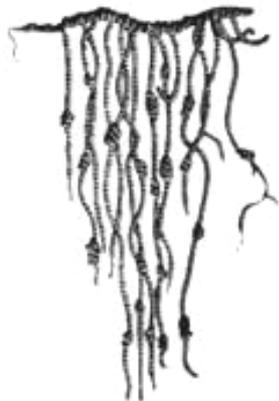


## Лекция 3-4. Идеальный конечный результат. Свертывание. Функции

### 3.1. Идеальный конечный результат. Закон стремления к идеальности



**Рис. 2.8. Узелковое письмо**

Один из фундаментальных законов ТРИЗ говорит о том, что технические системы стремятся в своем развитии (в филогенезе, т. е. в историческом развитии) к идеальному конечному результату: системы нет, а ее функция выполняется. Городские стены, например, в древности, да и в средние века играли очень важную защитную функцию. Со временем эта функция городов была передана государству, армии. Защитная функция городской стены просто отпала.

Это относится не только к техническим, но и к любым системам, созданным для реализации той или иной функции. Во времена завоевания Америки инки, например, для передачи информации использовали узелковое письмо, которое передавали специально обученные гонцы, для которых была создана специальная сеть дорог (рис. 2.8). Традиционное письмо с бумагой и конвертами теперь тоже отошло в прошлое. Имеется прогноз, что и электронные письма тоже будут вытеснены, например, общением в социальных сетях. Затраты для создания и передачи сообщений постоянно падают, а их полезные функции увеличиваются. Есть, правда и вредные функции, например, когда сообщение перехватывается недругами.

Закон стремления к идеальному конечному результату (ИКР) определяет основное направление в развитии систем. ИКР можно описать формулой:

$$ИКР = \frac{\sum F_{\text{полезные}}}{\sum \text{Затрат} + \sum F_{\text{вредные}}}$$

где **ИКР** – идеальный конечный результат для системы (**должен увеличиваться** ↑);

$\sum F_{\text{полезные}}$  – сумма полезных функций системы, например, функциональность программного продукта (**должна увеличиваться** ↑);

$\sum \text{Затрат}$  – сумма затрат для выполнения всех функций, например, времени и других ресурсов, затраты программистов и пользователей, при разработке, отладке, эксплуатации и других этапах жизненного цикла программы, при продвижении, продажах и доставке продукта и т. д. (**должна уменьшаться** ↓);

$\Sigma F_{\text{вредные}}$  – сумма вредных функций системы, например, возможности для вредоносных программ, незащищенность от различных режимов эксплуатации, рискованные возможности, предоставленные пользователю и т. д. (должна уменьшаться ↓).

**Идеальная система** – это система, которой нет, а ее функция выполняется.

Логика закона стремления к идеальности довольно понятная: прибыли должно быть больше, а затрат меньше. В ТРИЗ эта логика, этот закон превращены в инструмент для решения изобретательских задач. Для этого имеются специальные формулы определения ИКР для той или иной проблемной ситуации. Можно выделить три типа ИКР.

**Функциональный ИКР:** Элемент из системы (описать) САМ должен (описать действие), чтобы (описать ) при ограничениях (описать).

**Ресурсный ИКР:** X-элемент (из ресурсов системы), абсолютно не усложняя систему и не вызывая вредных явлений САМ (указать требуемое действие) в течение оперативного времени ОВ (указать) в пределах оперативной зоны ОЗ (указать), сохраняя (указать полезное действие или ограничения).

**ИКР свойств:** Оперативная зона (указать) в течение оперативного времени (указать) должна САМА обеспечивать (указать противоположные свойства).

Для примера сформулируем ИКР для задачи 3 о том, как мальчику открыть закрытую дверь.

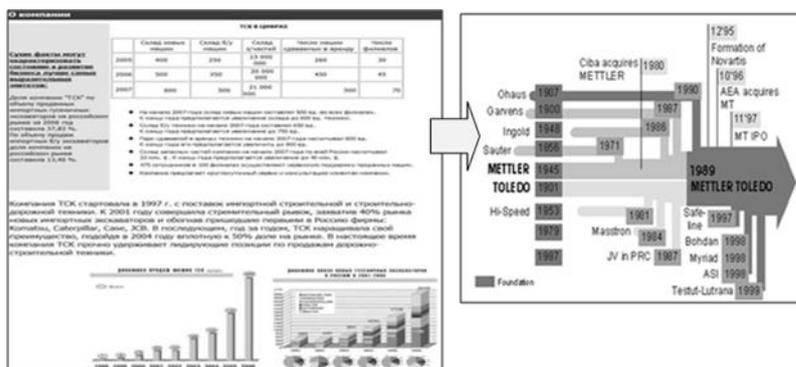
**Функциональный ИКР:** Сестра (*других подходящих элементов в этой системе просто нет*) САМА должна открыть дверь, не причинив никому вреда и не создавая опасности.

**Ресурсный ИКР:** X-элемент (скорее всего сестра), абсолютно не усложняя систему и не вызывая вредных явлений САМ открывает в течение оперативного времени переговоров брата с младшей сестрой в пределах оперативной зоны пространства рядом с дверью, сохраняя безопасность и не причинив никому вреда.

**ИКР свойств:** Сестра в течение оперативного времени переговоров брата с младшей сестрой должна САМА обладать желанием открыть дверь и желанием не открывать дверь одновременно.

Восьмилетний мальчик из этой задачи уже что-то знал про сказочные, идеальные решения. Он придвинул к двери небольшой ящик и сказал сестре: я тебя закрыл. Четырехлетняя Ксюша САМА открыла дверь, конфликтная ситуация превратилась в смешную.

Попробуем теперь решить задачу 2 о визуализации информации на сайте при помощи формулировок ИКР. Вспомним уже сформулированные противоречия и рекомендации по применению приемов, о которых мы уже



**Рис. 2.9. К задаче 2 о визуализации информации о компании: компромиссные решения не помогают**

писали выше.

Противоречие требований: ЕСЛИ предоставить информацию о компании во всех деталях, ТО она будет охватывать всю возможную информацию, НО при этом ею будет пользоваться неудобно, она будет плохо восприниматься.

Используя укороченную таблицу применения приемов, мы получили рекомендации применить комплекс приемов: динамичности, отброса и регенерации частей, наоборот и принцип обратной связи (приемы 15, 34, 13, 23).

Мы также сформулировали раньше рекомендации на основе принципов разрешения противоречий свойств: в какое-то время и в каком-то месте экрана информации должно быть много, а в какой-то – мало, система в целом должна иметь много информации, а ее доступные элементы, подсистемы должны иметь мало информации.

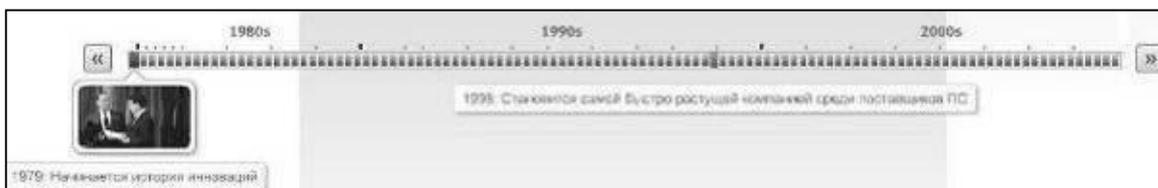
Теперь перейдем к формулировке ИКР.

**Функциональный ИКР:** Пользователь САМ должен отразить более детальную информацию, сохраняя возможность видеть обобщенную информацию.

**ИКР свойств:** Курсор пользователя в течение просмотра сайта должен САМ обеспечить подробность и обобщенность (краткость) информации.

После всех этих подсказок возможное решение становится практически очевидно (рис. 2.9). Один из возможных вариантов решения можно посмотреть на сайте <http://russia.emc.com>. Передвигая курсор вдоль линии дат, пользователь получает одновременно краткое описание и краткий визуальный ряд (рис. 2.10). При необходимости можно перейти к более детальной информации. Любопытно, что, используя подходы ТРИЗ, это решение может быть улучшено.

Если из условий задачи известно, какой должна быть готовая система, и задача сводится к определению способа получения этой системы, то можно использовать метод «шаг назад от ИКР». Для этого представляют или описывают готовую систему, которую необходимо получить в идеальном случае, а затем вносят в эту идеальную картину минимальное демонтирующее изменение.



**Рис. 2.10.** К задаче 2. Пример визуализации с сайта <http://russia.emc.com>

Например, мы представили, что идеальным образом истории компании является линия (линия времени). Демонтирующим изменением этой идеальной картинки является добавление информации о том, что же все же было в тот или иной год в компании. Как сделать, чтобы эта дополнительная информация была, и чтобы ее не было? Динамично возникающая информация о компании, относящаяся к нужному году – одно из возможных решений.

### **3.2. Линия развития «моно-би-поли-свертывание»**

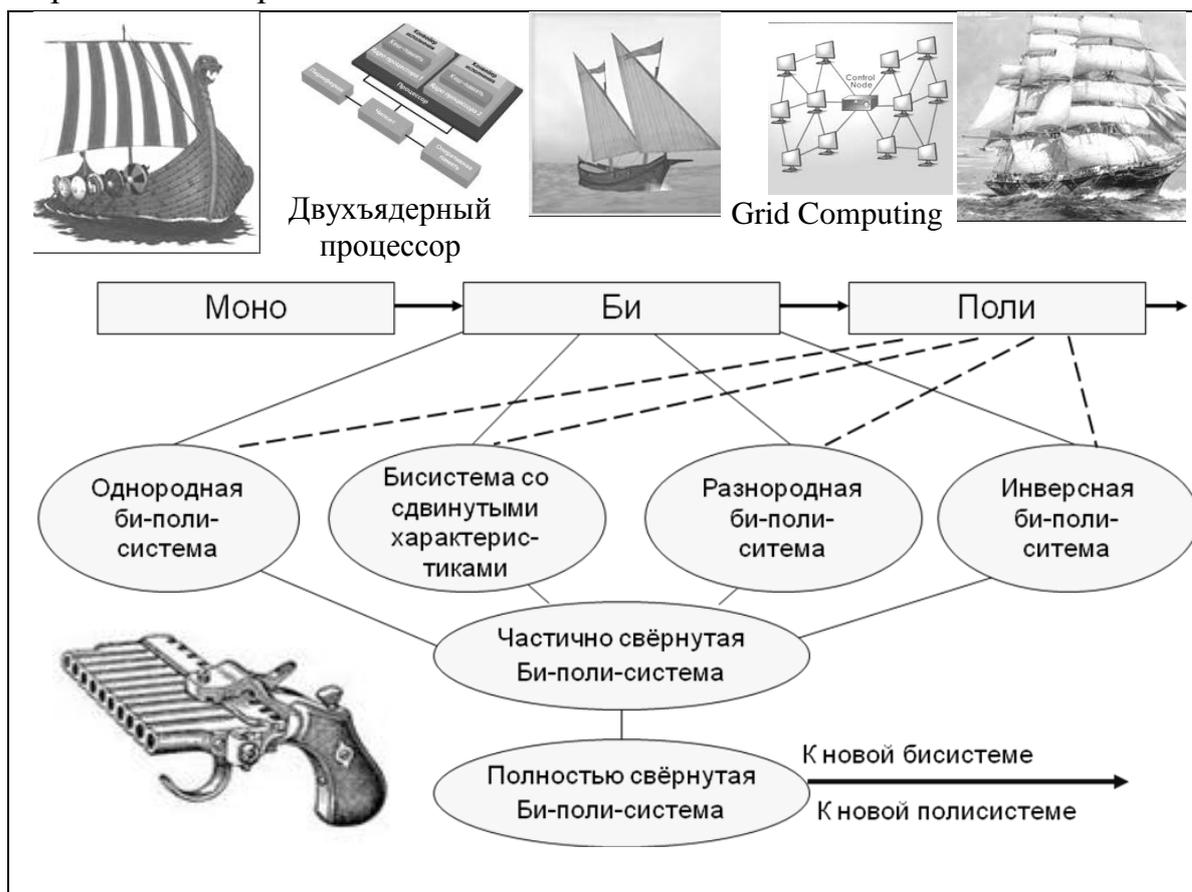
В ТРИЗ имеется очень эффективный инструмент приближения к ИКР. Речь идет о свертывании: система, состоящая из многих элементов, переходит к системе, состоящей из гораздо меньшего или отсутствующего числа элементов (системы нет, а ее функция выполняется). Прежде чем



**Рис. 2.11. Двойной планшетник**

перейти к этому инструменту развития систем, необходимо познакомиться с линией развития, которая получила в ТРИЗ название «МОНО – БИ – ПОЛИ»: МОНО-система со временем в развитии становится БИ-системой, а потом и ПОЛИ-системой. При этом у нее улучшаются важные для этой системы характеристики, но одновременно растут и затраты (рис. 2.11). Затем происходит свертывание: те же характеристики и функции удается получить при помощи другой МОНО-системы. Например, чайные клиперы с большим числом парусов вынуждены были

уступить обычным парходам, передвижение которых вообще не зависит от направления ветра.



**Рис. 2.12. Иллюстрации линии «моно-би-поли-свертывание»**

Эти же процессы наблюдаются в развитии компьютеров (один компьютер, два объединенных компьютера, сеть компьютеров, один компьютер по мощности выше сети старых компьютеров), процессоров (многоядерные процессоры), программ и программных комплексов. Например, редакторы текстов сейчас объединяют в себе возможности программ для рисования, контроля орфографии, перевода текстов на разные языки, электронных таблиц и т. д. От программ с одним окном перешли к многооконным системам, от программ с одним пользователем – к многопользовательским системам. Компьютер и телефон «объединились» в один планшетник, а теперь ожидают появления сдвоенных планшетников (рис. 2.11). Недавно возникла идея гибридных компьютеров: в одном узле объединены процессоры общего назначения и графические платы. Разнородность объединяемых систем, как правило, повышает эффективность объединения. Например, объединение пользовательских ресурсов и рекламы, деловых программ с игровыми и обучающими программами и т. д.

Закон перехода в надсистему содержит четыре тенденции:

- параметры объединяемых систем все больше отличаются от параметров технической системы;

- основные функции объединяемых систем все больше отличаются от функций технической системы;
- интеграция между технической системой и объединяемыми с нею системами становится глубже;
- увеличивается число объединяемых систем.



**Рис. 2.13. Многоствольный пистолет**

Свертывание позволяет значительно повысить эффективность объединения двух систем. Например, от винтовки перешли к применению двух винтовок. Затем появились двухстволки: на два ствола только один приклад. Одним прикладом стало меньше. Далее появились многоствольное оружие (рис. 2.13), затем опять свертывание с переходом к автоматическому оружию: ручка

(приклад) одна, ствол опять один, а функции многоствольного ружья или пистолета осталась. Получив опять МОНО-систему в виде автоматического оружия, цикл переходов повторяется: двухствольное автоматическое оружие, многоствольные автоматы и т.д.

Свертывание – это один из инструментов повышения идеальности систем: затраты и вредные функции снижаются, а функциональность сохраняется или повышается.

В ТРИЗ разработаны правила свертывания (удаления) элементов системы.

Правило 1. Элемент можно удалить, если нет объекта функции.

Правило 2. Элемент можно удалить, если функцию выполняет сам объект функции.

Правило 3. Элемент можно удалить, если функцию выполняют оставшиеся элементы системы.

Приведем самый простой пример – развитие зубной щетки с футляром (рис. 2.14).

<p>Свертывание корпуса футляра и ручки щетки</p>	<p>Свертывание ручки щетки</p>	<p>Свертывание ручки щетки</p>

## Рис. 2. 14. Свертывание в развитии зубной щетки с футляром

Эта система состоит с двух основных элементов: футляр и зубная щетка. Футляр состоит из двух элементов: корпуса и крышки. Зубная щетка состоит из щетины, подложки щетины и ручки. На фотографиях приведены примеры нескольких вариантов свертывания.

Эту методику свертывания можно применять и для программных продуктов.

**Задача 7.** Задача о программе вычисления произвольного полинома. В программе, предназначенной для вычисления значения произвольного полинома, имеется текстовое поле для ввода. Введенная строка затем проверяется на правильность (действительно ли пользователь ввел полином, а не произвольную строку), а затем распознается (создается модель полинома для его последующего вычисления). Однако написание такой программы вызывает трудности, возникает большое количество ошибок. Кроме того, усложняется структура выполнения программы. Необходимо упростить программу и повысить ее надежность.

В предложенной формулировке в явном виде отсутствует противоречие требований. Это не позволяет применять методы ТРИЗ, направленные на решение противоречий требований.

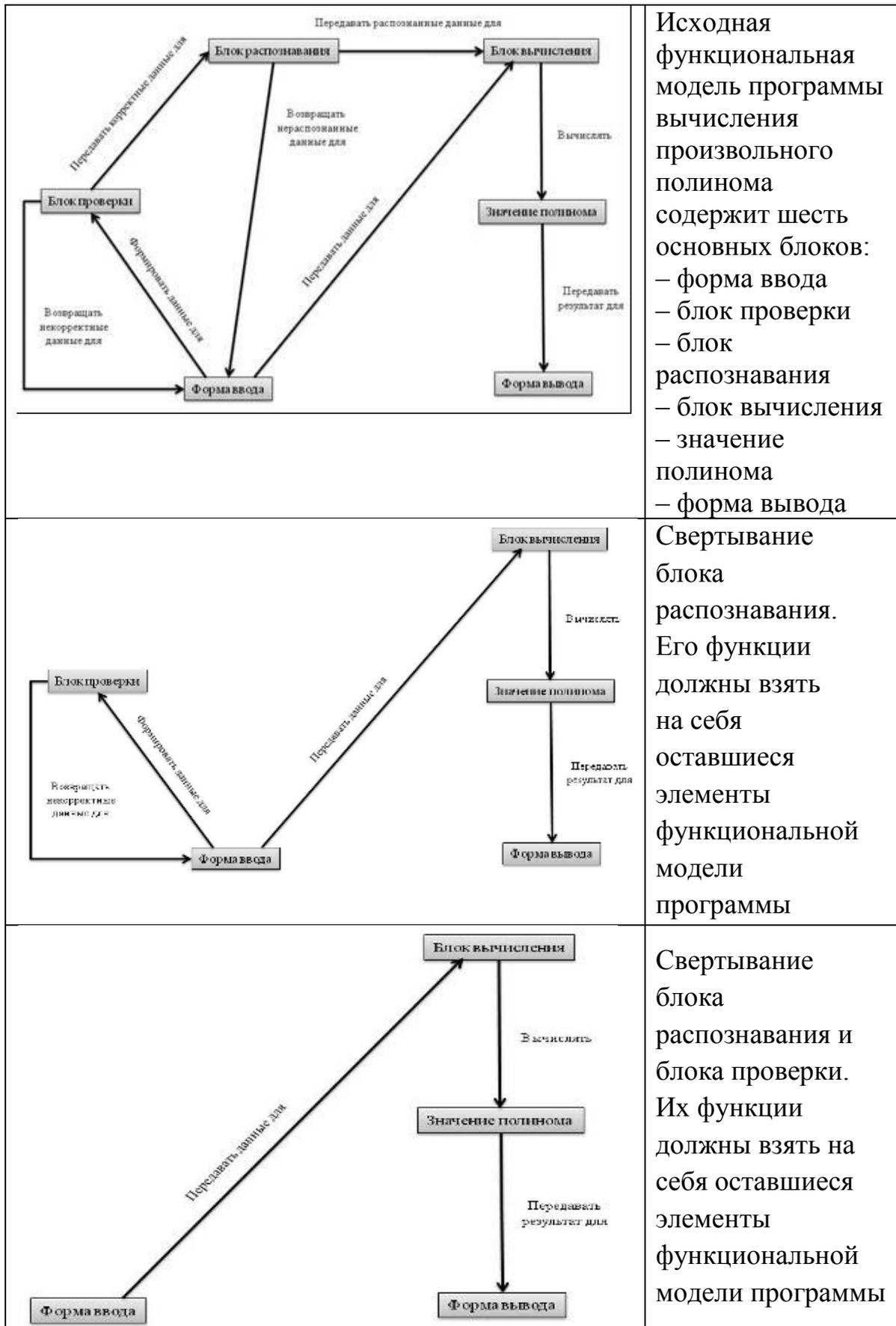
Для решения **задачи 7** воспользуемся правилами свертывания элементов системы. Для этого вначале составим функциональную схему такой программы и постараемся произвести свертывание отдельных элементов функциональной модели программы, перенося их функции на оставшиеся элементы.

Уточним формулировку **задачи 7** с учетом предложений по частичному свертыванию элементов функциональной модели программы.

Сформулируем задачу свертывания: вычислять значение полинома с помощью формы ввода и блока вычисления. При этом либо форма ввода, либо блок вычисления должны выполнять следующие функции: распознавать данные для блока вычисления, возвращать нераспознанные или некорректные данные для формы ввода (рис. 2.15).

Очевидно, эти функции должны выполняться формой ввода.

Задача свертывания в виде противоречия: форма ввода не должна накладывать ограничения, чтобы обеспечить возможность задания произвольного полинома, и должна иметь жесткие ограничения, чтобы проверять корректность введенной строки.



**Рис. 2.15.** К задаче 7: пример свертывания функциональной модели программы

Такую изобретательскую задачу вполне можно решить методами ТРИЗ. К этой задаче мы вернемся позже.

#### Лекция 4.

### 3.3. Объединение альтернативных систем. Перенос свойств

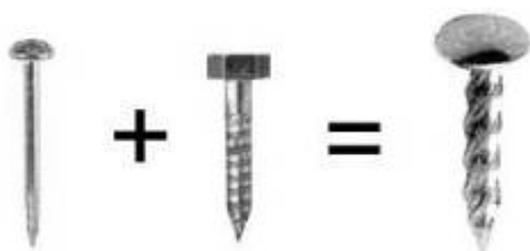
Одна из разновидностей перехода к БИ-системе с одновременным свертыванием – это методика объединения альтернативных систем или перенос свойств одной системы на другую. Это довольно простая и эффективная методика развития систем, которую можно применять и для развития программного обеспечения, и для развития информационных и других технологий.

Идея переноса свойств:

Перенос свойств открывает новые ресурсы для улучшения системы без дополнительных вложений и затрат. В предлагаемом подходе одновременно анализируются исходная система и альтернативная система с точки зрения свойств, желательных для исходной системы

Перенос свойств не требует дополнительных доказательств того, что нужные свойства могут быть получены, так как они уже есть у альтернативной системы

	Гвоздь	Шуруп
Легкость забивания	+	-
Эффективность удержания	-	+



**Рис. 2.16. Объединение свойств гвоздя и шурупа**

Базовая техническая система – система, на которую переносятся свойства альтернативной системы.

В качестве примера рассмотрим две альтернативные системы: гвоздь и шуруп. У них одна и та же функция: скрепить два объекта между собой. При этом у гвоздя преимущество в том, что он быстро забивается, но держит он

Перенос свойств – это аналитический инструмент для совершенствования Технической Системы (ТС) путем передачи ей требуемых свойств от Альтернативной Технической Системы.

Альтернативные Технические Системы (АТС) – это конкурирующие системы, которые имеют хотя бы одну пару противоположных достоинств и недостатков.

Конкурирующие Технические Системы – это системы, выполняющие одну и ту же Главную Функцию, но разными способами.

хуже, чем шуруп. У шурупа все наоборот: закручивается он тяжелее и медленнее, но держит лучше. Поставим задачу перенести свойства шурупа хорошо держать на гвоздь, который должен сохранить свое свойство быстро и легко забиваться.

Окончательным продуктом является гвоздь со спиральной поверхностью. Он вбивается легко, как обычный гвоздь, но, в то же время, благодаря резьбовой нарезке имеет высокую эффективность удержания, как шуруп (рис. 2.16).

Для переноса свойств с одной системы на другую может быть использовано сравнение двух или нескольких альтернативных систем.

По такой же методике объединения альтернативных систем могут формироваться образы, например, для более эффективных программных продуктов. Альтернативными системами можно считать, например, различные форматы файлов для редактирования текстов. В таблице приведено сравнение преимуществ и недостатков двух форматов: PDF и DOC.

**Таблица 2.5. Преимущества и недостатки форматов PDF и DOC**

	Преимущества	Недостатки
<b>PDF</b>	<ul style="list-style-type: none"> <li>✓ Работает на любой платформе.</li> <li>✓ Компактность (имеются алгоритмы сжатия).</li> <li>✓ Безопасность (многоуровневая система проверки подлинности).</li> </ul>	<ul style="list-style-type: none"> <li>✓ Неоптимальность использования для Web-страниц.</li> <li>✓ Сложности в редактировании текстов и изображений.</li> </ul>
<b>DOC</b>	<ul style="list-style-type: none"> <li>✓ Возможность редактировать текст, таблицы, изображения.</li> <li>✓ Проверка орфографии и синтаксиса.</li> <li>✓ Рецензирование текста.</li> <li>✓ Возможность вставки объектов</li> </ul>	<ul style="list-style-type: none"> <li>✓ Меняет вид документа с переходом на другие компьютеры (программы).</li> <li>✓ Иногда меняет содержание документа, когда это не нужно.</li> </ul>

Отбрасывая недостатки и оставляя преимущества можно сформулировать требования к новому типу формата для работы с текстовыми документами. Например, было бы полезно иметь формат данных, позволяющий легко управлять степенью изменяемости текста целиком или по частям. Нередко полезным оказывается возможность проследить (визуализировать) всю историю редактирования текста документа.

#### **Лекция 4. Модель функций. Функциональный анализ. Виды анализа**

Функция – это одно из важнейших понятий современной ТРИЗ. Модель функции представляет собой триаду: субъект (носитель) функции, действие,

объект функции. Действие может выражаться в виде глагола действия или параметра и направления его изменения. Например, пламя увеличивает температуру печки, пламя уменьшает вес воздушного шара, жидкость вблизи фазового перехода стабилизирует температуру объекта и т. д. Пример формулировки целевой функции: карандаш изменяет цвет (красит) бумаги; дешифратор восстанавливает исходное сообщение; отладочный интерфейс заносит необходимую информацию в log-файл.

Для проведения функционального анализа необходимо знать и понимать несколько терминов, которые мы перечислим ниже.

Носитель функции – субъект, реализующий рассматриваемую функцию.

Объект функции – объект, на который направлено действие рассматриваемой функции.

Полезная функция – функция, обуславливающая потребительские свойства объекта.

Вредная функция – функция, отрицательно влияющая на потребительские свойства объекта.

Нейтральная функция – функция, не влияющая на изменение потребительских свойств объекта.

Главная функция – полезная функция, отражающая назначение объекта (цель его создания).

Дополнительная функция – полезная функция, обеспечивающая совместно с главной функцией проявление потребительских свойств объекта.

Основная функция – функция, обеспечивающая выполнение главной.

Вспомогательная функция первого ранга – функция, обеспечивающая выполнение основной.

Вспомогательная функция второго ранга – функция, обеспечивающая выполнение вспомогательной функции первого ранга. Вспомогательные функции третьего и других более низких рангов – функции, подчиненные по отношению к функциям предыдущего ранга.

Ранг функции – значимость функции, определяющая ее место в иерархии функций, обеспечивающих выполнение главной функции.

Уровень выполнения функции – качество ее реализации, характеризующееся значением параметров носителя функции.

Требуемые параметры – параметры, соответствующие реальным условиям функционирования объекта.

Фактические параметры – параметры, присущие анализируемому объекту (существующему или проектируемому).

Адекватный уровень выполнения функции – соответствие фактических параметров требуемым.

Избыточный уровень выполнения функции – превышение фактических параметров над требуемыми.

Недостаточный уровень выполнения функции – превышение требуемых параметров над фактическими.

Компонентная модель – модель, отражающая состав объекта и иерархию (соподчиненность) его элементов.

Структурная модель – модель, отражающая взаимосвязи между элементами объекта. Создание компонентной и структурной моделей называется **компонентно-структурным анализом**.

Функциональная модель – модель, отражающая комплекс функций объекта анализа и его элементов.

Функционально-идеальная модель – функциональная модель после применения свертывания, отражающая комплекс функций объекта, реализуемых минимальным числом элементов.

Нежелательный эффект – недостаток объекта, выявленный в процессе анализа.

В некоторых случаях, когда нет направленного действия, а имеется лишь взаимодействие объектов, субъект функции невозможно отличить от объекта функции. Например, при взаимодействии двух радиоактивных веществ может активизироваться ядерная реакция и происходит взрыв. Оба вещества действуют в этом случае друг на друга. В оптике при создании зеркал на шлифовальной машине на каком-то этапе можно изготавливать сразу два зеркала: выпуклое и вогнутое. Оба стекла обрабатывают друг друга.

Понятие функции в ТРИЗ самым тесным образом связано с понятием «параметр». Параметр имеет несколько важных характеристик:

- параметр существует не сам по себе, он всегда привязан к тому или иному объекту, характеризует состояние этого объекта;
- изменить значение параметра можно, только воздействуя на объект;
- время является параметром для процессов или операций;
- параметр можно измерить тем или иным способом, включая экспертные оценки;
- для одного и того же параметра существуют не менее двух объектов, характеризующихся этим параметром, параметр не может быть уникальным только для одной системы;
- параметр можно увеличивать, уменьшать, стабилизировать, управлять, сравнивать;
- параметры объекта могут быть взаимосвязанными между собой;

- взаимная связь (зависимость) между параметрами объекта определяется свойствами этого объекта;
- объект может характеризоваться разными параметрами в зависимости от аспекта его рассмотрения.

Параметры объекта могут быть связаны причинно-следственными цепочками и создавать иерархические параметрические структуры.

Можно выделить материальные и нематериальные аспекты рассмотрения системы.

Материальные аспекты:

- физический (микро и макро)
- химический
- биологический
- технический
- искусство (материальная составляющая).

Нематериальные аспекты:

- психологический
- эстетически-художественный
- социальный (индивидуальный, групповой, общественный, поведенческий)
- организационно-структурный
- бизнес (бизнес-модель, методы и технологии ведения бизнеса)
- личностно-психологический
- лингвистический
- финансово-экономический
- юридически-правовой
- политический
- научно-исследовательский
- абстрактно-математический (множества, программы, формальная логика и пр.)

В зависимости от аспекта рассмотрения системы параметры могут быть:

- информационными (скорость передачи данных, надежность, защищенность и др.),
- техническими (производительность, надежность, точность измерения и др.),
- экономическими (прибыль, ликвидность, рентабельность и др.),
- физическими (температура, масса, давление, освещенность и др.),
- биохимическими (уровень глюкозы, уровень холестерина, титр антител и др.) и т. д.

Могут использоваться и узкоспециальные параметры. Например, для жестких магнитных дисков (винчестеров) используют специальные параметры: диаметр диска, число секторов на дорожке, скорость передачи данных, время перехода от одной дорожки к другой и т. д.

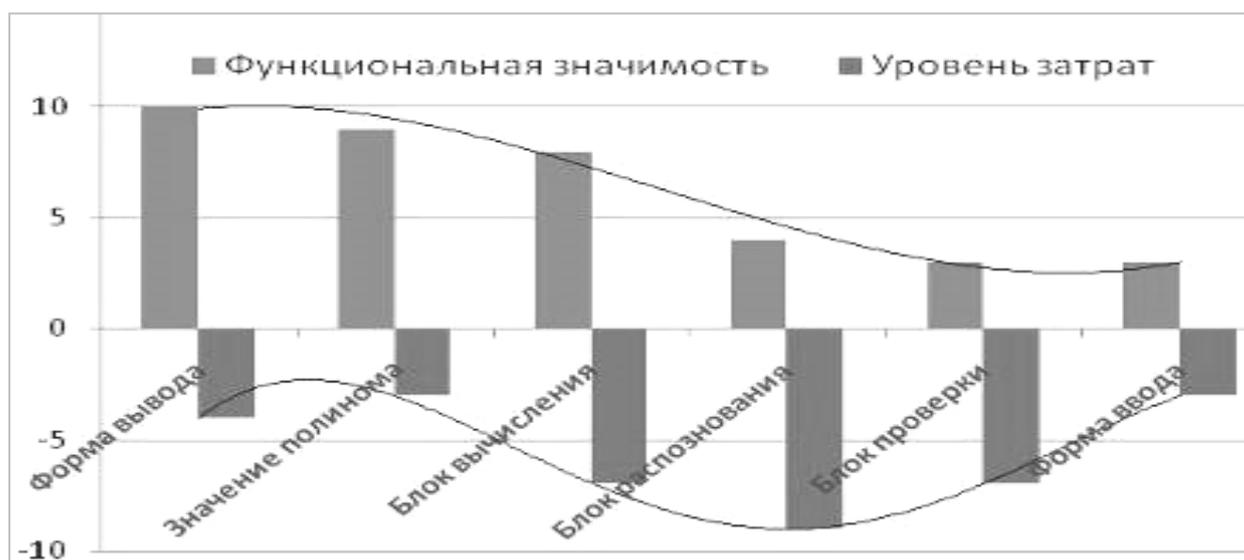
Одни параметры, например, информационные, могут формироваться как результат состояния других параметров, например, технических, физических, химических, биологических.

От качества формулировки моделей функций зависит эффективность всего функционального анализа. Имеется опасность сделать две принципиальные ошибки. Первая – сформулировать действия в форме глагола, который в действительности действие не описывает. Например, любить, работать, трудиться, исправлять – такие глаголы не помогут описать действие. Нужен конкретный параметр, который в результате этого действия изменяется. Вторая довольно типичная ошибка – неверная или неточная формулировка субъекта или объекта функции. Например, часто забывают, что объект главной функции находится за пределами рассматриваемой системы. Например, редакторы текстов направлены на взаимодействие с пользователем, который сам по себе не является частью этого редактора. При формулировках функций для нематериальных систем эти проблемы формулировки функций только обостряются. Например, в информационных технологиях объект функции и субъект функции очень часто меняются местами во времени. Так, при работе с базой данных пользователь является то поставщиком информации, то потребителем информации.

Пример функциональной модели программного продукта был приведен в разделе 2.3.2. Для построения функциональной модели необходимо вначале построить компонентную модель (из чего состоит система). Это полезно и с точки зрения поиска ресурсов для решения поставленной задачи. Затем строится структурная модель – какие элементы связаны друг с другом в системе, а какие нет. После этого для каждого компонента (элемента системы) формулируется функция или несколько функций и строится функциональная модель системы, на основе которой и проводится функциональный анализ.

Построенная функциональная модель системы позволяет, в частности, проводить причинно-следственный анализ, выделяя основные существующие в системе недостатки и выстраивая причинно-следственные цепочки для выяснения причин возникновения основных недостатков. Это позволяет сформулировать ключевые недостатки системы, решение которых, как по принципу падающих домино, приводит к устранению целой группы недостатков.

Один из вариантов функционального анализа – **функционально-стоимостный анализ (ФСА)**. Упрощенно его можно описать следующим образом. Каждому элементу ставят в соответствие определенную функцию или набор функций, определяют их значимость для системы в целом. После этого для тех же компонентов (элементов) определяют совокупные затраты. Распределение функциональной значимости элементов сравнивают с распределением затрат на этот элемент. Те элементы, которые имеют высокие затраты, связаны с большим количеством нежелательных элементов и при этом имеют не значительный функциональный ранг – это первые



**Рис. 2.17. К задаче 7: сравнение функциональной значимости элементов (сверху) и затрат на них (снизу)**

кандидаты на свертывание в этой системе.

Для примера приведена упрощенная диаграмма сравнения функциональной значимости и уровня затрат для задачи 7. Из диаграммы видно, что для блока распознавания и блока проверки соотношение между функциональной значимостью и затратами наихудшее. Эти блоки и нужно свернуть (удалить) в первую очередь (рис. 2.17).

Проведение глубокого функционального анализа с постановкой задач на свертывание – это самостоятельный раздел ТРИЗ, требующий более глубокого изучения и дополнительных инструментов анализа ситуации и постановки задач.

Еще один аналитический инструмент – **инструмент потокового анализа** (анализ имеющихся в системе потоков энергии, вещества и информации). При помощи этого аналитического инструмента могут быть выявлены недостатки, сформулированы задачи или выявлены причины их возникновения.

**Причинно-следственный анализ (ПСА)** основан на построении причинно-следственных цепочек имеющихся в системе недостатков. Эта цепочка может быть построена в виде графической или иной модели, отражающей взаимозависимость недостатков системы.

**Метод «Допустить недопустимое»** – еще один метод анализа проблемной ситуации и поиска ее решения. Его суть состоит в том, чтобы предположить такое изменение в системе, которое ни при каких обстоятельствах в условиях задачи не допускаются. Допустив такое «недопустимое» изменение далее выстраивается причинно-следственная цепочка: какие изменения возникают в системе, могут ли они снять те запреты, из-за которых нам нельзя было делать это изменение?

Простейшие примеры использования метода «Допустить недопустимое» можно взять из опыта создания презентаций. Очевидное ограничение: ширину текстового блока на слайде нельзя увеличивать так, чтобы этот блок «залезал» на окружающую его картинку. Сделаем как нельзя и все же увеличим ширину этого текстового блока. Довольно часто при этом высота текстового блока автоматически уменьшается, и увеличение его ширины уже не приводит к «залезанию» на окружающую картинку.

Для анализа ситуаций и постановки задач в ТРИЗ часто используют **диверсионный анализ**. Главная идея диверсионного анализа состоит в том, чтобы вместо решения проблемы, ставится вопрос о том, как можно создать проблему. Выделяют два направления применения диверсионного анализа в ТРИЗ. Первое – как объяснить возникновение того или иного явления. Для этого ставится задача: как создать это явление, используя только имеющиеся ресурсы системы. Второе – ставится задача о том, как можно было бы испортить систему. Это можно делать последовательно обращая все полезные функции системы на противоположные. Например, в программе сортировки нужно сделать так, чтобы элементы массива перемещались не туда, куда нужно. Зная это, можно избежать ошибку при создании программы.

### **Контрольные вопросы и задания**

1. Сформулируйте закон стремления систем к идеальности. Приведите примеры развития систем в направлении повышения идеальности.
2. Что такое ИКР? Для чего он применяется?
3. Назовите три типа ИКР?
4. Приведите примеры формулировок ИКР для разных задач.
5. Опишите линию развития «МОНО – БИ – ПОЛИ». Какие особенности можно выделить в этой линии развития. Приведите примеры.

6. Что такое свертывание систем. Приведите примеры.
7. Перечислите основные правила свертывания.
8. В чем состоит методика объединения альтернативных систем и переноса свойств? Приведите примеры использования этой методики.
9. Опишите модель функции в ТРИЗ. Приведите примеры.
10. Как можно характеризовать параметры в моделях функций? Приведите примеры параметров в функциях систем.
11. Какие аспекты рассмотрения систем можно выделить?
12. Для чего необходим функциональный анализ? Что такое ФСА?