для обеспечения полиграфических предприятий персоналом. В него заносится информация о специалистах, желающих найти работу.

СМ рассматривается как объект типа *Управляющая Система*. Если система имеет несколько подразделений, то необходимо ввести несколько объектов данного типа.

**Микроситуации.** Каждый из основных объектов может находиться в любой момент времени в одном из состояний (микроситуаций). Кроме того, необходимо ввести еще ряд микроситуаций, связанных с непрогнозируемыми событиями (*Наводнение, Пожар, Перекрытие дороги, Падение валютного курса*). Назовем класс объектов, отвечающих за эти события, *Форс-мажорные Обстоятельства*.

Для представления микроситуаций используется двухместный предикат, который можно применить к любому объекту СМ. Например, состояние заказчика можно рассматривать как микроситуацию *Mis\_Заказчик(x, mis)*. Количество различных состояний ограничено и определяется следующим образом:

$$\{mis\_zak\}_{m1} = \{Запрос, Заказ, Сопряжение, Взаимодействие, Ожидание, Отказ\}$$

$$Z_{ms}^k = \{\xi_{zap}^{zk}, \xi_{zak}^{zk}, \xi_{sopr}^{zk}, \xi_{vzim}^{zk}, \xi_{ozid}^{zk}, \xi_{otk}^{zk}\}; (\forall \xi_j \in Z_{ms}^k) \left( \xi_j \cap \bigcup_{\infty} \xi \neq \emptyset \right)$$

*Запрос* — это состояние заказчика с момента появления запроса от клиента в системе до начала обслуживания.

*Заказ* — это состояние клиента в момент обслуживания, т.е. после начала работы с ним оператора оформления и расчета заказа.

*Сопряжение* — это состояние клиента после определения характеристик заказа до установления контактов с исполнителем. Это состояние может быть длительным, если заказчик размещает заказ на будущее или ожидает изменение условий.

*Взаимодействие* — это состояние клиента в момент передачи исходных данных заказчику и проверки результатов.

*Ожидание* — это состояние клиента в момент выполнения услуг исполнителя. Это состояние может смениться либо состоянием взаимодействия, либо отказа, либо завершением работы.

*Отказ* — это состояние клиента после отказа от предоставляемых услуг до момента корректного завершения его взаимодействия с исполнителем.

$Mic\_Заказчик(x, mis) = True$	$x \in \text{Заказчик}$
--------------------------------	-------------------------

$$Z_{ms}^k(\xi_j, x_i) = True \Leftrightarrow (x_i \cap Z^k \neq \emptyset) \wedge (\xi_j \cap Z_{ms}^k \neq \emptyset)$$

Различные типы объектов могут находиться в одинаковых состояниях, поэтому в СМ можно создать единое множество микроситуаций. На основании этого множества формируются пересекающиеся подмножества микроситуаций отдельных объектов. Микроситуации для основных объектов описаны в табл. 3.2

<b><i>Mic_Предикат(x,mis) = True</i></b>	<b><i>Если (mis ∈ ...) &amp; (Предикат(x) = True)</i></b>
<i>Mic Заказчик</i>	$\{mis\ zak\}_{m1}$
<i>Mic Исполнитель</i>	$\{mis\ isp\}_{m2}$
<i>Mic Поставщик Оборудования</i>	$\{mis\ edu\}_{m3}$
<i>Mic Обучающий Институт</i>	$\{mis\ rash\}_{m4}$
<i>Mic Поставщик Расходных Материалов</i>	$\{mis\ tech\}_{m5}$
<i>Mic Транспортная Организация</i>	$\{mis\ trans\}_{m6}$
<i>Mic Банк Расчетов</i>	$\{mis\ bank\}_{m7}$
<i>Mic Банк Кадров</i>	$\{mis\ pers\}_{m8}$
<i>Mic Управляющая Система</i>	$\{mis\ sys\}_{m9}$
<i>Mic Форс-мажорные Обстоятельства</i>	$\{mis\ fors\}_{m10}$

Табл. 3.2. Представление микроситуаций для основных объектов.

$\{mis\_zak\}_{m1}$  — множество микроситуаций всех заказчиков;  
 $mis\_zak_i = Muc\_Заказчик(x, mis)$   
 $\{mis\_isp\}_{m2}$  — множество микроситуаций всех исполнителей;  
 $mis\_isp_i = Muc\_Исполнитель(x, mis)$   
 $\{mis\_edu\}_{m3}$ , — множество микроситуаций всех Обучающих институтов;  
 $\{mis\_rash\}_{m4}$ , — множество микроситуаций всех Поставщиков расходных материалов;  
 $\{mis\_tech\}_{m5}$ , — множество микроситуаций всех Поставщиков оборудования;  
 $\{mis\_trans\}_{m6}$ , — множество микроситуаций всех Транспортных организаций;  
 $\{mis\_bank\}_{m7}$ , — множество микроситуаций всех Банков расчетов;  
 $\{mis\_pers\}_{m8}$ , — множество микроситуаций всех Банков кадров;  
 $\{mis\_sys\}_{m9}$ , — множество микроситуаций Управляющей системы;  
 $\{mis\_fors\}_{m10}$  — множество микроситуаций всех Форс-мажорных обстоятельств;  
 $m_i$  — размерность множества микроситуаций для соответствующих объектов.

Ниже приводятся еще несколько фрагментов из полного описания множества микроситуаций.

$\{mis\_isp\}_{m2} = \{занят, не занят, не работоспособен, ожидание, отказ\}$

*Занят* — это состояние производителя (исполнителя), когда он не может принимать новые заказы;

*Не занят* — это состояние производителя (исполнителя), когда он может принимать новые заказы;

*Не работоспособен* — это состояние производителя (исполнителя), когда он не может выполнять заказы из-за технических или других обстоятельств;

*Ожидание* — это состояние производителя (исполнителя), когда он работоспособен, но не выполняет заказ;

*Отказ* — это состояние исполнителя после отказа от выполняемых услуг до момента корректного завершения его взаимодействия с заказчиками.

$\{mis\_edu\}_{m3} = \{Подготовка\_специалистов, Стажировка, Отказ\}$

*Подготовка специалистов* — это состояние учебного заведения, когда оно осуществляет подготовку специалистов, т.е. во время учебного процесса;

*Стажировка* — это состояние учебного заведения, когда подготавливаемые кадры проходят практику или стажировку внутри какой-либо организации;

*Отказ* — это состояние учебного заведения, когда оно отказывается предоставить кадры по заданной специальности. Такая ситуация может возникнуть в случае переквалификации специалистов, изменения учебных планов и др.

$\{mis\_pers\}_{ms} = \{\text{Пустой}, \text{Переполнение}, \text{Средний\_Уровень}\}$

*Пустой* — это состояние банка кадров, когда в нем отсутствуют кадры или их количество невелико;

*Переполнение* — это состояние банка кадров, когда в нем находится слишком много "безработных";

*Средний\_Уровень* — это среднее количество кадров, находящихся в банке.

**Функции ССМ.** Выделим основные функции, которые может выполнять ССМ в рамках ССПО.

1. *Поиск информации.* Поиск обычно выполняет оператор во время диалога с клиентом. Для отображения информации и взаимодействия с СМ требуется использование специального языка и интерфейса.

Для того чтобы использование поисковых функций было доступно любому пользователю, необходимо разработать соответствующее программное обеспечение. Оно должно иметь ограничения к доступу данных и работать с удаленной БД по технологии Интернет/Инtranет.

2. *Функция посредника (Контактная функция).* Схема взаимодействия в качестве посредника между заказчиком и исполнителем полиграфических услуг представлена на рис. 3.2 и содержит следующие объекты:

*Исполнитель; Заказчик (Клиент); Поставщики расходных материалов; Транспортная организация; Управляющая система.*

Данная функция позволяет найти подходящие варианты для заказчика и установить контакты между исполнителем, клиентом, поставщиком и транспортной организацией.

Как видно из рис. 3.2 взаимодействие исполнителя заказа с поставщиками расходных материалов (ПРМ) и транспортной организацией (ТРО) может осуществляться самостоятельно или через ССМ (управляющую систему). Если у клиента имеются свои возможности реализации взаимодействия с ПРМ и ТРО, то они учитываются в качестве исходных данных и контролю не подлежат.

Если исполнитель решит взаимодействовать по своим каналам с ПРМ и ТРО, которые входят в СМ, то необходимо контролировать только свободные ресурсы в соответствующих организациях. Полный

контроль обеспечивает большую надежность проводимых операций, но он снижает гибкость системы, которая является существенным преимуществом фирмы-посредника.

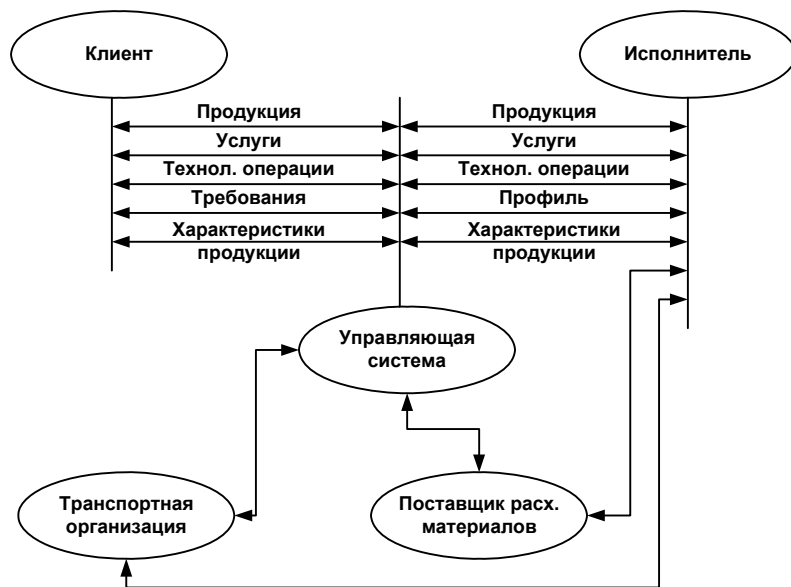


Рис. 3.2. Контактная функция ССМ.

Если исполнитель решит взаимодействовать по своим каналам с ПРМ и ТРО, которые входят в СМ, то необходимо контролировать только свободные ресурсы в соответствующих организациях. Полный контроль обеспечивает большую надежность проводимых операций, но он снижает гибкость системы, которая является существенным преимуществом фирмы-посредника.

В рамках этой функции определены следующие классы информационных объектов:

*Продукция, услуги, технологические операции, требования, характеристики продукции, характеристики перевозки, характеристики расходных материалов, профиль.*

3. *Функция ведения заказа.* Эта функция относится к этапу взаимодействия заказчика и исполнителя, во время которого ССМ осуществляет контроль за ходом выполнения заказа. В случае необходимости она может осуществлять функцию финансового посредника. При этом от-



крываются широкие возможности при работе с постоянными клиентами:

- взаимный зачет услуг;
- оплата в кредит;
- оплата в определенные моменты времени или при достижении критической суммы;
- защита клиентов от отказов исполнителя, блокировка финансовых средств и изменение рейтинга.

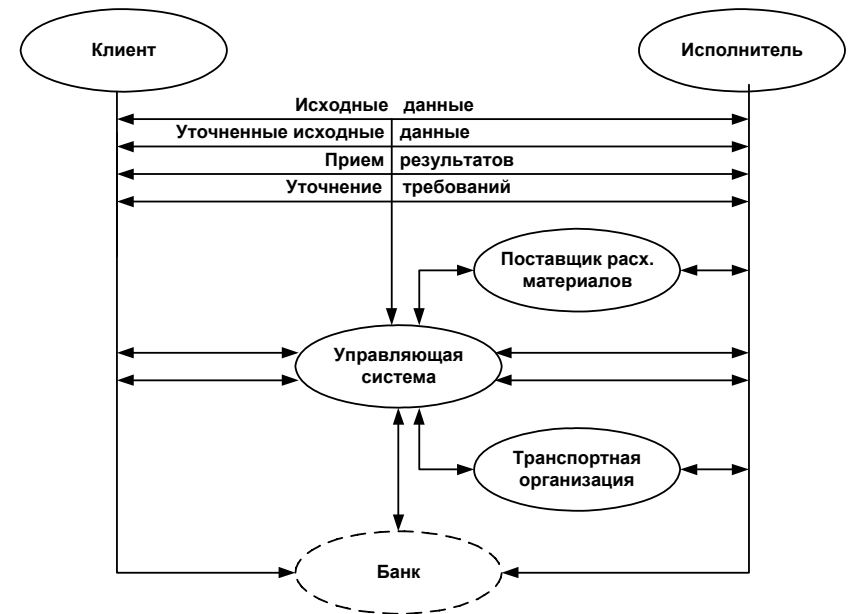


Рис. 3.2. Контактная функция ССМ.

Предполагается, что использование других финансовых посредников должно, по возможности, контролироваться. Функция ведения заказа подразумевает также легкий доступ к ПРМ и ТРО, а также заказ отдельных услуг в других организациях.

Между объектами передается информация. Все данные сгруппированы в несколько классов, каждому из которых соответствует своя категория в СМ. Определены следующие классы информационных объектов: *Продукция, Вид деятельности, Услуги, Технологические операции, Требования, Исходные данные, Уточненные исходные данные, При-*

ем результатов, Уточненные требования, Расчетные характеристики и др.

В табл. 3.3. приведены фрагменты множества информационных объектов.

<b>Предикат(x) = True</b>	<b>Если x ∈</b>
Продукция	<i>Prod_Бланк, Prod_Учебник, Prod_Рекламная_листовка, Prod_План, Prod_Календарь, Prod_Плакат, Prod_Обложка, Prod_Открытка, Prod_Газета, Prod_Буклет, Prod_Печать, Prod_Диплом, Prod_Таблица, Prod_Свидетельство, Prod_Пригласительный_билет, Prod_Удостоверение, Prod_Афиша, Prod_Визитная_карточка, Prod_Почетная_грамота, Prod_Презентация, Prod_Анкета, Prod_Сертификат, Prod_Папка, Prod_Вкладыш_в_аудиокассеты_и_ лазерные_диски, Prod_Эмблема, Prod_Монография, Prod_Ценник, Prod_Книга, Prod_Брошюра, Prod_Фирменный_бланк, Prod_Каталог, Prod_Блокнот, Prod_Наклейка, Prod_Лицензия, Prod_Другая</i>
Вид_Деятельности	<i>Prof_Допечатная_подготовка, Prof_Печать, Prof_Ручной_переплет, Prof_Послепечатная_обработка, Prof_Дизайн_студия, Prof_Фотовывод, Prof_Изготовление_штампов</i>
Услуги	<i>Serv_Сканирование, Serv_Верстка, Serv_Фотовывод, Serv_Цветопроба, Serv_Дизайн, Serv_Фотографирование, Serv_Съемка_на_видеокамеру, Serv_Печать_визиток, Serv_Изготовление_флексоформ, Serv_Тампонная_печать, Serv_Печать_на_ризографе, Serv_Изготовление_штампов, Serv_Брошюровка, Serv_Переплет,</i>

	<i>Serv_Тиснение, Serv_Резка, Serv_Склейка, Serv_Другая</i>
<i>Технологические_операции</i>	<i>Oper_Отрисовка_изображений, Oper_Респечатка_на_принтере, Oper_Корректурa, Oper_Обработка_изображений</i>
<i>Требования</i>	<i>Treb_Обязательная_доставка, Treb_Приоритетная_организация, Treb_Сроки, Treb_Качество</i>
<i>Характеристики_продукции</i>	<i>Prop_Красочность, Prop_Количество, Prop_Формат</i>
<i>Характеристики_перевозки</i>	<i>Trans_Расстояние, Trans_Вес, Trans_Пункт_доставки, Trans_Время_перевозки</i>
<i>Характеристики_расходных_материалов</i>	<i>Rash_Количество, Rash_Тип, Rash_Производитель, Rash_Цена</i>
<i>Исходные_данные</i>	<i>Isx_Текст, Isx_Изображения, Isx_Эмблема, Isx_Шрифт</i>
<i>Расчетные_характеристики</i>	<i>Bank_Название_банка, Bank_Сумма Bank_Номер_банковского_счета, Bank_Реквизиты, Bank_Платеж,</i>

Табл. 3.3. Информационные объекты СМ.

*Уточненные исходные данные.* Уточненные исходные данные имеют ту же структуру, что и класс объектов Исходные данные. Для их различия увеличена размерность предиката следующим образом:

<i>Исходные_данные(x=Isx_*, y=False)=True</i>	<i>Первоначальные исходные данные</i>
<i>Исходные_данные(x=Isx_*, y=True)=True</i>	<i>Уточненные исходные данные</i>

*Уточненные требования.* Аналогично рассматриваются уточненные требования.

<i>Требования(x=Treb_*, y=False)=True</i>	<i>Первоначальные требования</i>
<i>Требования(x=Treb_*, y=True)=True</i>	<i>Уточненные требования</i>

4. *Функция обеспечения производства.* Эта функция ССМ предназначена для обеспечения полиграфической организацией техникой и квалифицированным персоналом. Схема взаимодействия включает следующие объекты:

*Исполнитель; Обучающий институт; Поставщики оборудования;  
Банк кадров.*

ССПО реализует поиск и организацию учебных процессов для подготовки, переквалификации и повышения квалификации персонала полиграфических фирм.

В схеме, представленной на рис. 3.4, также включен банк кадров, который позволяет обеспечить пополнение кадров из внешних источников (неучтенные обучающие институты, сокращенные сотрудники и т.д.). Банк данных содержит информацию о физических лицах.

Роль поставщиков оборудования расширена за счет их связи с обучающими институтами. Это позволяет спланировать обучение специалистов перед закупкой нового оборудования, демонстрировать возможности техники и технологий, тестировать технику, проводить эксперименты.

Обучающий институт может готовить специалистов разного профиля. При этом кадры могут находиться на разных уровнях и стадиях подготовки в зависимости от срока обучения и других факторов. Тогда следует разделить обучающий институт на факультеты, кафедры, курсы повышения квалификации. Каждому необходимому элементу следует сопоставить отдельный объект.

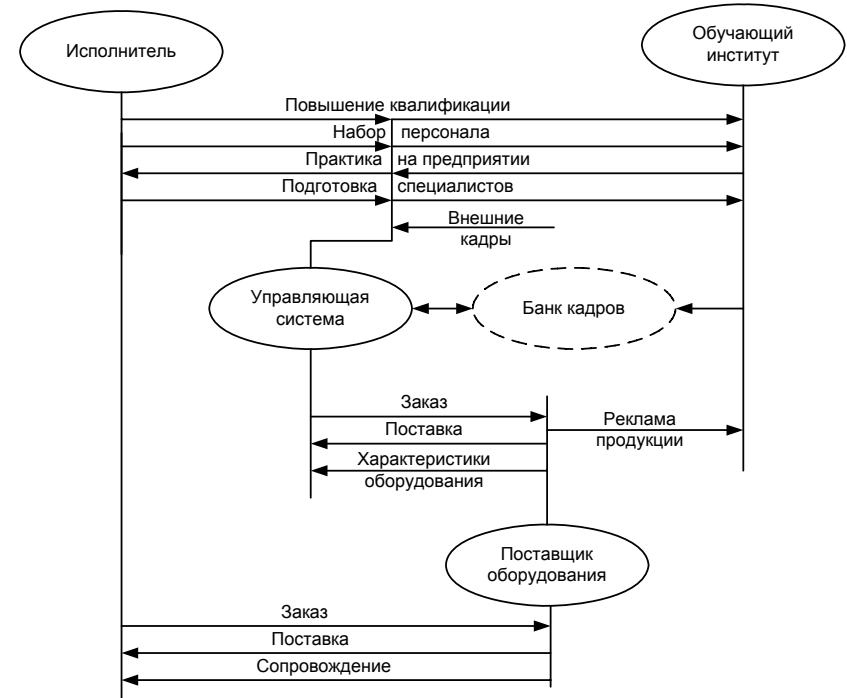


Рис. 3.4. Функция обеспечения производства.

### Срезы ситуационной модели

Важной задачей является определение возможности декомпозиции системы (СМ), так как количество элементов велико для отображения их всех. Каждая декомпозиция может соответствовать определенному срезу СМ, уровню детальности. Выделяются основные срезы:

*Пространственный срез.* Этот срез предназначен для отображения объектов полиграфического рынка с привязкой к географической карте.

*Деятельностный срез.* Этот срез позволяет выделить объекты по виду деятельности, отображение производится с привязкой к пространственному срезу. Аналогично могут осуществляться срезы по филиалам, объединениям, корпорациям.

*Производственный срез.* Этот срез позволяет выделить объекты по типу продукции, с которой работают предприятия. Он может отображаться вместе с деятельностным и пространственным срезами, так как тип продукции может совпадать для различных видов деятельности.

*Срез услуг.* Услуги или возможные технологические операции могут служить альтернативой производственного среза. Срез необходим для проектирования объектов, которые выполняют часть заказа, а не целиком. Каждому виду продукции и виду деятельности может быть поставлен набор необходимых услуг. Однако нельзя сопоставить виду деятельности однозначный список услуг. Он будет варьироваться в зависимости от ресурсов предприятия и навыков исполнителей. Основные услуги могут совпадать и считаться постоянными.

Описанные срезы являются составными, т.е. могут включать несколько критериев, наборов услуг, продуктов, видов деятельности. Каждый объект может принадлежать одновременно нескольким классам и содержать несколько характеристик. Особенностью данных характеристик является их бинарное значение, т.е. либо они есть, либо их нет.

*Временной срез.* ССМ предполагает режим отображения ситуации в реальном времени. Временные срезы позволят осуществлять прогнозирование и анализ предыдущих ситуаций. Кроме того, загрузку оборудования или время действия критериев удобно задавать дискретными интервалами, а не осуществлять запрос в реальном режиме времени или при изменении модельного времени на определенную величину. Это позволит существенно разгрузить ресурсы ССМ.

*Ценовой срез.* Срез является принципиальным и может задаваться диапазоном или точечными значениями. Цена должна вычисляться как сумма затрат на все необходимые услуги. Этот срез является одним из наиболее сложных, так как формирование цены зависит от большого числа параметров: объем тиража, сроки, стоимость материала и др.

#### **Использование макроситуаций**

Общее число микроситуаций в системе зависит от количества объектов в системе. Количество объектов заданных типов будет определяться соответствующими переменными:

*Количество\_Заказчиков;*

*Количество\_Исполнителей;*

*Количество\_Обучающих\_Институтов;*

*Количество\_Поставщиков\_Оборудования;*

*Количество\_Поставщиков\_Расходных\_Материалов;*

*Количество\_Транспортных\_Организаций;*

*Количество\_Банков\_Расчетов;*

*Количество\_Банков\_Кадров;*

*Количество\_Элементов\_Управляющей\_Системы.*

Для каждой макроситуации, точнее говоря, во время макроситуации все объекты могут находиться только в одной из микроситуаций. Тогда макроситуация  $S$  зависит от суммарного количества микроситуаций всех объектов СМ.

$$S = \text{func}(\{ \text{Мис\_Заказчик} \}_{n_1}, \{ \text{Мис\_Исполнитель} \}_{n_2}, \{ \text{Мис\_Обучающий\_институт} \}_{n_3}, \{ \text{Мис\_Поставщик\_Оборудования} \}_{n_4}, \\ \{ \text{Мис\_Поставщик\_Расходных\_Материалов} \}_{n_5}, \\ \{ \text{Мис\_Транспортная\_Организация} \}_{n_6}, \{ \text{Мис\_Банк\_Расчетов} \}_{n_7}, \\ \{ \text{Мис\_Банк\_Кадров} \}_{n_8}, \{ \text{Мис\_Управляющая\_Система} \}_{n_9}, \{ \text{Мис\_Форс-мажорные\_Обстоятельства} \}_{n_{10}})$$

где  $n_i$  — количество элементов заданного типа.

В общем случае макроситуацию можно определить с помощью ЭС. Приведем примеры некоторых группы макроситуаций:

*Мас\_Отказ*( $x$ ), где *Исполнитель*( $x$ )=*True*;

*Мас\_Переполнение*( $x$ ), где *Исполнитель*( $x$ )=*True*;

*Мас\_Появление\_исполнителя*( $x$ ), где *Исполнитель*( $x$ )=*True*;

*Мас\_Удаление\_исполнителя*( $x$ ), где *Исполнитель*( $x$ )=*True*;

*Мас\_Снижение\_числа\_заказов*;

*Мас\_Возрастание\_числа\_заказов*;

*Мас\_Простой*( $x$ ), где *Исполнитель*( $x$ )=*True*;

*Мас\_Неплановая\_ситуация*.

### 3.2. Ситуационный язык

Взаимодействие субъекта с ситуационной моделью требует разработки соответствующего языка, который может быть реализован различными средствами: в виде текстового запроса на ограниченном естественном языке (семиотическом), в виде исчисления предикатов, с использованием визуальных (нажатием кнопок, выбором меню, наведением курсора) или других средств. В любом случае язык должен иметь следующие возможности:

- создавать новые и удалять старые объекты СМ;
- изменять характеристики существующих объектов СМ;
- выдавать подробную информацию об объектах;
- осуществлять многокритериальный поиск объектов.

Для описания ситуационного языка было выбрано операторное (предикатное) представление. Создание интерфейсов на естественном языке и дополнительных визуальных средств может быть реализовано на его основе. В состав ситуационного языка входят следующие операторы:

- оператор *WhatIs*, выдающий информацию об объектах;

- оператор *ListOf*, описывающий структуру вложенных объектов;
- оператор *Set*, позволяющий изменять характеристики объектов;
- оператор *Create*, создающий новые объекты СМ;
- оператор *Delete*, удаляющий ненужные объекты СМ;
- оператор *Select*, описывающий условия запросов к СМ.

### Синтаксис операторов

Все операторы начинаются с ключевого слова (например, *Create*) и представляют собой предикат. Конструкция операторов может состоять из необязательных компонент, которые заключаются в угловые скобки:

*<Where>*

В квадратные скобки заключаются конструкции, которые вводятся пользователем или выводятся программой (в конструкции *return*). Слова внутри скобок обозначают тип вводимых значений.

*[Object], [Name], [Type]*

В фигурные скобки заключаются допустимые значения, зарезервированные слова. Если вариантов несколько, то они перечисляются через запятую. Если вариант только один, то скобки могут опускаться.

*{Best, Worst, Random}*

Некоторые зарезервированные слова имеют большую длину, поэтому язык поддерживает их сокращенное обозначение. Необязательная концовка слова указывается в круглых скобках.

*Mic(rosituation), Mac(rosituation), Var(iable), Obj(ect)*

Конструкции могут сочетать несколько скобок. Порядок их следования должен выбираться следующим образом:

- скобки возможных конструкций  $\langle \rangle$ ,
- скобки множественности вариантов  $\{ \}$ ,
- скобки вводимых значений  $[ ]$ ,
- скобки сокращений  $( )$ .

*<{Mic(rosituation), Mac(rosituation), Var(iable), Obj(ect)}>*

В ситуационной модели используются иерархические объекты, которые могут содержать вложенные объекты. Для обозначения вложенности используется знак “/”.

*Object1/Object2/Object3*

Имена объектов, микроситуаций и других элементов СМ обозначаются буквами английского алфавита и могут состоять из нескольких слов соединенных между собой знаком нижнего подчеркивания.

*Moscow\_State\_University\_Of\_Printing\_Arts*

С помощью конструкции *Return* описываются сообщения возвращаемые системой после выполнения операторов.



*Оператор Create.* Оператор *Create* позволяет создавать объекты, микроситуации, макроситуации и переменные различного типа. Синтаксис допускает использование вложенных объектов.

*Create* <{*Mic(rosituation)*, *Mac(rosituation)*, *Var(iable)*, *Obj(ect)*}>  
<[*Object*]/> [*Name*] of [*Type*]

*Return* {*Microsituation*, *Macrosituation*, *Variable*, *Object*} <not> *created*  
<*Error* [*Type of error*]>

По умолчанию, если не указан класс, или используется конструкция для вложенного объекта, то автоматически создается объект. Если ошибочно указывается класс объекта, т.е. используется конструкция вложенности с вариантами *Mic(rosituation)*, *Mac(rosituation)* или *Var(iable)*, то выдается ошибка с кодом 0. Ниже приведен перечень возможных ошибок и их соответствие кодам<sup>1</sup>:

Код ошибки	Название ошибки	Описание
0	Syntax Error	Ошибка в синтаксисе конструкции. Неправильно набраны элементы конструкции или выбраны противоречивые комбинации.
1	Error Name	Ошибка в наборе имени. Используются недопустимые символы или объект не существует.
2	Error Parent	Ошибка вложенности. Один из предков не существует. Возникает при неправильном задании вложенности.
3	Duplicate Error	Ошибка в наборе имени. Объект с таким именем уже существует.
4	Type Error	Ошибка типа. Указан неверный тип объекта, неподходящий данному классу.

Табл. 3.4. Коды ошибок для оператора *Create*.

*Оператор Delete.* Оператор *Delete* позволяет удалять объекты, микроситуации, макроситуации и переменные различного типа. Синтаксис позволяет удалять вложенные объекты и осуществлять каскадное удаление.

*Delete* <[*Object*]/> [*Name*] <*Cascade*> <*All*>

*Return* [*Name*] <not> *deleted*  
<*Error* [*Type of error*]>

Ключевое слово *Cascade* используется для удаления вложенных объектов. При попытке удаления вложенного объекта без этого пара-

<sup>1</sup> Коды ошибок 0, 1, 2 и их значения являются общими для всех операторов и далее в таблицах не повторяются.

метра возникнет ошибка. Конструкция *All* позволяет удалить объекты заданного типа во всех объектах СМ.

При включении опции *All* несколько изменяется вывод сообщений. Указывается количество удаленных объектов (*Summary*) и список имен удаленных объектов (*List of objects*).

*Return [Summary] objects deleted [List of objects]*

Код ошибки	Название ошибки	Описание
3	Error Child	Ошибка вложенности. Объект содержит вложенные объекты. Необходимо использовать конструкцию <i>Cascade</i> .

Табл. 3.5. Коды ошибок для оператора *Delete*.

*Оператор Set*. Оператор *Set* позволяет устанавливать значения для всех типов объектов.

*Set <[Object]/> [Name] [Value] <Rew(rite)>*

*Return [Name] <not> set*

*<Error [Type of error]>*

Код ошибки	Название ошибки	Описание
3	Type Error	Ошибка типов. Несоответствие введенного значения типу объекта.
4	Write Error	Ошибка записи. Объект уже имеет значение. Необходимо использовать конструкцию <i>Rew(rite)</i> .

Табл. 3.6. Коды ошибок для оператора *Set*.

*Оператор WhatIs*. Оператор *WhatIs* позволяет получить информации о классе, типе и значении объекта, микро и макроситуации, переменной.

*WhatIs <[Object]/> [Name]*

*Return [Class, Type] [Name] is <[Value]>*

*<Error [Type of error]>*

Если искомый элемент ситуационной модели имеет тип, то сообщается и класс объекта, и его тип (*Class, Type*). Значения (*Value*) указываются для переменных, ситуаций и объектов, не имеющих вложений. В случае вложенности вместо значения будет выдано слово *Container*. Если значение не определено, то будет выдано слово *Unknown*. Если объекта не существует, то будет выдано сообщение *Not Available*.

*Оператор ListOf.* Оператор *ListOf* предназначен для просмотра вложенных объектов и переменных.

*ListOf* <[Object]/> [Name]

Return [Type] [Name of Object1] is <[Value]>

[Type] [Name of Object2] is <[Value]>

...

[Type] [Name of Objectn] is <[Value]>

<Error [Type of error]>

Значения (*Value*) указываются для переменных и объектов, не имеющих вложений. В случае вложенности вместо значения будет выдано слово *Container*. Если значение не определено, то будет выдано слово *Unknown*. Если объекта-родителя не существует, то будет выдано сообщение *Not Available*.

Код ошибки	Название ошибки	Описание
3	Error Child	Ошибка вложенности. Объект не является вложенным.

Табл. 3.7. Коды ошибок для оператора *ListOf*.

*Оператор Select.* Оператор *Select* предназначен для поиска объектов СМ, удовлетворяющим условиям запроса.

*Select* <{*Bes(t)*, *Wor(st)*, *Ran(dom)*}>

[List of Type Objects]

<Quantity [Value]>

<Where [Object]/[Name] [Compare Sign][Value] ...>

Return [Name of Object1]

[Name of Object2]

...

[Name of Objectn]

<Error [Type of error]>

Ключевые слова *Best*, *Worst*, *Random* предназначены для выбора лучших, худших и случайных вариантов соответственно. Если количество выбираемых объектов ограничено (*Quantity*), то в результирующую выборку попадают объекты, для которых заданное условие наиболее соответствует выбранному варианту. Если осуществляется выборка всех объектов, то ключевые слова оказывают влияние на упорядочивание.

В конструкции *List of Type Objects* можно задать перечень типов объектов, классов переменных или ситуаций. Перечисление осуществляется через запятую.

Предикат поиска *Where* позволяет наложить ограничения на результирующую выборку с помощью знаков равенства и сравнения (*Compare Sign*). Может накладываться несколько условий, которые должны соединяться между собой знаками *&*, *|* и *not*.

В результате выполнения запроса создается список объектов. Если список пустой, т.е. нет объектов, удовлетворяющих условиям, то выдается сообщение *List is Empty*.

Код ошибки	Название ошибки	Описание
3	Type Error	Ошибка типов. Несоответствие введенного значения типу объекта.

Табл.3.8. Коды ошибок для оператора *Select*.

### 3.3. Исследование эффективности ситуационного моделирования

Данный параграф посвящен исследованию эффективности ситуационного моделирования. Рассматриваются преимущества данного подхода для решения задач поиска и оценки результатов в системах, имеющих большую размерность. Чаще всего эффективность можно определить исходя из затрачиваемых ресурсов, производительности оборудования и труда.

Основные задачи ситуационного моделирования направлены на принятие решение субъектом (человеком), поэтому наиболее важной характеристикой является производительность труда, эффективность человеческой деятельности. Применительно к программам, к средствам автоматизации можно говорить об эффективности алгоритмов.

Таким образом, основным ресурсом и главным показателем является время выполнения той или иной задачи в рамках ситуационной модели. Для оценки алгоритмов будем использовать также понятие *порядка времени выполнения* программы.

#### Время выполнения программ

Ситуационное моделирование предназначено для работы с системами, имеющими большую размерность. Под размерностью мы понимаем количество исходных данных. В ситуационной модели используется множество различных типов данных, начиная с простых (числовых и текстовых) и заканчивая сложными иерархическими объектами (микро и макроситуациями).

Для оценки мощности данных введем понятие элементарного типа данных *Dt*. Будем считать, что каждый другой тип данных можно свести к количественному выражению в *Dt*:

$$Dt_{\text{макроситуация}} = k * Dt,$$

где  $Dt_{\text{макроситуация}}$  — тип данных для представления макроситуаций;  
 $k$  — коэффициент соответствия.

Каждому типу данных должен соответствовать свой коэффициент, поэтому можно говорить о наборе или векторе коэффициентов соотношений для типов данных.

где  $m$  — количество типов данных

Чтобы посчитать размерность системы  $N$  необходимо учесть помимо коэффициентов соответствия количество объектов заданного типа  $r_i$ .

$$N = k_1 * r_1 * Dt + k_2 * r_2 * Dt + \dots + k_m * r_m * Dt = Dt (k_1 * r_1 + k_2 * r_2 + \dots + k_m * r_m)$$

$$N = Dt * \sum_1^m k_i * r_i$$

$$\vec{K} = \{K_1, K_2, \dots, K_m\}$$

где  $N$  — размерность системы;

$r_i$  — количество объектов заданного типа.

Для упрощения вычислений осуществим нормализацию формулы, т.е. будем считать, что  $Dt = 1$ , а за показатель размерности системы примем параметр  $n$ . Он соответствует количеству данных, выраженных в элементарных объектах.

$$n = \frac{N}{Dt} = \sum_1^m k_i * r_i$$

$$\text{При } Dt = 1 \Rightarrow n = N$$

Пусть время выполнения программы (алгоритма)  $T$  функционально зависит от количества исходных данных, т.е.:

$$T = f(n) \quad \Leftrightarrow T(n)$$

Для описания скорости роста функции воспользуемся  $O$ -символикой [Ахо, 2000]. Функция имеет порядок  $O(n^s)$ , если существуют положительные константы  $c$  и  $n_0$ , такие что:

$$\forall n \geq n_0 \quad T(n) \leq c * n^s$$

Функция роста не обязательно должна быть степенной, поэтому будем считать, что если  $T(n)$  имеет степень роста  $O(f(n))$ , то это верхняя граница скорости роста  $T(n)$ . Нижнюю границу будем обозначать  $\Omega(g(n))$ .

### Эффективность ситуационного подхода

Рассмотрим эффективность ситуационного моделирования на примере реализации операции поиска. Для этого воспользуемся конструк-

цией *Select*, используемой в ситуационном языке. Составим укрупненный алгоритм выполнения запросов данного типа (рис. 3.5.):

*Анализ языковой конструкции и поиск ошибок.* Будем считать, что время разбора языковых инструкций ( $T_{sy}$ ) намного меньше, чем время выполнения запроса ( $T_{Qu}$ ).

$$T_{sy} \ll T_{Qu}$$

*Ввод исходных данных для поиска.* После разбора конструкции необходимо определить условия запроса (правила сортировки, количество объектов, условия отбора и т.д.). При расчетах не будем учитывать время, затрачиваемое на чтение/запись значений переменных  $T_{rw}$  (исходных данных), так как оно не зависит от размерности системы ( $n$ ). Будем считать, что это время является одинаковым для ситуационного и других подходов.

$$T_{rw} = const \neq Func(n)$$

*Обработка всех объектов СМ.* Этот этап представляет собой цикл, обрабатывающий все объекты СМ. Количество итераций зависит от используемого подхода. Если не использовать ситуационное представление, то цикл будет выполняться  $n$  раз.

*Проверка на соответствие типу искомых объектов.* На этом этапе организуется цикл проверки соответствия объекта заданным типам. Размерность цикла  $a$  зависит от количества различных типов объектов.

*Проверка на соответствие заданных условий.* На этом этапе организуется цикл проверки условий. Размерность цикла  $b$  зависит от количества условий. Будем считать, что условия простые и не имеют вложенностей.

*Сортировка полученной выборки.* Направление сортировки данных (по убыванию, по возрастанию, случайная выборка) определяются из условий запроса. Будем считать, что сортировка осуществляется по одному из алгоритмов с наименьшей степенью роста.

$$T_{sort} = S * \text{Log} S,$$

где  $T_{sort}$  — время сортировки;

$S$  — количество элементов в результирующей выборке.

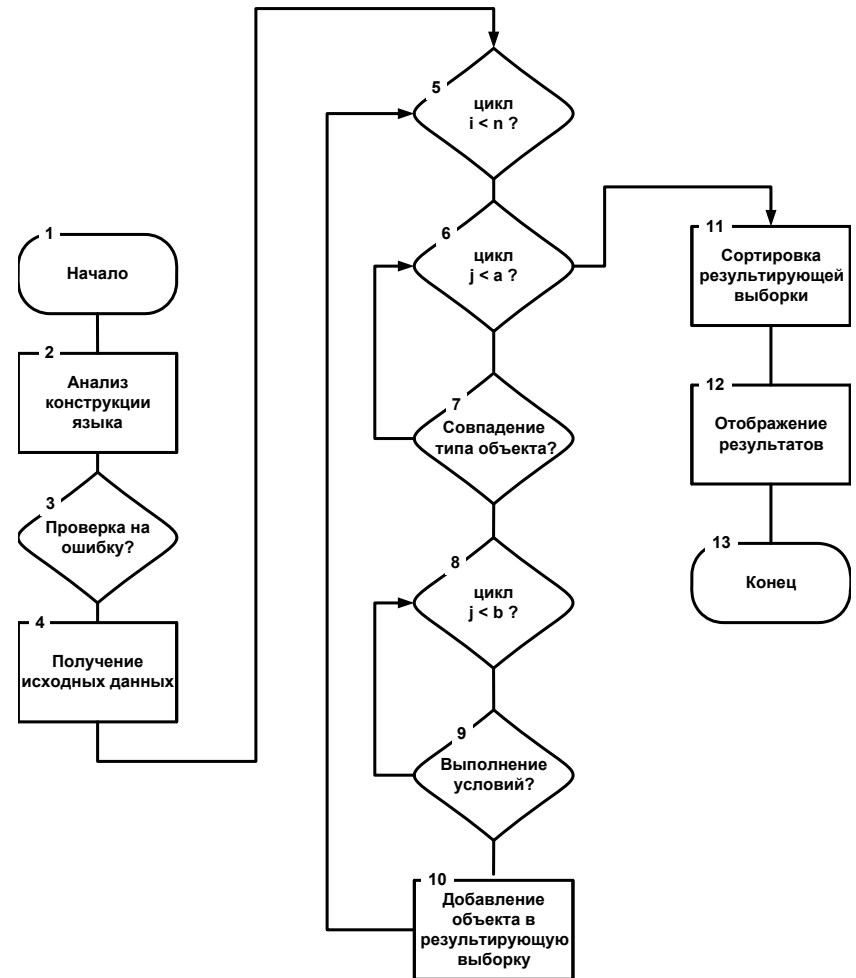


Рис. 3.5. Стандартный алгоритм поиска.

Отображение заданного количества объектов. Отделение операции отображения от операции ввода/вывода связано с тем, что различное представление информации требует разного времени ее анализа (просмотра)  $T_{out}$ . Таким образом, суммарное время выполнения алгоритма поиска будет выглядеть следующим образом:

$$T_{select} = n*(a + b) + S*LogS + T_{out}$$

Выразим суммарное время через размерность системы и соответствующие константы:

$$T_{select} = c_1 * n + c_2 * n * \text{Log}(c_2 * n) + c_3 * n$$

$$\text{где } c_1 = a + b; \quad c_2 * n = S; \quad c_3 * n = T_{out}$$

$$T_{select} = c_4 * n + c_2 * n * \text{Log}(c_2 * n) = n * (c_4 + c_2 * \text{Log}(c_2 * n)) = \\ = n * (1 + c_5 * \text{Log}(c_2 * n))$$

$$\text{где } c_4 = c_1 + c_3; \quad c_5 = c_2 / c_4;$$

Если считать, что результирующая выборка значительно меньше общего количества элементов в СМ, то временем на выполнение сортировки и отображения результатов можно пренебречь.

$$\text{При } (S \ll n) \ \& \ (T_{out} \ll n) \Rightarrow T_{select} = n * (a + b) = c_1 * n$$

Таким образом, функция имеет порядок роста  $O(n)$ , а константа  $c_1$  определяет крутизну роста функции.

Используя ситуационное представление информации, можно значительно сократить время поиска. Это преимущество появляется за счет использования иерархического поиска. Имея категориальные характеристики (состояния, ситуации и др.), можно отсеять неподходящие варианты на верхнем (категориальном) уровне. Это утверждение достаточно очевидно. В качестве ближайшего аналога можно использовать понятия В-деревьев, которые применяются для построения индексов к базам данных и в различных поисковых системах.

Представим, что имеется три уровня абстракции: характеристики объекта, микроситуация (состояние) объекта и ситуация, в которой находится группа объектов. Пусть  $N$  — общее количество характеристик (размерность системы), тогда чтобы найти значение нужной характеристики необходимо совершить  $N$  сравнений.

Предположим теперь, что на уровне ситуаций можно отсеять все объекты и их характеристики, которые находятся в очень сложной финансовой ситуации. Тогда количество переборov можно сократить.

$$K = N - P_1,$$

где  $K$  — количество переборov (сравнений);

$P_1$  — количество отбрасываемых характеристик.

Далее предположим, что на уровне микроситуаций (состояний) можно отбросить все объекты, которые находятся в неработоспособном состоянии или сильно перегружены.

$$K = (N - P_1) - P_2,$$

где  $P_2$  — количество отбрасываемых характеристик на уровне микроситуаций.



Очевидно, что чем больше уровней иерархии, тем меньше остается переборов на самом нижнем уровне.

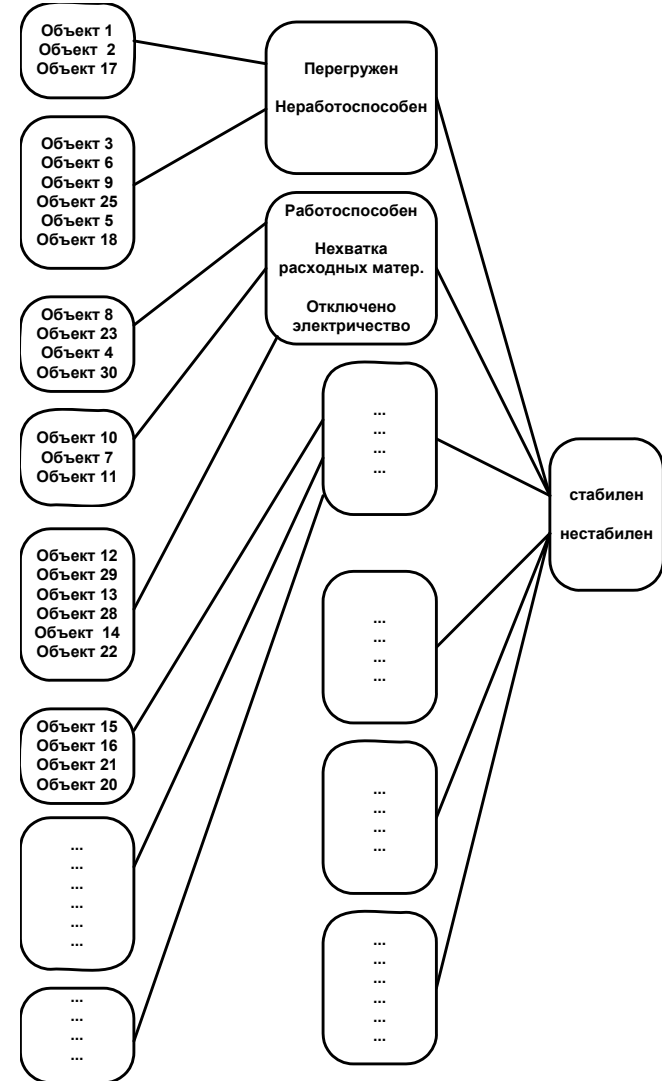


Рис. 3.6. Ситуационное дерево поиска.

Сравним теперь это представление с В-деревом. Каждый узел, кроме корневого, имеет не менее  $m/2$  и не более  $m$  потомков. Корневой узел имеет как минимум два потомка. Пример ситуационного дерева поиска приведен на рис. 3.6.

Основное отличие применения индексов от создания ситуационной модели состоит в том, что разбиение по уровням осуществляется не по содержимому отдельных ячеек (характеристик), а по специально введенным категориальным понятиям — ситуациям.

Некоторые ситуации невозможно (или очень) сложно определить формально (с помощью ЭВМ), поэтому иногда может потребоваться субъективное вмешательство.

Итак, ситуационная модель и ситуационный подход в логике предикатов позволяют реализовать категориальный аппарат и создать сложную иерархическую модель. Количество уровней иерархии может определяться по формуле, используемой для В-деревьев [Кнут, 2000]

$$L = 1 + \text{Log}_{(m/2)}[(N+1)/2],$$

где  $L$  — количество уровней;

$m$  — максимальное число характеристик, приходящихся на одну категорию;

$N$  — общее число элементов в системе.

Для сравнения степени роста функции при таком подходе примем, что максимальное число характеристик, приходящихся на одну категорию, не превышает двух десятков:

$$m = 20 \Rightarrow L = 1 + \text{Lg}(N/2) \Rightarrow$$

$$T_{\text{select}} = m * L * t_{\text{op}} = 20 * t_{\text{op}} * (1 + \text{Lg}(N/2)) = 20 * t_{\text{op}} + 20 \text{Lg}(n/2) * t_{\text{op}},$$

где  $t_{\text{op}}$  — время на обработку одной характеристики.

Необходимо отметить, что нельзя пренебречь значением  $20 * t_{\text{op}}$ , так как оно соизмеримо со значением  $20 * t_{\text{op}} * \text{Lg}(N/2)$ . Это наглядно иллюстрируется на рис. 3.8. На рис. 3.7. показаны зависимость эффективности для различных значений времени обработки одной операции.

Таким образом, функция для ситуационного подхода имеет порядок роста  $O(\text{Lg}(n/2))$ . Важно отметить, что при предложенной организации системы требуются существенные временные затраты на создание новых объектов, так как необходимо переформировать дерево. Аналогично следует поступать при значительных изменениях.

В случае добавления узла в В-дереве может потребоваться разделить до  $l$  узлов, однако, среднее количество таких разделений существенно

$$p \leq 1 + \frac{N-1}{[m/2]-1} \quad \text{меньше.}$$

где  $p$  — количество разделений.

Недостатком оценки ситуационной модели с использованием в качестве аналога структуры В-дерева является то, что в СМ количество уровней определяется не из целесообразности поиска (уменьшения времени поиска), а из семантического принципа.

$$T_{sel1}(n) := 20 \cdot \log\left(\frac{n+1}{2}\right) \quad T_{sel2}(n) := 20 + 20 \cdot \log\left(\frac{n+1}{2}\right)$$

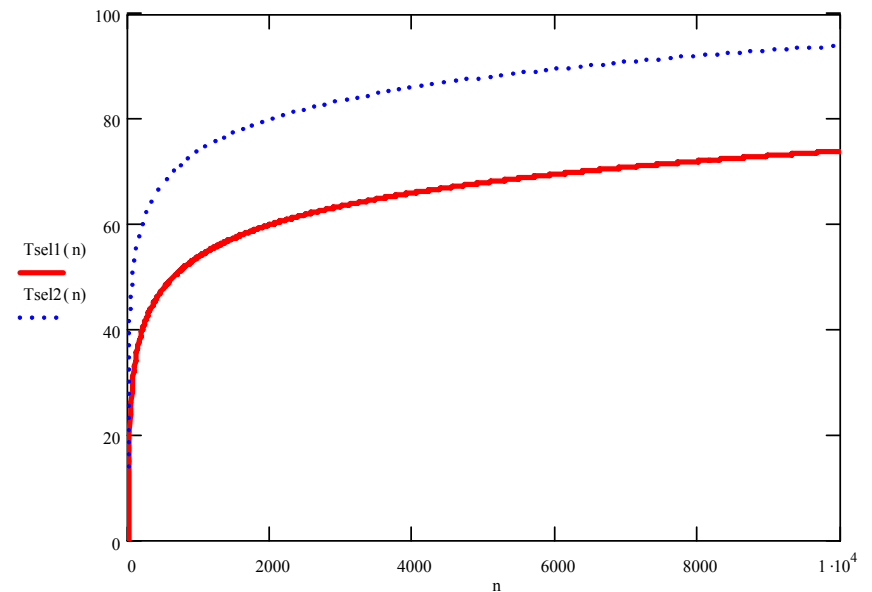


Рис. 3.7. Эффективность ситуационного подхода.

Например, при описании полиграфического рынка России общее число фирм  $m$  может достигнуть значения 3000, что не позволяет соблюсти ограничение в 20 характеристик на всех уровнях. Кроме того, два разных объекта (например, макроситуации) могут иметь общие характеристики на более низких уровнях, что недопустимо в В-дереве.

Исходя из этих замечаний, следует отметить, что логарифмическая зависимость времени поиска в ситуационных системах является заниженной оценкой и требует более детального изучения. Однако, на основании полученных результатов, можно утверждать, что ситуационный подход показывает большую эффективность.

$$t1 := 1 \qquad t2 := 2$$

$$Tsel3(n) := 20 \cdot t1 \cdot \left( 1 + \log\left(\frac{n+1}{2}\right) \right) \qquad Tsel4(n) := 20 \cdot t2 \cdot \left( 1 + \log\left(\frac{n+1}{2}\right) \right)$$

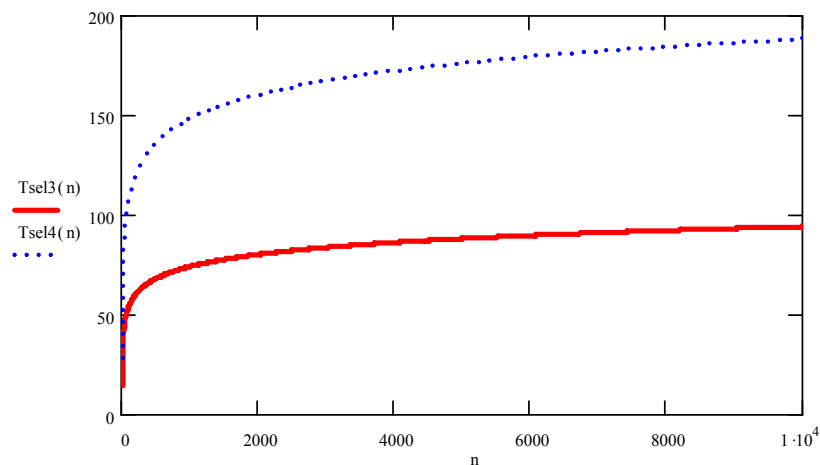


Рис. 3.8. Эффективность ситуационного подхода для различного времени выполнения одной операции.

## Выводы

В главе рассмотрен третий этап методики интеграции систем. Он подразумевает создание ситуационной модели (СМ) и языка взаимодействия с ней. В настоящее время не существует универсальных систем ситуационного моделирования, поэтому их необходимо разрабатывать полностью. Теоретически использование только ситуационной системы может оказаться достаточным для решения поставленных задач. На практике ситуационные системы (СЦ) включают в себя различные программные компоненты для поддержки принятия решения, анализа и прогнозирования.

При интеграции ССМ с другими системами следует выбирать такую модель представления знаний и такой язык взаимодействия, которые обеспечат максимальную совместимость. При разработке всех компонент СЦ следует выбрать единую модель представления знаний, пример которой приводится в следующей главе.

В первой части главы представлен пример проектирования ситуационной модели рынка полиграфических услуг (СМ РПУ). В параграфе кратко представлена структура рынка, возможные срезы модели и варианты декомпозиции, а также четыре основные функции проектируемой системы: поиска информации, посредника (контактная), ведения заказа, обеспечения производства.

В качестве математического аппарата представления ситуационной модели выбирается логика предикатов первого порядка с использованием ситуационных расширений. Данный ЯПЗ широко используется в ЭС.

В рамках основных функций определяется более 200 различных классов, типов объектов, переменных и относящихся к ним микроситуаций. Для некоторых объектов системы приводится описание пространства состояний (возможные микроситуации). Особо указываются категории объектов. Выделяются около 10 групп макрообъектов. В последней части главы описывается ситуационный язык, предназначенный для работы с СМ. Язык носит схематичный характер и предусматривает только простейшие операции создания, изменения, поиска и удаления объектов СМ.

В третьей части главы с помощью аналитических методов исследуется эффективность ситуационного подхода на примере выполнения оператора SELECT. Зависимость времени выполнения запроса от размерности системы при ситуационном подходе имеет вид логарифмической кривой. Однако расчеты носят приближенный характер, так как рассматривают упрощенный пример.