

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

П. И. Соснин, В. В. Валюх

ЧЕЛОВЕКО-КОМПЬЮТЕРНОЕ ВЗАИМОДЕЙСТВИЕ

Учебное пособие

Ульяновск
УлГТУ
2020

УДК 621.397:004.738 (075)

ББК 32.973.202 я7

С 66

Рецензенты: доктор технических наук, главный научный сотрудник
ФНПЦ АО «НПО Марс» Иванов А. К.,
доктор технических наук, главный научный сотрудник
ФНПЦ АО «НПО «Марс» Токмаков Г. П.

Утверждено редакционно-издательским советом университета
в качестве учебного пособия

Соснин, Пётр Иванович

С 66 Человечно-компьютерное взаимодействие : учебное пособие /
П. И. Соснин, В. В. Валюх. – Ульяновск : УлГТУ, 2020. – 119 с.

ISBN 978-5-9795-2020-9

В учебном пособии представлена предметная область, прикладные задачи которых связаны с взаимодействием в человеко-компьютерных средах.

Предназначено для студентов, обучающихся по направлению «Информатика и вычислительная техника» и аспирантов, изучающих вопросы интеллектуализации активности в разработке и эксплуатации систем с программным обеспечением. Может быть полезным для специалистов, исследующих и разрабатывающих средства и системы автоматизации проектирования.

УДК 621.397:004.738 (075)

ББК 32.973.202 я7

© Соснин П. И., Валюх В. В., 2020.

© Оформление. УлГТУ, 2020.

ISBN 978-5-9795-2020- 9

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
Глава 1. ЧЕЛОВЕКО-КОМПЬЮТЕРНАЯ ДЕЯТЕЛЬНОСТЬ.....	9
1.1. Деятельность	9
1.1.1. Проблемы удобства и естественности.....	9
1.1.2. Структура деятельности.....	10
1.1.3. Опыт деятельности	11
1.1.4. Модель опыта	15
1.1.5. Модели опыта в процессах деятельности.....	18
1.2. Динамика деятельности	19
1.2.1. Активность.....	19
1.2.2. Опыт интеллектуальных действий	22
1.2.3. Управление в деятельности	24
1.2.4. Человеко-компьютерное взаимодействие	25
1.3. Предметная область «Человеко-компьютерная диалогика»	29
1.3.1. Искусственный интеллект.....	29
1.3.2. Человеко-компьютерный диалог.....	30
Контрольные вопросы	32
ГЛАВА 2. ПРОБЛЕМЫ ЧЕЛОВЕКО-КОМПЬЮТЕРНОЙ ДЕЯТЕЛЬНОСТИ	33
2.1. Особенности проектирования автоматизированных систем..	33
2.1.1. Факторы успеха и неудачи	37
2.2. Проблема сложности	39
2.2.1. Сложность и её преодоление.....	39
2.2.2. Технологические аспекты человеко-компьютерных взаимодействий	45
2.2.3. Стандарты человеко-компьютерной деятельности.....	50
2.3. Особенности деятельности человека в физическом мире и компьютерной среде	55
2.3.1. Эмпирическая природа человеческой деятельности в физическом мире	55
2.3.2. Особенности человеческой деятельности в компьютеризированной среде	61
2.3.3. Человеко-компьютерное взаимодействие с точки зрения сложности	65
Контрольные вопросы	66

ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНЫЕ ДЕЙСТВИЯ НАД КОНЦЕПТУАЛЬНЫМИ ОБЪЕКТАМИ	67
3.1. Содержательно-эволюционный подход к проектированию систем	67
3.1.1. Теоретизация в проектировании систем	67
3.1.2. Становление теории	69
3.1.3. Содержательно эволюционный подход к становлению теории	75
3.2. Экспериментирование в человеческой деятельности	79
3.2.1. Особенности экспериментальной деятельности	79
3.2.2. Разработка концепций и экспериментирование	82
3.2.3. Дизайн-мышление	87
3.2.4. Особенности мысленного экспериментирования	89
3.3. Вопросно-ответный подход к разработке теории проекта	92
3.3.1. Вопросно-ответный анализ проектных задач	92
3.3.2. Координация дизайн-мышления и QA-анализа	98
3.3.3. Вопросно-ответная структура теории проекта	100
3.3.4. Концептуальное экспериментирование в концептуальном пространстве	103
3.3.5. Теоретически экспериментальный взгляд на проект	106
3.3.6. Взаимодействие на основе опыта	109
Контрольные вопросы	110
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	111
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	119

ВВЕДЕНИЕ

Деятельностные отношения человека с компьютером образуют специфическую реальность, которая существенно изменила и продолжает изменять окружающий мир. Отношения такого типа конструируются, управляемы и результативны, но, как и любая реальность, бесконечны в своих многообразиях, а значит, открыты для исследований, нацеленных на их совершенствование.

Традиционные следующие точки зрения на роль компьютера в деятельностных отношениях:

- компьютер является не более чем инструментом, который используется субъектом деятельности (человеком или группой лиц);
- компьютерная составляющая выполняет функции, позволяющие приписать ей статус «псевдосубъекта», образующего вместе с человеком рабочую «группу» (коллектив, составной субъект деятельности).

И первая, и вторая роли предполагают наличие средств, связывающих человека и компьютер в единое целое. Вторая роль существенно богаче, поскольку её реализация должна учитывать, по крайней мере, распределение общей работы между человеком и компьютером и координированное исполнение распределённого труда. А это возможно только при наличии определённой системы человеко-компьютерного взаимодействия.

Автор предлагает ещё две точки зрения на отношения между человеком и компьютером, согласно которым:

- члены рабочего коллектива, способности каждого из которых к взаимодействию с коллегами (по общей работе) усилены компьютером, согласованно выполняют распределённую на каждого из них часть работы;
- человек использует компьютер в работе для «взаимодействия с самим собой», явно или неявно имитируя выполнение работы «коллективом».

Эти две точки зрения соответствуют такой интерпретации взаимодействия, которой больше подходит название человеко-человеко-компьютерного взаимодействия через интеллектуальный компьютерный канал связи.

Каждая из авторских точек зрения допускает понимание и реализацию компьютерной составляющей в виде интеллектуального инструментария или в виде «псевдосубъекта» («посредника» во взаимодействии), что сохраняет традиционные точки зрения.

Независимо от приведённых точек зрения, естественными аналогами для построений систем взаимодействия в компьютерной среде следует признать методы и средства взаимодействия, которые используются (или

использовались) в коллективной работе. Отбор аналогов и заимствования из них должны ориентироваться, в первую очередь, на инвариантные средства взаимодействия, что позволяет сформулировать проблему:

В слишком богатом опыте деятельностного взаимодействия, существенная часть которого используется неявно, выявить и конструктивно определить инвариантную (к объекту деятельности) часть, практически полезную для обеспечения взаимодействия в человеко-компьютерной среде.

Очень важная часть опыта взаимодействий содержится в естественном (естественно-профессиональном) языке и опыте его деятельностного употребления, в рамках которого сформировался феномен, получивший название диалог. Роль диалога, определяющего основы человеческого сознания, настолько велика, что в сформулированной выше проблеме целесообразно выделить подпроблему:

Исследовать феномен диалога и выявить в нём то конструктивное, что может быть практически полезно для взаимодействия в человеко-компьютерных средах.

Подпроблема исходит из того, что диалог – это естественно-искусственное образование, знаковое выражение которого является вторичным. Первичное назначение диалога, подтвердившее свою практическую полезность в эволюционном развитии человечества, воплощено в феномене сознания. Сознание и его механизмы инвариантны к объекту деятельности. Приведённое выше понимание деятельностного взаимодействия и диалога указывают на принципиальную роль эксперимента в исследованиях реальностей такого типа. В этом плане просматривается следующий подход к попыткам решения отмеченных проблем.

Применения компьютеров в экспериментах по исследованию феноменов деятельностного взаимодействия и диалога приведут к разработке совокупности методов и средств такой экспериментальной деятельности, в составе которых будет присутствовать система методов и средств, обеспечивающая взаимодействие и диалог в человеко-компьютерных средах.

К месту заметить, что роль эксперимента способен выполнить конкретный процесс деятельности в человеко-компьютерной среде, например, процесс решения задач по ходу проектирования некоторого объекта. Такая роль деятельности открывает возможность исследования взаимодействий и диалога, которые используются (или использовались) в работе. Открыты различные версии реализации сформулированного выше подхода, в которых логично ориентироваться на применение

традиционных подходов: системного, функционального, эволюционного и других подходов.

Особого внимания заслуживает эволюционная версия реализации подхода, использующая метод пошаговой детализации, которая нацеливает на повторение эволюционного развития системы методов и средств деятельностного взаимодействия. Такая версия, названная авторами «содержательно-эволюционный подход», исследовалась им с различных позиций, в частности для определения системы методов и средств диалога, инвариантных к содержанию деятельности.

В основе версии лежат:

- интерпретация конкретной деятельности в человеко-компьютерной среде как «эксперимента», который нацелен на выявление (или, по крайней мере, проверку и подтверждение) практически полезных методов и средств деятельностного взаимодействия;

- сбор полезной первичной информации в таком «эксперименте» в виде протокола, регистрируемого в вопросно-ответной форме;

- понимание вопроса как специфического образования реальности, представляющего рассогласование между тем, что требуется для действия в текущей ситуации «эксперимента», и опытом экспериментатора (или доступным ему опытом),

- отработка рассогласований-вопросов через формирование соответствующих ответов (знаковое выражение вопросов и ответов считается вторичным, но помогающим закодировать и раскрыть их первичную сущность);

- систематизация опыта, приобретённого в «эксперименте», в том числе в виде прикладной теории содержательно-эволюционного типа.

В научном коллективе, которым руководил профессор Соснин П.И., проведён ряд «экспериментов», каждый из которых связан с разработкой определённой программной системы (например, САПР тестов микропроцессорных схем, конструктор реляционных баз данных, ряд автоматизированных обучающих курсов). В таких «экспериментах» был выявлен, конструктивно определён и реализован набор инвариантных методов и средств, автоматизирующих реализацию содержательно-эволюционного подхода.

В состав построенного набора входит подсистема методов и средств, обслуживающих диалог в человеко-компьютерной среде. Этот набор существенно отличается от традиционных методов и средств построения и использования систем, получивших название диалоговых. Отличие заключается в том, что диалог понимается и реализуется как средство доступа к «опыту», представленному в человеко-компьютерной среде. Обнаружено, что управление доступом к опыту вводит в деятельностные процессы специфическую составляющую, названную «вопросно-ответным

управлением». С исследованиями человеко-человеко-компьютерного взаимодействиями и разработками инструментария и технологий, согласованного с таким взаимодействием, автор предлагает связать отдельную предметную область и назвать её «человеко-компьютерная диалогика».

К «человеко-компьютерной диалогике» авторы относят множество задач, в работе с которыми приходится моделировать «опыт» и взаимодействие с «опытом» в человеко-компьютерных средах. В учебном пособии раскрывается место предметной области «человеко-компьютерная диалогика» в проблематике систем искусственного интеллекта, приводятся её основные спецификации и детально раскрывается всё существенное. Основная цель пособия – заинтересовать потенциальных исследователей и практиков уже достигнутыми результатами, для того чтобы расширить исследования человеко-человеко-компьютерного взаимодействия и направить разработки на создание более совершенных инструментальных систем и технологий деятельности в человеко-компьютерных средах. По этой причине текст данного пособия часто переходит на демонстрационно-утверждающий стиль.

Глава 1. ЧЕЛОВЕКО-КОМПЬЮТЕРНАЯ ДЕЯТЕЛЬНОСТЬ

1.1. Деятельность

1.1.1. Проблемы удобства и естественности

В основе новых информационных технологий лежит интерактивное взаимодействие человека и компьютера, выполняющих общую работу в человеко-компьютерной среде. У каждого из участников вычислительного процесса своя роль: человек, как активная созидаящая сила, осуществляет ситуативное управление ходом работы (деятельности), принимая решения и контролируя их влияние на вычислительный процесс; компьютер обеспечивает инструментальную поддержку действиям человека и реализует возложенную на компьютер часть вычислений.

Результативность вычислительных процессов в интерактивных системах и другие показатели рациональности процессов в существенной мере зависят от решений, принимаемых человеком, и от условий интерактивного (человеко-компьютерного) взаимодействия, которое используется для поддержки принятия решений и для воплощения их в реальность. Принципиальное влияние на коэффициент полезного действия человека в такой работе оказывают характеристики, различаемые человеческой психикой и связанные с феноменами естественности и удобства.

Реальность человеко-компьютерной практики такова, что существует и постоянно проявляет себя феномен неудобства во взаимодействии человека с компьютерными средами. Можно утверждать, что предоставляемая человеку степень естественности во взаимодействии с компьютером явно недостаточна. Человеку неудобно потому, что в отношениях с компьютером он лишён естественной опоры деятельности (прочной основы на врождённую и приобретённую в деятельности систему автоматизмов психики). При взаимодействии с компьютером у человека в его мозговых структурах нет концептуальных пространства и времени, которые вызывают впечатление реальности (естественности «окружающей» компьютерной среды) и к которым можно проникнуться доверием на уровне здравого смысла.

Человек – целеустремлённое, деятельностное «образование», и в его взаимодействии с вычислительной средой наблюдается «дефект естества», который возникает из-за неадекватности естественных способностей человека средствам связи с вычислительной средой, предоставляемым ему компьютером. Отмеченное противоречие указывает на актуальность исследований и разработок, направленных на построение инструментальных средств человеко-компьютерной деятельности,

удобных и естественных для принятия решений, что приводит к проблеме деятельности P_0 :

P_0 : Для увеличения результативности вклада компьютерного инструментария в практику, инструментарий необходимо построить (модифицировать) так, чтобы он стал более естественным и, посредством этого, более удобным и рациональным для принятия решений в человеко-компьютерной деятельности.

1.1.2. Структура деятельности

Для работы над проблемой P_0 , в первую очередь, следует определиться с понятием «человеко-компьютерная деятельность». Любой вариант употребления понятия «деятельность» предполагает существование основных составляющих деятельности – субъекта и объекта. Субъект Sb деятельности D своей активностью $\alpha(t)$ как своеобразной созидающей силой обеспечивает как порождение D , так и продвижение ее к цели – практически полезному образованию (или преобразованию) объекта Ob деятельности D . Любая деятельность – это искусственный объект D , который, как и любое практически полезное образование, создается в результате специфической деятельности – метадеятельности MD метасубъекта Sb^{MD} над метаобъектом $Sb^{MD} = D$. Такое отношение между деятельностью и метадеятельностью представлено в образной форме на рис. 1.1.

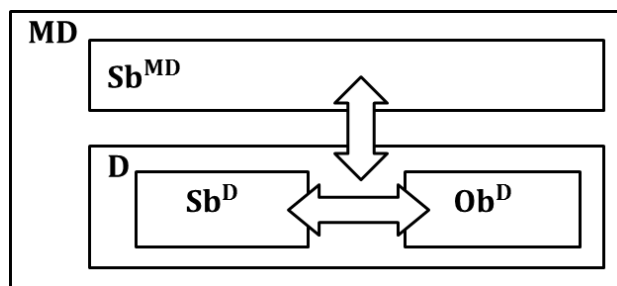


Рис. 1.1. Схема деятельности

В общем случае реальные отношения между деятельностью и метадеятельностью намного сложнее. Во-первых, для метадеятельности как деятельности, в принципе, должна существовать метаметадеятельность и так далее. Во-вторых, каждая деятельность в состоянии породить подчинённую ей деятельность с соответствующим метаобеспечением. А значит, реальное развёртывание практически полезной активности может привести (и обычно приводит) к сложно структурированным образованиям из «блоков», относящихся к категории деятельностей, причём естественную структуризацию любой деятельности определяет

система принятых решений. Исследование деятельностных отношений ставит вопросы о разработке и реализации взаимодействия, связывающего субъектов деятельности и метадеятельности. Объектом такой деятельности является субъект Sb как объективированный субъект Sb^{Ob} . Типичным и самым простейшим способом взаимодействия является управление субъекта Sb метасубъектом Sb^{MD} . В общем случае субъект Sb может состоять из некоторого множества составляющих Sb_i , каждая из которых имеет статус субъекта. В этом случае приходится объединять составляющие Sb_i в единое целое – составной субъект. Взаимодействие субъектов усложняется. Одного управления оказывается уже недостаточным. Перед метасубъектом Sb^{MD} встает задача распределения труда между субъектами Sb_i , исполнителями общей работы, а для субъектов Sb_i возникает принципиальная проблема общения выработки их общности как составного субъекта. Типичной формой установления общностей является диалог субъектов. Необходимая общность обязана достигаться и при взаимодействии субъектов Sb_i и Sb^{MD} .

Для практики типичен и интересен случай (рис. 1.2), когда субъект Sb^{MD} метадеятельности MD является субъектом Sb^D и деятельности D .

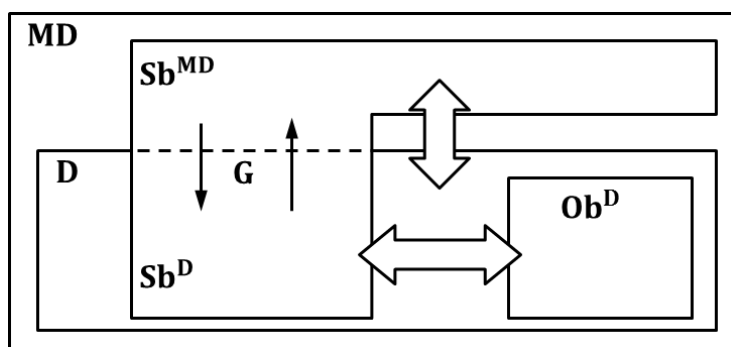


Рис. 1.2. Схема деятельности

Этот случай ставит вопросы о реализации переключений между MD и D через границу G между $Sb^D = Sb^{MD} = Sb$, а также рациональности переключений и необходимых для этой работы средствах (инструментах). Переходы между MD и D также можно связать с взаимодействием, положив в его основу самоуправление и само общение субъекта, переключающегося между ролями субъекта и метасубъекта.

1.1.3. Опыт деятельности

Как уже отмечалось, продвижение D к цели (динамика D , развертывание $D(t)$ во времени t) зависит от активности $\alpha(t)$ субъекта, принимающего решения и воплощающего их в $D(t)$. Активность $\alpha(t)$ – естественно-искусственный процесс, сущность которого определяет то,

что получило название «принятие решений». Естественность $\alpha(t)$ обусловлена психофизиологической природой человека, а искусственность – управляемым введением в её процессы специальных деятельностных составляющих.

В общем случае активность $\alpha(t)$ – сложно структурированное образование, состоящее из активностей $\{\alpha^i(t)\}$, специализирующихся на выполнении определённых функций. Внешнее проявление $\alpha^B(t)$ активности $\alpha(t)$, приводящее к конкретным практически полезным изменениям в субъекте и/или его окружении, основано на опережающем отображении человеком осуществляемой им деятельности. В опережении активно мышление $\alpha^M(t)$ субъекта, открывающее возможность для прогнозирования отношений человека с реальностью R . Опережающему отображению D^O можно приписать статус деятельности, но конструируемой в субъективной действительности R^C , локализованной в мозговых структурах субъекта (рис. 1.3).

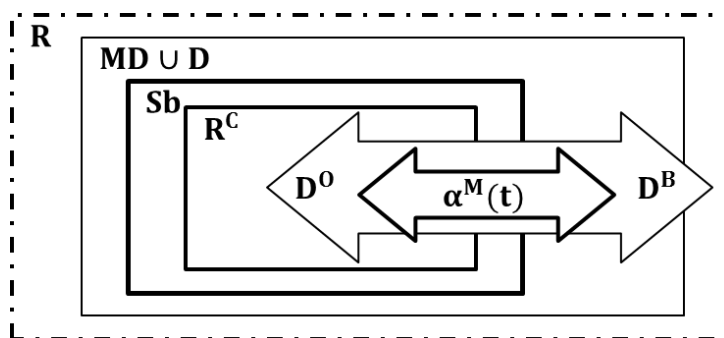


Рис. 1.3. Активность субъекта

Между D^B как внешним (по отношению к телу человека) проявлением деятельности D и опережающей составляющей D^O существует отношение потенциального репродуцирования, которое активизируется, если субъект принимает решение включить в деятельность D составляющую D^B , реализуемую по образцу D^O .

Формирование образцов типа D^O , их накопление «впрок» и репродуцирование (в подходящих условиях в формах определённых действий) эволюция довела до образования того, что получило название «опыт».

Для обозначения опыта введём символ E , а для обозначения действия как составной части $D(t)$ – символику $d(t)$. Действия $d(t)$ традиционно принято понимать как средства структуризации $D(t)$.

Под опытом E_i конкретного индивида будем понимать систему условных деятельностных рефлексов, информационные коды которой локализованы в мозговых структурах носителя опыта.

Такое понимание определяет позицию авторов, которую они будут развивать и обосновывать, определяя и применяя использованные названия. Так, с каждым условным деятельностным рефлексом $d^e_i(t)$ свяжем логическую структуру следующего типа:

$$\left. \begin{array}{l}
 d^e_i(t): \text{ так как [мотивы } \mathbf{M}] \\
 \text{поскольку [цели } \mathbf{C}] \\
 \text{в} \left\{ \begin{array}{l} \text{если [предусловия } \mathbf{U}'], \\ \text{то [реакция } \mathbf{r}_q(t), \\ \text{из-за чего [постусловия } \mathbf{U}''] \\ \text{-----} \\ \text{были альтернативы } [\{\mathbf{r}_p\}], \end{array} \right.
 \end{array} \right\} \quad (1.1)$$

осуществляемых в определённой системе причинно-следственных отношений (мотивов, целей, предусловий и постусловий, потенциальных альтернатив) субъекта с окружающей его действительностью.

Замечание 1. Название «условный деятельностный рефлекс» для тех образований, которые можно поставить за логической схемой (1.1), выбрано специально, поскольку в деятельностной активности человека слишком принципиальна и велика роль психического.

Замечание 2. «Трёхступенчатая» схема условия в рефлексе отражает отношения субъекта с тем, что прямо или опосредовано вовлечено в деятельность (рис. 1.4)

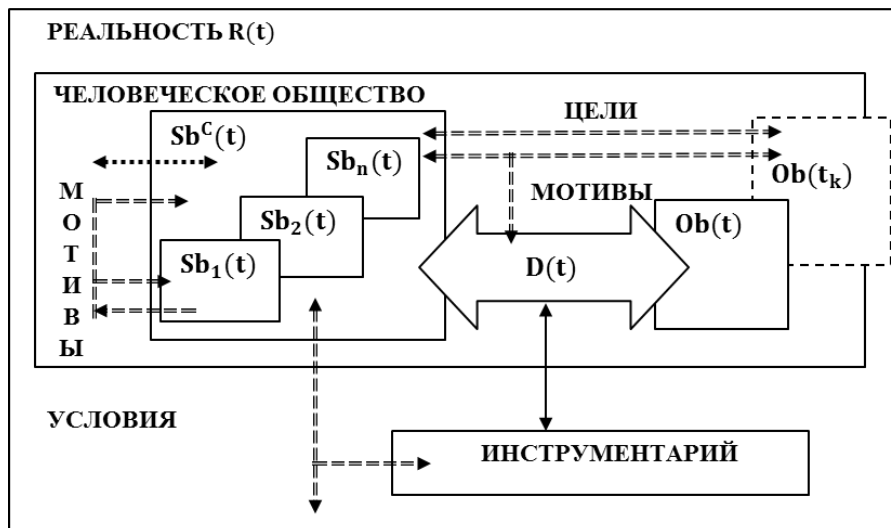


Рис. 1.4. Схема отношений субъекта

Рефлексы типа (1.1) наблюдаемы и проявляют своё реальное существование подобным образом в подобных условиях. Их можно изучать, используя методы и средства научного исследования.

Рефлексы типа (1.1) являются единицами, которые включаются в структуру $D(t)$ и, по этой причине, могут быть отнесены к классу действий. Условный деятельностный рефлекс содержит все существенные спецификации того, что принято называть «действием». Открыты два типовых варианта активизации и исполнения деятельностного рефлекса:

- как навыка, то есть в форме автоматической реакции, в процессе реализации которой все необходимые проверки условий рефлекса проводятся автоматически без явного обращения к сознанию исполнителя;
- как умения, в котором, по крайней мере, часть проверок условий, требует явного обращения к сознанию.

Конкретное умение из-за частых и успешных применений способно перейти в класс навыков, что существенно экономит расход умственных усилий и сокращает время реагирования. Как только неявные проверки условий в навыке фиксируют выход исследуемого условия за границы области определения «предполагаемого рефлекса», должен вырабатываться информационный сигнал к сознанию «обратить на этот факт внимание».

Одно из направлений научного исследования множества рефлексов нацелено на выявление и конструктивное овладение естественными механизмами доступа к информационным кодам рефлексов, то есть овладение механизмами доступа, которые использует человеческий мозг. Нейронная сеть мозга – это среда (сложная среда), в которой развёртываются доступ к информационным кодам рефлексов и их настройка на реагирование.

Таким образом, для условно-деятельностного реагирования в мозговых структурах субъекта деятельности должна создаваться, использоваться и совершенствоваться система доступа к исполнительным кодам навыков и умений (к процедурам навыков и умений). С такой системой доступа логично связать понятие «знания субъекта», что позволяет обобщённо представить опыт E_i субъекта так, как показано на рис. 1.5.

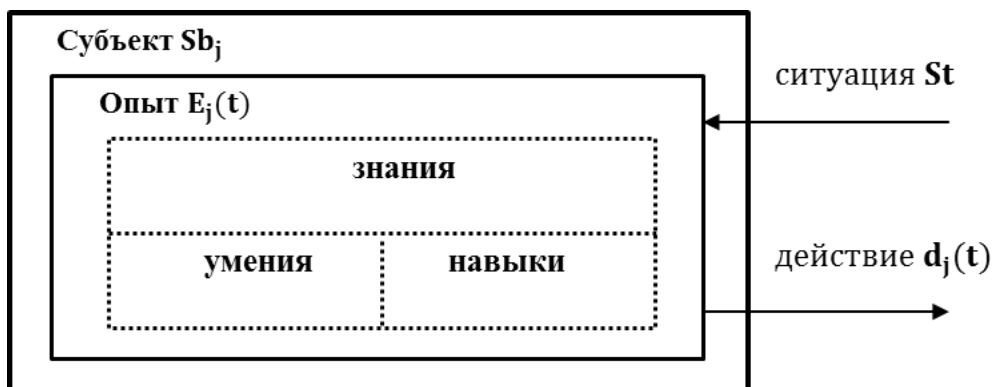


Рис. 1.5. Структура опыта

В структуре опыта граница между умениями и навыками помечена штриховой линией, для указания на возможность переходов процедур из одного класса в другой.

Коллективная деятельность (обозначим такой тип деятельности $D^K(t)$) приводит к необходимости объединения (рис. 1.6) и согласованного использования опыта $E = \cup E_i$ группы субъектов, что обеспечивается специальными средствами (естественно-профессиональный язык, другие модели систем доступа или их частей, то есть другие модели знаний).

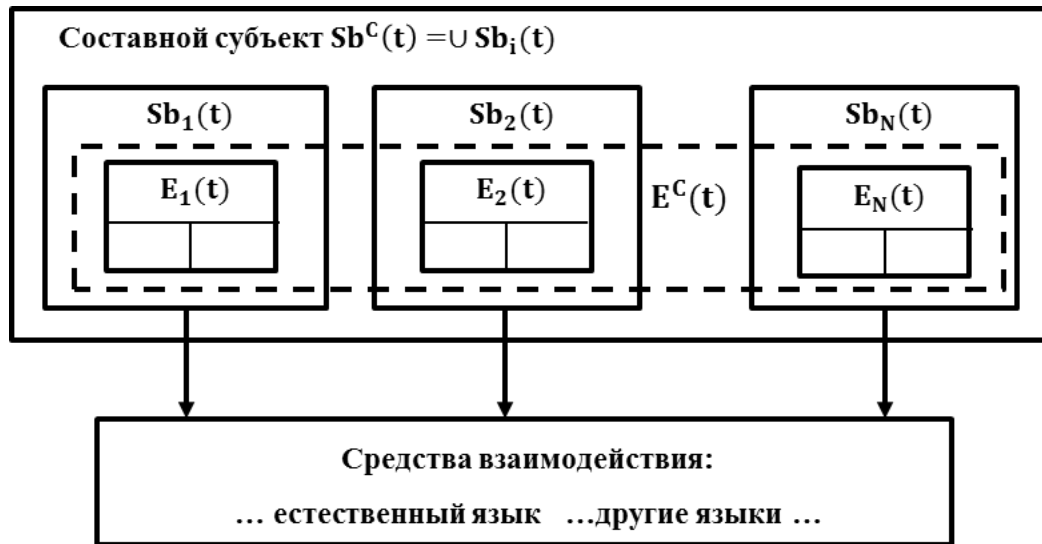


Рис. 1.6. Общий опыт

В коллективной деятельности уже давно осознана ценность коллективного опыта и объединённых знаний, как системы методов и средств доступа к коллективному опыту.

1.1.4. Модель опыта

Объединённые знания при их использовании в коллективной деятельности $D^K(t)$ требуют представить определённую часть системы опыта $E = S(\{E_i\})$ в форме, вынесенной за пределы мозговых структур, то есть в форме некоторой модели $M_j(E)$ опыта E . Это необходимо для согласованного взаимного действия субъектов в рамках $D^K(t)$.

К построениям модели можно подойти с позиций систематизации множества деятельностных рефлексов, исходя из их типовой логической структуры (1.1). Для того чтобы рефлекс стал доступным группе субъектов, он должен получить информационное представление в виде кода, взаимодействие с которым способно привести к исполнению рефлекса. В этом плане информационный код должен быть подобен компьютерной программе и содержать декларативные и процедурные

составляющие, управляющие проверками условий и исполнением «тела рефлекса».

Исполнение рефлекса предполагает и содержит проверку условий на трёх уровнях:

- проверка на уровне мотивов определяет достаточность причин, побуждающих субъекта (или субъектов) к приложению своих сил к исполнению рефлекса;

- проверка на уровне целей устанавливает соответствие предполагаемого реагирования целям деятельности, в которую рефлекс должен включиться;

- проверка на уровне предусловий и постусловий, контролирует, во-первых, возможность реагирования в сложившихся условиях, а во-вторых, положительность (практическую полезность) изменений в среде деятельности (в том числе положительность изменений объекта деятельности).

Такие проверки носят контролирующий характер, а значит, возможны только при условии информационного взаимодействия субъекта (или субъектов) со средой деятельности, в основе которого лежит феномен «восприятия».

Субъект, используя код рефлекса как подсказку:

- с помощью органов чувств и средств измерений извлекает и селекционирует первичную информацию о сложившейся ситуации;

- обрабатывает первичную информацию, порождая входные данные для процедур проверки.

Сложившийся опыт моделирования феномена «восприятия» способен подсказать подход к выбору способов и средств кодирования рефлексов. Наиболее результативно и практически полезно феномен «восприятия» моделируется в предметной области «распознавание образов и анализ сцен». Ориентация на этот опыт полезна и в том, что распознавание образов и анализ сцен входит в область интересов искусственного интеллекта.

Предположим, что по аналогиям с задачами распознавания образов для кодирования мотивов, целей, предусловий и постусловий используется признаковое пространство, то есть конкретная распознаваемая единица представляется перечнем признаков. В этом случае множество распознаваемых единиц (множество перечней признаков с их значениями) отображается на систему классов, которая способна выполнить роль системы доступа в модели $M(\{E_i\})$.

В распознавании образов и анализе сцен накоплен большой багаж методов и средств, ориентированный не только на представления в пространстве признаков. В частности, для повышения результативности и эффективности распознавания, первичную информацию отображают на

различного рода формальные системы (алгебраические, логические, лингвистические и другие), что присоединяет к системам классификаций дополнительные системные средства доступа и приводит к новым типам моделей $M(\{E_i\})$.

На практике используют как автоматические, так и автоматизированные средства, исходящие из оперативного использования распознающих и аналитических способностей человека. В автоматизированном распознавании не избежать применений естественного или естественно-профессионального языка, а значит и вложенных в такой язык средств доступа к опыту, по крайней мере, доступа к опыту распознавания образов и анализа сцен.

Автоматизированный подход к кодированию, систематизации и распознаванию деятельностных рефлексов широко используется при разработке и использовании экспертных систем различного назначения. В экспертных системах сложилась технология построения прикладных моделей опыта $M_j(\{E_i\})$, в рамках которой применяются системы классификаций, различного рода формальные системы и естественно-профессиональные языки.

Обобщая вышесказанное, будем утверждать, что в моделях опыта, в зависимости от того для каких приложений они предназначены, можно применять и применяют различные системы доступа (различные конструктивные представления знаний), но в каждой из них, в то или иной мере, присутствуют элементы распознавания образов и анализа сцен, элементы формализации и употребления естественно-профессионального языка. Этот факт в образной форме представлен на рис. 1.7.



Рис. 1.7. Модели опыта

Одной из принципиальных характеристик любой модели, в том числе и моделей опыта, является их адекватность моделируемому объекту. Степень адекватности должна оцениваться, подтверждаться и предъявляться вместе с моделью. В том случае, когда адекватность модели

подтверждена, модель выполняет функции индикатора существования объекта, что позволяет дополнить ответ приращением ответа:

Модели индивидуального или коллективного опыта, подтвердившие свою адекватность в их применении, являются одной из форм, через которую опыт проявляет своё существование.

К числу таких моделей можно отнести все модели знаний, подтвердившие свою полезность на практике, например: системы классификаций; различного рода теории; системы инструментальных средств как материализованные знания. Моделирование знаний составляет существенную часть предметной области искусственного интеллекта.

1.1.5. Модели опыта в процессах деятельности

Предположим, что субъект Sb_i построил модель опыта $M_j(\{E_i\})$ и использует её в своей профессиональной деятельности. С деятельностных позиций открыты два варианта интерпретации $M_j(\{E_i\})$:

- как дополнения к субъекту, расширяющего структуры его опыта;
- как дополнения к инструментальным средствам.

В проблематике искусственного интеллекта обычно используется первая интерпретация, поскольку она ставит вопросы о действиях над $M_j(\{E_i\})$ и взаимодействиях с $M_j(\{E_i\})$. В такой интерпретации естественным аналогом действий и взаимодействий является специальный класс деятельностных рефлексов, освоенных в работах с опытом E . Именно этому классу принадлежат рефлексы, получившие название «интеллектуальные действия».

В более общем случае модель опыта строится и используется группой субъектов (рис. 1.8).

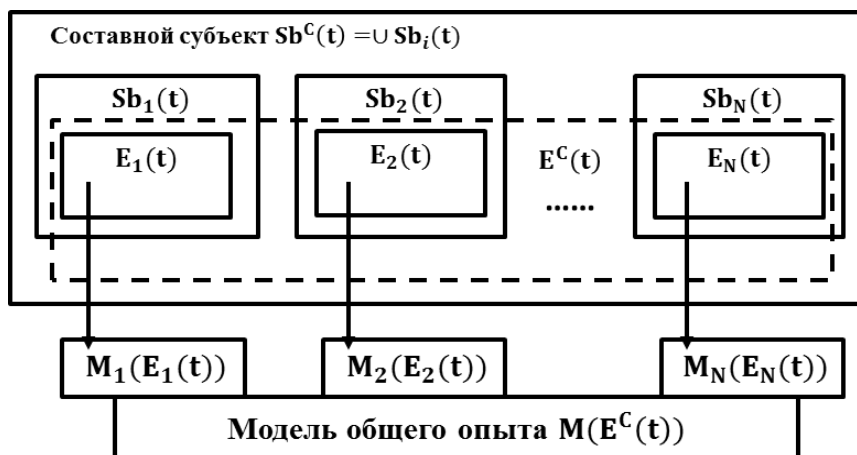


Рис. 1.8. Опыт в совместной деятельности

Структура опыта каждого из субъектов расширяется, а расширения пересекаются, обеспечивая объединение опыта субъектов в единое целое – коллективный опыт. Появляется возможность переходов в общей сети доступа от опыта одного субъекта к опыту другого, например:

- для согласованного распределения и выполнения работ при исполнении одного и того же деятельностного рефлекса;
- для объединения рефлексов, исполняемых каждым из субъектов в единый деятельностный процесс.

Следует ожидать (и это наблюдается), что коллективное построение и применение опыта расширяет количество действий над моделью опыта, расширяя и множество интеллектуальных действий. К месту заметить, что в построениях, развитии, совершенствовании и применениях общей модели опыта особо принципиальна роль средств систематизации и доступа, которые образуют естественно язык профессиональной коллективной деятельности группы субъектов. Совокупность действий, обслуживающих построения и применения естественно-профессионального языка (обозначим **L**), относится к классу действий с опытом. В этот класс действий входят и такие, которые принято квалифицировать как интеллектуальные.

1.2. Динамика деятельности

1.2.1. Активность

Для деятельности $D(t)$ наиболее принципиальными являются отношения между активностью $\alpha(t)$ субъекта и опытом E . Если эти отношения рассматривать с объектно-ориентированных позиций, то опыт **E** логично понимать как образование, которое входит (рис. 1.9) в состав декларативной составляющей активности $\alpha(t)$.



Рис. 1.9. Активность и опыт

Процедурную часть $\alpha(t)$ составляют различные механизмы, в том числе и те, которые обслуживают феномены «восприятия», «представления», «установки», «сознания», «понимания», «интуиции», «внимания» и «ассоциаций», каждый из которых проявляет себя в «мышлении».

Мыслительная активность $\alpha^M(t)$ отвечает за взаимодействие субъекта с субъективной действительностью R^C , отображающей реальность R в формах, способствующих опережающему отображению D^0 , то есть в формах опыта. Субъективная реальность R^C распределена между правым и левым полушариями мозга, каждое из которых использует свои способы структуризации соответствующей части R^C и предоставляет активности собственные специфические методы и средства для созидания D^0 . Основу правополушарных механизмов, используемых в правополушарной части ($R^C_{\text{П}}$) реальности R^C , определяют представления образований из R (отображения единиц R в единицы R^0), которые получили название «образ», а основу левополушарных механизмов (в левополушарной $R^C_{\text{Л}}$ – части) определяют представления, получившие название «понятия». Наличие право- и левополушарных сфер отображения приводит к необходимости в активностях для каждой из сфер. Особенно важной в активности $\alpha^M_{\text{П}}(t)$ правого полушария является интуиция, а в активность $\alpha^M_{\text{Л}}(t)$ левого полушария принципиальна роль сознания. Разумеется, активности $\alpha^M_{\text{П}}(t)$ и $\alpha^M_{\text{Л}}(t)$ входят в состав $\alpha^M(t)$, подчиняясь (рис. 1.10) общему управлению.

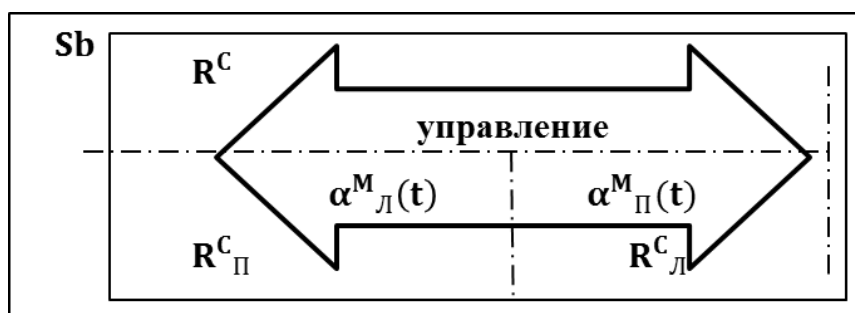


Рис. 1.10. Среда мышления

Деятельность D^0 – искусственный объект, создаваемый и используемый в результате метадеятельности, включающей в работу активность $\alpha^{MM}(t)$ как метамышление, что немислимо без представления D^0 (и составляющих D^0) для реализации целей метадеятельности MD . Необходимость в подобных представлениях настолько принципиальна, что приводила и приводит к изобретениям и разработкам специальных инструментальных средств, в число которых входят различные знаковые системы, в том числе языки. Знаковые средства позволяют вынести представления, используемые в деятельности D^0 , из мозговых структур наружу в реальность R , построив для D^0 практически полезные модели, например, выполняющие роль протоколов работы активности $\alpha^M(t)$, например, её составной части – активности сознания $\alpha^C(t)$.

Типичной формой протокола, овеществляющего объект D , является проект деятельности D , подготовленный для его репродуцирования на внешнее проявление D^B деятельности D . Типичной формой протокола работы сознания является текст T на соответствующем языке. Типичными овеществлёнными (вынесенными за пределы мозга) формами для правополушарных (целостных) образований являются схема, рисунок, знак или описание (обычно в виде текста) представляемого образа. Опережающее отображение D^O или его составные части, а также овеществленные представления D^O «подчинены» деятельности MD , а значит и активности $\alpha^{MM}(t)$. Особо важными функциями метамышления $\alpha^{MM}(t)$ являются:

- замысливание D^O или частей D^O , в котором основные надежды на интуицию $\alpha^I(t)$;
- подготовка «рабочего места» в «пространстве» R^C с помощью процессов $\alpha^Y(t)$ формирования установки;
- поиск подходящего деятельностного образца (или образцов) в опыте E_i с опорой на ассоциативную активность $\alpha^A(t)$;
- контроль за ходом D^O и её результатами в основном в формах логической активности $\alpha^L(t)$ и в формах понимания $\alpha^П(t)$.

Проведённые рассуждения об активности демонстрируют, что с каждым из названных выше феноменов в мышлении (и другими, не названными феноменами, образующимися в «мышлении») удобно связать определённый тип активности $\alpha^i(t)$, что открывает возможность для структуризации (рис. 1.11) процесса $\alpha(t)$.

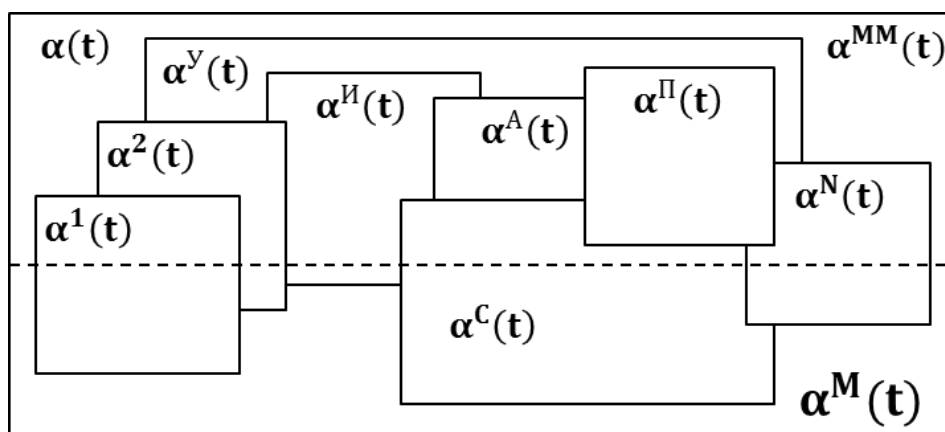


Рис. 1.11. Структуризация активности

В конкретном мыслительном процессе естественно или искусственно создаются условия, в которых порождаются и протекают различные комбинации процессов из множества $\{\alpha^i(t)\}$, вносящих свой вклад в $D(t)$. С искусственной составляющей $\alpha^i(t)$ или их комбинаций полезно

связывать определённый тип мыслительного действия или деятельности, например «соображение», «рассуждение» и другие. Подключение к $\alpha^i(t)$ искусственной составляющей позволяет управлять феноменом $\alpha^i(t)$, а значит и результативностью его вклада в $D(t)$.

1.2.2. Опыт интеллектуальных действий

В множестве $\{\alpha^i(t)\}$ принято выделять подмножество интеллектуальных активностей, составляющих (и проявляющих) то, что получило название «интеллект». Интеллект также следует рассматривать как естественно-искусственное образование, локализованное в мозговых структурах субъекта, в котором искусственная составляющая открыта для развития и совершенствования.

Интеллектуальные действия принципиально необходимы и выполняют существенную роль: в речевой деятельности (производство, преобразования и восприятие текстов) и языковой деятельности (разработка знаковых средств и механизмов, обеспечивающих реализацию речевой деятельности); в распознавании образов и анализе сцен; в оценочной деятельности, в том числе с помощью экспертов; в процессах решения задач и принятия решений (выбора решения из множества альтернатив).

Авторы убеждены, что основным объектом интеллектуальных действий является опыт E , в частности: его развитие за счет включения в опыт E кодов нового деятельностного рефлекса; внесение положительных изменений в систему доступа (в знания), улучшающих характеристики доступа; явное или неявное взаимодействие с опытом при его применениях. К категории интеллектуальных действий следует относить и действия по созданию, развитию, совершенствованию и использованию коллективного опыта, в том числе и той его части, которая погружается в компьютерные среды.

Совокупность интеллектуальных действий выделяет в опыте человека его часть, которую логично назвать опытом интеллектуальных действий. Опыт интеллектуальных действий конкретного субъекта целесообразно рассматривать как метаопыт, то есть как опыт E^M_i действий с опытом E_i , входящий в состав E_i . Опыт E^M включает метазнания, метанавыки и метаумения (рис. 1.12).

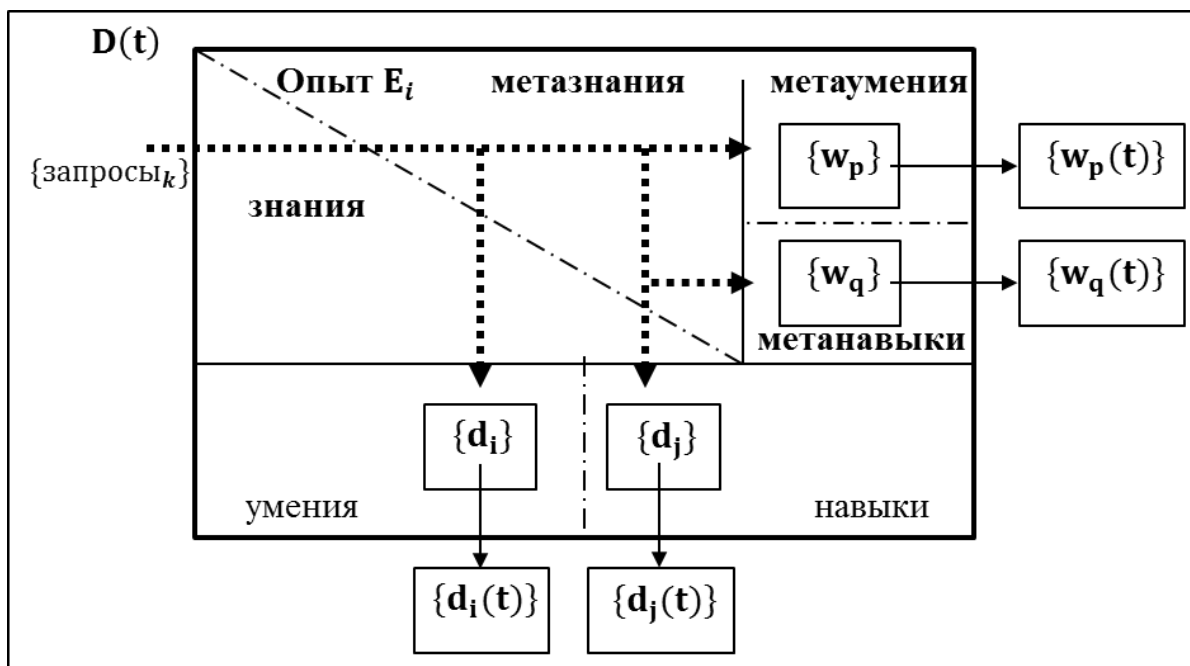


Рис. 1.12. Структура опыта

Наиболее целенаправленно и активно опыт интеллектуальных действий формируется и используется в научной деятельности, основным продуктом которой являются знания. Знания производят в первую очередь для того, чтобы их применять при взаимодействии с окружающим миром. В производстве знаний различают и согласованно используют два основных типа деятельности: понятийную деятельность (построения различного рода знаковых систематизаций – систем понятий, теорий) и предметную деятельность (сущность которой составляет экспериментирование).

Исследования названных типов деятельностей позволяют выявить определённый набор интеллектуальных действий и систематизировать их, образовав определённую модель опыта таких действий. Строя такую модель следует ориентироваться на её применения не только в отмеченных выше типах деятельности, но и в деятельности других типов.

Опыт – это целостное образование, которое включается в работу, как только для этого создаются условия, нашедшие своё отражение в опыте. Практика демонстрирует, что при планировании и исполнении конкретной деятельности, действия различного типа образуют параллельно-последовательные комплексы, в составе которых выделяются и различаются «линии действий», например, линии «понятийных действий» и «предметных действий», а в составе понятийных действий, например, линии «речевых действий» и «языковых действий».

Такое представление деятельностного процесса предполагает распределение ресурсов субъекта (в том числе и его опыта) между

линиями действий и переключений между линиями (исполняемых навыков и умений). Нет деятельности, в которой бы явно или неявно не использовался опыт интеллектуальных действий. А значит, обладая моделью опыта интеллектуальных действий, её можно применить для улучшения характеристик любой деятельности.

Более того, интеллектуальные действия можно выявить, систематизировать и развить, выполняя и (параллельно с исполнением) исследуя любые деятельностные процессы. Роль исполнителя такой работы целесообразно возложить на субъект деятельности, поскольку:

- количество источников новых знаний об интеллектуальных действиях существенно увеличивается;
- обнаруживается специфика интеллектуальных действий и их композиций, зависящая от типа исполняемой деятельности;
- субъекты типовой профессиональной деятельности заинтересованы в совершенствовании своего опыта, в том числе его интеллектуальной составляющей.

Выявление и конструктивное овладение опытом интеллектуальных действий представляет собой сложную задачу. Это обусловлено тем, что слишком большая часть интеллектуальных действий осваивается и переводится в разряд автоматизмов (навыков) в детском периоде. Последующее развитие совокупности интеллектуальных действий чаще всего происходит неявно, и успешные их образцы также переходят в разряд автоматизмов. Даже явное и контролируемое развитие совокупности интеллектуальных действий переводит успешные и часто используемые образцы действий в разряд автоматизмов, исключаемых из под контроля сознания и со временем «забываемых».

1.2.3. Управление в деятельности

Предположим, что в определённый момент времени t_0 в определённой области пространства \mathbf{R} сложилась ситуация $St(t_0)$, в которой индивид Sb_i должен был приступить к деятельности $D(t)$. И пусть в момент времени t_k он завершил процесс $D(t)$ с положительным результатом воздействия на объект Ob_k .

Реализацию $D(t)$ сопровождал определённый объем активности $\alpha(t)$ субъекта в форме определённой комбинации моторных и психических действий, включающей мыслительные действия, в том числе и интеллектуальные действия. Логично считать, что от организации этих действий существенно зависят характеристики как процесса $D(t)$, так и его результата в Ob_k .

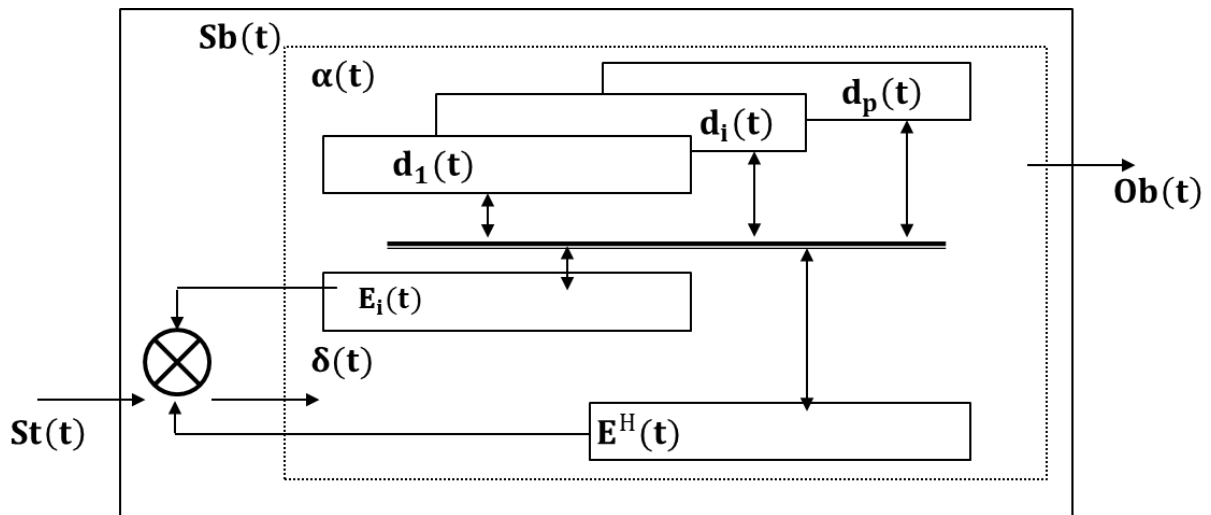


Рис. 1.13. Схема управления

В дальнейшем будем исходить из того, что в основе комбинаторики действий в $\alpha(t)$ лежит специальный тип управления (рис. 1.13), использующий опыт в выявлении управляющих рассогласований $\delta(t)$ и их отработке. Более того, будем полагать, что рассогласование $\delta(t)$ формируется в процессах сравнения опыта $E^H(t)$, который необходим для выполнения $D(t)$ и опыта $E_i(t)$, которым субъект деятельности $Sb_i(t)$ обладает.

Так как опыт $E_i(t)$ дискретен (структурирован в единицах типа «условный деятельностный рефлекс»), то для реализации такого управления необходимы специальные описательно-оценочные коды и действия, ориентированные на эвристику «применить подходящую единицу опыта», то есть ориентированные на деятельность по «прецедентам». Рассогласование $\delta(t)$ формируется в результате сравнения нормативного описания единицы из $E_i(t)$ и ситуативного описания опыта $E^H(t)$, а результат сравнения оценивается и используется в управляющих целях. Если для реализации $D(t)$ достаточна репродукция единицы из опыта $E_i(t)$, то на первый план в активности $\alpha(t)$ выходит поиск в $E_i(t)$, критерием завершённости которого служит значение $\delta(t) = 0$. На момент времени t_0 это значение определяется кодом $E^H(t_0)$.

Для нового деятельностного реагирования $D(t)$ отношения между $E^H(t)$ и $E_i(t)$ сложнее. В общем случае в формирование рассогласования $\delta(t)$ и его отработку включаются все феномены мышления. На первый план в активности $\alpha(t)$ выходит то, что получило название «рассуждение».

1.2.4. Человеко-компьютерное взаимодействие

Проблема естественности, удобства и рациональности действий человека сформулирована (в п. 1.1) для человеко-компьютерной

деятельности (обозначим $D^K(t)$), что требует раскрыть отношения между человеком и компьютером более детально. Начнём с расширения понятий «субъект» и «объект» деятельности.

Ни что не мешает включить инструментарий (любого типа) в состав субъекта, считая инструментарий «продолжением» тела или органов чувств. Если такое «продолжение» имеет функциональное подобие с активностью человека (например, компьютерная система), то ему можно приписать статус «псевдосубъекта» и считать объединение субъекта и псевдосубъекта как составной субъект. В этом случае правомерно использование схемы деятельности, представленной на рис. 1.14, в которой компьютерная составляющая $D^K(t)$ обозначена как $Sb^K(t)$, а действительный субъект деятельности как $Sb^Ч(t)$.

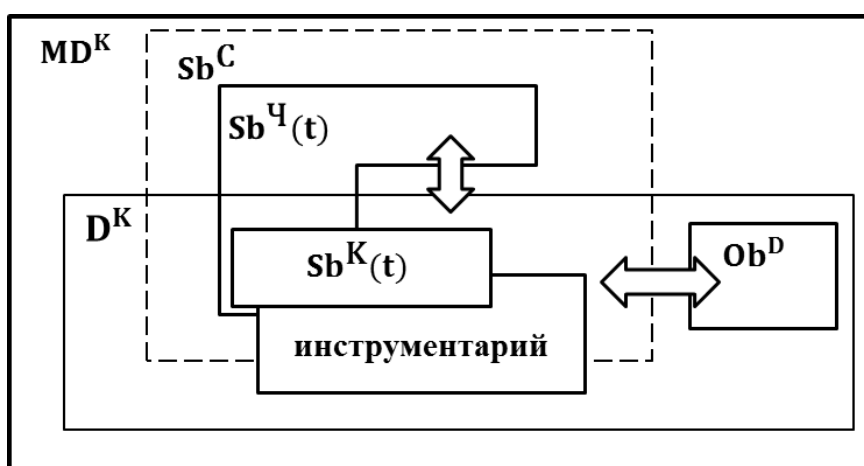


Рис. 1.14. Схема человеко-компьютерной деятельности

На практике применяют и очеловечивание объекта деятельности $Ob(t)$, приписывая ему статус субъективированного объекта $Ob^C(t)$. Такое приписывание наделяет $Ob^C(t)$ человеческими свойствами и позволяет рассматривать объединение $Ob^C(t)$ и $Sb^C(t)$ как составной субъект, а отношения между $Ob^C(t)$ и $Sb^C(t)$ как коммуникацию. Приписывание особенно полезно для объектов, материализованных в компьютерных средах. Общий случай деятельностных отношений человека и компьютера раскрывает схема, представленная на рис. 1.15. В схеме составной субъект Sb , включающий составляющую $Sb^Ч$ и субъективированную компьютерную составляющую Sb^K , совмещает роли субъекта Sb^D и метасубъекта Sb^{MD} . Предполагается, что субъект Sb по ходу $D^K(t)$ может вступать во взаимодействие с субъективированной объектной средой $Ob^C = \{Ob^C_j\}$, вложенной в тот же компьютер, что и Sb^K .

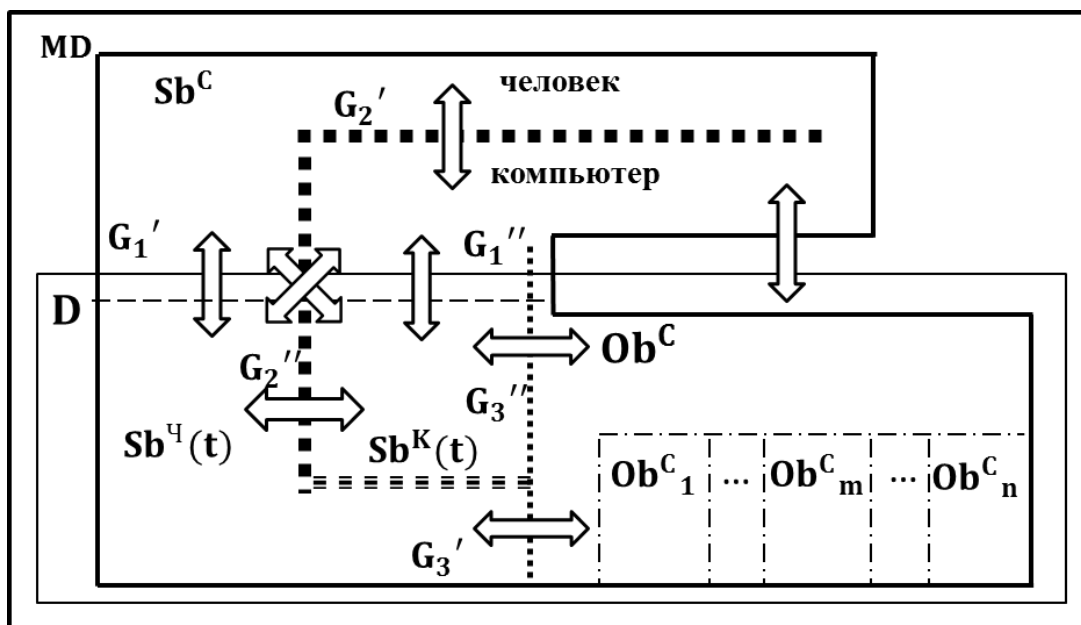


Рис. 1.15. Обобщенная субъект-объектная схема

В этом случае в составном субъекте Sb^C , включающем и субъективированную объектную среду, различимы следующие компоненты $Sb^C(t)$, $Sb^K(t)$ и Ob^C , состоящий из множества $\{Ob^C_m\}$. Для выделения такого компонентного состава на рис. 1.15 используют границы G_1 , G_2 и G_3 , указывающие на тот факт, что для граничащих частей схемы должно быть определено понятие «взаимодействие» в том или ином варианте его употребления (управление, разделение труда, общение или некоторая совокупность таких действий).

Представим переходы (они помечены на рис. 1.15 фигурными стрелками) между границами G_1 , G_2 и G_3 детальнее, сначала для случая, когда за $Sb^C(t)$ стоит индивид, а все компьютерные составляющие вложены в один компьютер.

Граница G_1 разделяет в $Sb^C(t)$ и $Sb^K(t)$ метасубъектную часть от субъектной части (участки границ G_1' и G_1''). Участок границы G_1' локализован в мозговых структурах. Переходы через этот участок естественны. Они обслуживают взаимодействие в наиболее общем значении этого слова, обеспечивающем процессы активности $\alpha(t)$. Участок границы G_1'' , разделяющий одну компьютерную составляющую от другой, должен моделировать G_1' , причём модель G_1'' должна быть реализована как программный комплекс, в котором на человека ложится не более чем выбор из определённого множества альтернатив.

Будем полагать, что у границ G_1' и G_1'' есть общая часть, через которую метасубъектная составляющая $Sb^C(t)$ взаимодействует с субъектной составляющей $Sb^K(t)$ и метасубъектная составляющая $Sb^K(t)$ взаимодействует с субъектной составляющей $Sb^C(t)$.

Граница G_2 также состоит из двух участков, каждый из которых разделяет человека и компьютер. Через первый участок происходит взаимодействие двух метасубъектных составляющих, а через второй участок – взаимодействие двух субъектных составляющих. Эти взаимодействия подобны взаимодействиям через общую часть границ G'_1 и G''_1 . Через участок границы G'_3 взаимодействуют субъективированная часть $Sb^C(t)$ и объективированный субъект (то есть взаимодействуют человек и компьютер), а через участок G''_3 взаимодействуют одна компьютерная составляющая с другой (что подобно переходам через границу G''_1).

Представленная на рис. 1.15 структуризация открывает возможность строить «трассы» взаимодействия, например « $Sb^C(t) - Sb^K(t) - Ob^C(t)$ » или « $Sb^C(t) - Sb^K(t) - Sb^C(t)$ », нагружая трассы определённым содержанием. Это позволяет моделировать взаимодействия индивида с самим собой через «компьютерного посредника».

Расширим содержание, которое стоит за обозначениями на схеме рис. 1.15, допустив, что $Sb^C(t)$ обозначает коллектив индивидов, а компьютерные составляющие реализованы на компьютерной сети. Такое расширение позволяет строить дополнительные типы «трасс взаимодействия» в человеко-компьютерных средах, например « $Sb^C_i(t) - Sb^K_j(t) - Sb^C_k(t)$ ».

Таким образом, правомерны следующие интерпретации (и реализации) отношений между человеком и компьютером в человеко-компьютерной деятельности:

- компьютер является не более, чем инструментом, который используется субъектом деятельности (человеком) для усиления его активности;
- компьютер выполняет функции технического посредника между индивидами в их коллективной деятельности;
- компьютерная составляющая $D(t)$ выполняет функции псевдосубъекта (или составного псевдосубъекта), образующего вместе с человеком (или группой лиц) «рабочую группу»;
- компьютер выполняет функции интеллектуального посредника между индивидами (или между индивидом и им самим).

Независимо от функционального назначения взаимодействие между человеком и компьютером осуществляется на основе определённого информационно-лингвистического обеспечения. В дальнейшем будем исходить из того, что в основе любого акта взаимодействия между человеком и компьютером лежит обмен текстовой T_i , наблюдаемой на соответствующем участке границ G_1 , G_2 или G_3 .

В этом плане любой участок границы может быть представлен текстами, «следы» которых «изображаются» на нём с помощью используемого на участке G_i языка. Среди этих «следов» особо важны следы на экране дисплея, изображённые на естественно-профессиональном языке. При переходе текстов с границы на границу происходят их преобразования, подобные машинному переводу с одного языка на другой.

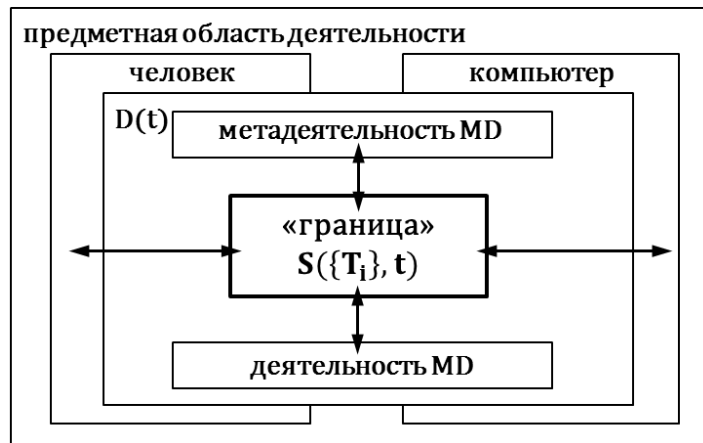


Рис. 1.16. Текстовые «следы» взаимодействия

В конкретной деятельности единицы из множества $\{T_i\}$ образуют систему $S(\{T_i\}, t)$ от характеристик которой существенно зависят как эффективность $D(t)$, так и её результативность (рис. 1.16).

1.3. Предметная область «Человеко-компьютерная диалогика»

1.3.1. Искусственный интеллект

К искусственному интеллекту (ИИ) относят методы и средства, обеспечивающие решение интеллектуальных задач в человеко-компьютерных средах. Задача считается интеллектуальной, если её решение априори неизвестно, а значит поиск механизмов решения и его построение невозможны без привлечения интеллектуальных действий. Наиболее общим подходом к раскрытию содержания, которое вкладывается в понятие «искусственный интеллект», следует признать деятельностный подход, раскрывающий «искусственный интеллект» как предметную область деятельности. Такой подход требует, в первую очередь, указать множество объектов деятельности, определяющих сущность предметной области «искусственного интеллекта».

В пособии эта сущность связана с опытом индивидов, коллективным опытом и моделями опыта как инвариантными к типам деятельности, так и имитирующими опыт в деятельности специального типа. Интеллектуальные действия неотделимы от опыта, дополнительные к нему

как процедурная составляющая, которая отвечает за доступ к опыту, его развитие, модельные представления и использование. Интеллектуальные действия расширяют множество объектов, которые исследуются, усиливаются и моделируются в «искусственном интеллекте». Первоначально работы в «искусственном интеллекте» были нацелены на построение компьютерной модели человека, имитирующей его интеллектуальную активность:

- в процессах решения задач и принятии решений;
- в общении, в частности в переводе текстов с одного естественного языка на другой;
- в распознавании образов и анализе сцен.

По ходу овладения искусственным интеллектом (как освоенными методами и средствами) акцент на практическую роль ИИ смещался от его понимания как самостоятельной интеллектуальной сущности (не зависящей в процессах оперативного использования от человека) к пониманию ИИ как помощника человеку в его интеллектуальных действиях.

Авторы являются убеждёнными сторонниками второй ценностной ориентации и связанной с ней системой мотивов. В дальнейшем будем полагать, что выявление и конструктивное овладение интеллектуальными действиями с целью построения компьютерных моделей опыта интеллектуальных действий, повышающих «коэффициент полезного действия» интеллектуальной активности человека, приводит к классу задач, которые определяют сущность искусственного интеллекта как предметной области.

1.3.2. Человеко-компьютерный диалог

Человеко-компьютерное взаимодействие является необходимой составляющей человеко-компьютерной деятельности $D^K(t)$. Из-за участия в нём человека, такое взаимодействие представляет собой естественно-искусственное образование, в котором активность $\alpha(t)$ и опыт $E_i(t)$ способны проявляться в полном объёме. Только часть $\alpha(t)$ и $E_i(t)$, причём в формах подходящих моделей, может быть вложена в компьютерную составляющую взаимодействия.

Человеко-компьютерное взаимодействие (должно быть) управляемо из-за влияния на него внешних (в том числе и искусственных) факторов и за счёт включения в его состав искусственных составляющих. Процесс взаимодействия можно строить (и понимать) как «коллективную» деятельность специального типа.

Одной из принципиальных целей взаимодействия является управление процессами в $D^K(t)$. Авторов интересует в основном эта цель,

и всё последующее содержание пособия развёртывается в контексте проблем управления. Более того, в пособии представлены результаты исследований управления, в котором выявляется и отрабатывается рассогласование $\delta(t)$ между опытом $E^H(t)$, который субъекту необходим, и его личным опытом $E_i(t)$.

Авторы понимают рассогласование $\delta(t)$ как то, что следует называть «вопросом», а отработку рассогласования как то, что следует называть «ответом». Такое понимание приводит к интерпретации процесса управления (через рассогласование $\delta(t)$) как вопросно-ответного процесса, реализация которого распределена между человеком и компьютером. В распределении работ целесообразно рассчитывать на «активность» компьютерного псевдосубъекта, который «способен» проявлять инициативу в вопросно-ответном процессе управления, то есть включаться в процесс управления, как с «вопросами», так и с «ответами».

Таким образом, в реализациях вопросно-ответного процесса управления может наблюдаться естественно-искусственное образование человеко-компьютерного диалога, знаковая модель которого в форме системы текстовых единиц $S(\{T_i\}, t)$ выводится на экран дисплея (и открыта для использования участниками диалога в оперативных целях).

Представленное выше понимание рассогласования $\delta(t)$ и соответствующая ему установка (на исследования и разработки) позволяют расширить значение общепринятого (специалистами) понятия «человеко-компьютерный диалог», включив в него новые варианты употребления. Наиболее интересными и важными вариантами являются те, которые раскрывают взаимодействия (в процессах управления) человека с самим собой или с другим человеком. Человеко-компьютерный диалог в таких взаимодействиях было бы более правомерно назвать «человеко-человеко-компьютерным».

Связав с взаимодействием в формах вопросно-ответного управления специальный тип деятельности, раскроем этот тип с деятельностных позиций. В первую очередь определимся с объектной сущностью «вопросно-ответного управления» как предметной области.

К основным объектам отнесём вопросы (обозначим Q), ответы (обозначим A), вопросно-ответные образования (обозначим QA). Каждый из конкретных экземпляров Q , A и QA является естественно-искусственным образованием, локализованным в пространстве и времени (учтём это обозначениями $Q(t)$, $A(t)$ и $QA(t)$). Вопросно-ответное управление как деятельность (обозначим $D^{QA}(t)$) включает метадеятельностную составляющую, в которой используются подходящие модели $Q^M_i(t)$, $A^M_j(t)$ и $QA^M_k(t)$ для объектов $Q(t)$, $A(t)$ и $QA(t)$, среди которых наиболее полезны знаковые модели.

Контрольные вопросы

1. Как формулируется основная проблема человеко-компьютерной деятельности?
2. Какие составляющие включает в себя деятельность?
3. В чём заключается роль основных составляющих деятельности?
4. Что подразумевается под опытом деятельности?
5. Какова структура условно-деятельностного рефлекса?
6. Какие положительные эффекты даёт переход умения в класс навыков?
7. Как осуществляется исполнение рефлекса?
8. Как соотносятся феномен «восприятие» и методы распознавания образов?
9. Как формируется опыт интеллектуальных действий?
10. Что такое «управляющие рассогласования»?
11. По каким направлениям строятся «отношения» человека с компьютером в человеко-компьютерной деятельности?
12. Что представляет собой рассогласование между опытом, который субъекту необходим, и его личным опытом?

ГЛАВА 2. ПРОБЛЕМЫ ЧЕЛОВЕКО-КОМПЬЮТЕРНОЙ ДЕЯТЕЛЬНОСТИ

2.1. Особенности проектирования автоматизированных систем

Развитие и постоянное совершенствование социокиберфизической реальности (SCP-reality) полезно понимать как искусственный эволюционный процесс, который тесно переплетается с естественными процессами на Земле, в том числе с процессами жизни. В этом переплетении искусственные и естественные процессы влияют друг на друга так, что это приводит к важным событиям $\{e_i\}$, оценки которых полезны для будущего.

В частности, подобные события соответствуют ситуациям завершения работы проектировщиков над проектами с системами, интенсивно использующими программное обеспечение. Реальность указанного вида проектирования и характеристика его результатов показали, что существует проблема достижения прогнозируемого успеха в данном виде деятельности.

Прежде чем описывать эту проблему, необходимо представить названный класс систем. Ниже, в тексте, мы будем ориентироваться на следующее определение «система, интенсивно использующая программное обеспечение (иными словами, автоматизированная система) – это система, в которой программное обеспечение представляет собой значительный сегмент в любом из следующих моментов: функциональность системы, стоимость системы, риск разработки системы, время разработки» (Software Intensive systems, 2006).

В целом данное определение допускает, что автоматизированная система может включать социальные, физические и программные компоненты, взаимодействующие друг с другом. Таким образом, можно предположить, что сейчас и в будущем автоматизированные системы (АС) являются и будут основным компонентом социокиберфизической реальности.

Данная книга посвящена повышению естественности человеко-компьютерного взаимодействия, современное состояние которой наиболее полно раскрывает проблемы проектирования АС. Вот почему ниже текст книги будет явно или неявно касаться такого рода систем. Кроме того, системы программного обеспечения мы будем квалифицировать как подкласс АС, и это позволит нам использовать термин «система» как для АС, так и для системы программного обеспечения, когда это не принципиально.

В частности, наш выбор проектирования АС как источника используемых искусственных и естественных процессов и проблем в их переплетении был обусловлен следующими причинами:

1. Разработка АС требует богатого опыта разработки программного обеспечения, а также опыта в различных предметных областях.

2. Часть накопленного опыта проектирования отвечает за поддержание непрерывного совершенствования процессов, рабочей силы и продукции.

3. Существует богатейшая статистика успехов и неудач реализованных проектов с программным обеспечением.

4. Многочисленные и разнообразные разработки различных АС также привели к созданию зрелых технологий, рабочие процессы которых помогают творчески решать задачи различных типов.

5. В процессах жизненного цикла проектировщики создают и используют различные формы человеко-компьютерного взаимодействия, которые поддерживают персональную и совместную работу проектировщиков в режиме реального времени.

Возвращаясь к проблемам достижения успеха в проектировании систем, полезно проанализировать отчёты, регулярно публикуемые компанией Standish Group (Reports, 2016) [54]. Каждый из этих отчётов оценивает определённый набор завершённых проектов на предмет их успешности. В отчёте оцениваемые проекты распределяются по трём классам со следующими характеристиками: «успешный», «неудачный» и «проблемный».

Первый отчёт, который получил название «Chaos Report», был опубликован в 1994 году. В названии этого отчёта прилагательное «chaos» (хаотичный) отражает крайне низкую степень успеха в проектировании программных систем, поскольку только 16% проектов были оценены как успешные. Другие отчёты так же зарегистрировали недопустимо низкие результаты, что продемонстрировано на рис. 2.1.

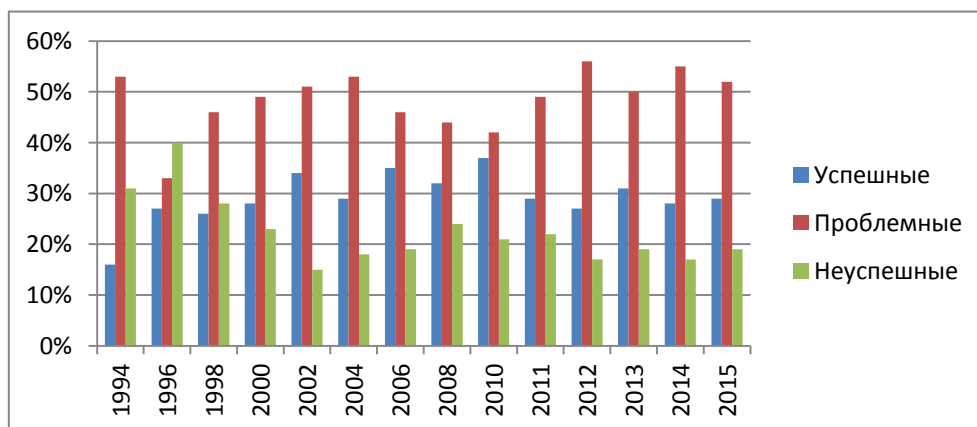


Рис. 2.1. Оценки успешности проектов по годам

Во всех записях до 2015 года их разработчики использовали следующие критерии распределения классов: все обещанные требования («onTarget») в пределах запланированного времени («onTime») в рамках выделенного бюджета («onBudget»).

Следует отметить, что данный набор критериев (называемый железным треугольником) понимается и используется в качестве классической основы для оценки процессов управления в разработках систем. Таким образом, отмеченные отчёты Standish Group отразили успешность только со стороны управления проектом.

В своём употреблении слово «успех» понимается как многоаспектное понятие, и эта его особенность должна найти соответствующее отображение в разработке систем для измерения их успешности. Поэтому для адекватной оценки успешности той или иной системы заинтересованные стороны должны использовать характеристики жизненного цикла этой системы.

Поэтому существует множество публикаций, авторы которых предлагают другие варианты наборов критериев. Это было вызвано другими интерпретациями успеха и неудачи. В качестве примера, отметим следующие версии интерпретаций:

1. В статье (Wasmun 1993), которая была опубликована до 1994 года, её авторы интерпретировали успех как многомерную конструкцию, интегрирующую: качество системы (измерение разработанной системы), качество информации (измерение выхода системы), удовлетворённость пользователей, использование системы, индивидуальное воздействие (влияние на поведение) и организационное воздействие пользователя (влияние организационной производительности).

2. В 2008 году авторы статьи (El Emam and Koru, 2008) [66] предложили набор критериев, включающий три аспекта: *успех управления проектом* (соблюдение сроков, бюджета, удовлетворённость спонсоров, руководящей группы, команды проекта, клиентов/пользователей и заинтересованных сторон), *технический успех* (удовлетворённость клиентов/пользователей и заинтересованных сторон, внедрение системы, соблюдение требований, качество системы, использование системы) и *деловой успех* (непрерывность бизнеса, достижение коммерческих целей, получение выгоды).

Также в некоторых работах, как, например, (Jørgensen and Moløkken, 2006) [74] и (Glass 2005) [68], критерии, используемые Standish Group, были раскритикованы и оценивались как не отражающие реальность системы с программным обеспечением.

С учётом критики и совершенствования используемого подхода компания Standish Group в 2015 году «переопределила успешность проекта как соблюдение сроков и бюджета с удовлетворительным результатом,

который включает критерии достижения цели (процент выполненных требований) и удовлетворённости (от очень высокой до очень низкой), ценности (от очень высокой до очень низкой), и выполнения стратегических корпоративных задач (от точного до частичного)» (Chaos, 2015).

Расширение набора критериев приводит к изменению распределения картины по классам, указанным на рис. 2.1. В эту картину Standish Group включила ортогональные меры, соответствующие ценностному представлению о характеристиках анализируемых проектов. Кроме того, в статье (Haze, 2015) [70] было продемонстрировано сравнение статистики с точки зрения успешности и значимости. Например, таблица 2.1 показывает ту часть этого сравнения, где первая строка подчёркивает различный подход к оценкам успешности.

Таблица 2.1. Сопоставление ортогональных взглядов на успешность

Характеристика	По успешности	По значимости
Измерения	Тройные ограничения	Польза для компании
Диапазон измерений	Индивидуальные проекты	Портфолио проектов
Сфера	Управление ИТ	Управление бизнесом
Корпоративная культура	Избежание рисков	Толерантность к рискам
Руководство	Управляющие проектом	Владельцы проекта
Сопровождение	Офис управления портфелем проектов	Куратор проекта
Заинтересованные стороны	Отсутствуют, не участвуют	Участвуют и заинтересованы
Процесс управления	Тяжёлый	Лёгкий
При соблюдении требований	Отчётность	Результат
Общая направленность	Процесс	Люди
Аварийная кнопка	Рекомендуется	Не рекомендуется
Бюджетный процесс	Для каждого проекта	Портфолио (Breadbasket)
Выпуск продукта	Долгий	Быстрый
Использование инструментов управления проектами	Обильное и зависимое	Лёгкое и независимое
Типы приложений	Статичное	Инновационное

Даже часть этой таблицы отражает разнообразие оценок успешности с позиций процесса разработки и использования завершённых проектов. Такое разнообразие может привести к «неудачным успехам» и «успешным неудачам», что указывает на необходимость тщательного взвешивания используемых критериев оценки проекта. Тем не менее, до сих пор в проектировании систем, независимо от выбранных критериев, существует проблема успешности.

2.1.1. Факторы успеха и неудачи

Как было сказано выше, статистика, накопленная в базе Standish Group, является важным источником информации для её использования в исследованиях, направленных на поиск путей повышения степени успешности проектирования систем. Центральное место в этой информации занимают выявленные факторы успехов и неудач. Каждый отчёт не только раскрывает перечни этих факторов, но и упорядочивает их по степени их влияния в позитивном или негативном ключе. Объединение этой информации в таблицы по каждому критерию подсказывает возможные направления исследования проблемы успешности. Одна из этих таблиц – таблица 2.2, которая показывает факторы успешности с оценкой до десяти.

Таблица 2.2 Рейтинг факторов успешности (CHAOS Report)

Фактор	1994	1998	2000	2002	2006	2008	2010	2012
Вовлечение пользователя	1	1	2	1	1	1	2	2
Поддержка со стороны исполнительного руководства	2	2	1	2	2	2	1	1
Чёткое изложение требований	3	6	7					
Правильное планирование	4	8						
Реалистичные ожидания	5							
Дробление этапов проекта	6	5						
Компетентная команда	7	7		10	8	8	8	4
Право собственности	8	9						
Чёткое видение и цели	9	3	4	4	3	3	3	7
Трудолюбивая целеустремлённая команда	10							
Управление проектом		4	3	3	6	7	7	5
Сокращение масштабов			5	5				
Стандартные инструменты и инфраструктура			6	7	10	10	10	10
Фактор	1994	1998	2000	2002	2006	2008	2010	2012
Формальная методология			8	8	9			
Достоверные оценки			9	9				
Фактор	1994	1998	2000	2002	2006	2008	2010	2012
Гибкий процесс работы с требованиями				6	5	6		
Оптимизация масштабов / Оптимизация					4	5	5	3
Управление финансами					7		4	4
Эмоциональная зрелость						4		6
Реализация						9	9	9
Другое		10	10					

Факторы, обозначенные в таблице 2.1, представляют собой условия проектирования, нарушение которых может привести к неудаче. В отчётах каждый из этих факторов успешности имеет определение, которое помогает организовать и реализовать соответствующие условия для выполнения проекта. Отчёты также содержат полезную информацию о факторах неудачи. Так, оба этих источника информации могут быть использованы проектировщиками для рационального управления проектами.

Для нового набора критериев, используемых в отчётах Standish Group после их переопределения понятия «успех» (2012 год) и для других наборов критериев также возможно построение соответствующих таблиц, помогающих перечислить все виды факторов успеха и неудачи, названных в отчётах и результатах исследований. Как говорилось выше, в дополнение к другим факторам эти таблицы будут включать описания условий, которые будут влиять на достижение удовлетворённости заинтересованных сторон.

Таким образом, совокупность выявленных факторов явно или неявно указывает на возможные направления исследований, которые могут привести к положительным результатам в достижении успеха при проектировании систем. Одним из таких направлений является поиск инноваций в предметной области ЧКВ.

Например, таблица 2.1 включает фактор «Вовлечение пользователя», который находится на верхних позициях. В целом данный фактор отражает активность потенциальных пользователей, взаимодействующих с проектировщиками и состояниями проектируемой системы в выбранных моментах её жизненного цикла. Результаты этой деятельности существенно зависят от того, как она организована, какие методы и средства взаимодействия используются и от ответов на другие вопросы, которые могут быть дополнительно сформулированы и исследованы.

Факторы «Чёткое видение и цели» и «Компетентная команда» также неявно связаны с человеко-компьютерным взаимодействием, инновации в которых могут способствовать повышению успешности разработки систем. Первый из этих факторов указывает на достижение необходимого понимания командой проектировщиков на ранних этапах их работы с концепцией проекта, который должен быть разработан. Например, в методологии «Rational Unified Process» эти работы проводятся в рамках интерактивных рабочих процессов «бизнес-моделирование», которые приводят к созданию очень важного артефакта «видение». В этой и других технологиях компетентные сотрудники должны использовать интерактивные рабочие процессы, сопровождающие все стадии и этапы разработки систем.

Как было сказано выше, оценивая успех, необходимо учитывать удовлетворённость заинтересованных сторон всех категорий (заказчиков, пользователей, членов команды, менеджеров и других). Большая часть этих оценок должна основываться на результатах взаимодействия заинтересованных сторон с процессами и продуктами работ в жизненном цикле систем.

Обобщая, отметим, что область разработки систем обладает богатым потенциалом для проведения исследований в человеко-компьютерном взаимодействии, направленных не только на повышение успешности в данной предметной области, но и на повышение эффективности человеко-компьютерной деятельности. Именно поэтому в этом пособии эта область была выбрана для представления SCP-реальности в поиске новинок в человеко-компьютерном взаимодействии.

2.2. Проблема сложности

2.2.1. Сложность и её преодоление

Крайне низкая степень успешности разработок систем является важной причиной поиска новых подходов к проектированию в данной предметной области. Один из таких подходов может быть связан с учётом не только программной интенсивности, но и интенсивного использования знаний и опыта в условиях человекоёмких систем. Дополнительные интенсивности непосредственно связаны с проектированием систем, поскольку в такой работе команда проектировщиков должна творчески использовать рабочие процессы со сложными наборами инструментов, которые также относятся к классу систем (АС). Интегральным результатом этих интенсивностей является высокая сложность среды, с которой проектировщик вынужден взаимодействовать.

Существует много определений сложности, что обусловлено многочисленными видами её проявлений, но, в любом случае, сложность характеризует взгляд на систему с множеством элементов и форм отношений между этими элементами. В общем объяснении сложность выражает меру, которая оценивает трудность для человека, взаимодействующего с представлением соответствующей системы или её части с определёнными целями. Одной из важнейших среди этих целей является достижение необходимого понимания.

Система или какой-либо её компонент является сложным, если проектировщик (взаимодействующий с системой) не имеет достаточных ресурсов для достижения необходимого уровня понимания или достижения других запланированных целей. В большинстве случаев сложность или простота является функцией трёх переменных – времени,

точности (диапазона) и объёма информации. Сложная система порождает больше трудностей, чем простая.

Достаточно часто для оценки степени сложности системы применяются различные интерпретации меры Колмогорова (Li M. и Vitani, 2008) [78]. Эта мера связана с «минимальной длиной программы P , обеспечивающей построение системы S из её исходного описания D ». Различия в интерпретациях обычно обусловлены особенностями системы S , а также тем, какое содержание связано с конструктами P и D и как определяется это содержание. В любой интерпретации конструкт P отражает деятельность разработчиков, которые используют соответствующие технологии и инструментарий в разработке системы. Таким образом, процессу разработки системы соответствует P -программа, раскрывающая этот процесс (P -процесс) с точки зрения «программирования».

При создании системы программы P -типа поэтапно встраиваются в процесс проектирования с использованием определённого «метода программирования M ». Реальность такой работы показывает, что сложность « P -программы» не меньше, чем сложность системы в любом её используемом состоянии. Более того, любое использование M -программирования, обеспечивающее построение P -программы, должно строиться на основе того же начального описания D системы S . Это может быть представлено следующей последовательностью: $D \rightarrow M \rightarrow P \rightarrow S$.

Названные отношения между D , M , P могут быть использованы проектировщиками для разделения процесса проектирования на этапы $[D(t_0) \rightarrow M_1 \rightarrow P_1 \rightarrow S(t_1)]$, $[D(t_1) \rightarrow M_2 \rightarrow P_2 \rightarrow S(t_2)]$, ..., $[D(t_i) \rightarrow M_{i+1} \rightarrow P_{i+1} \rightarrow S(t_{i+1})]$, ..., $[D(t_{n-1}) \rightarrow M_n \rightarrow P_n \rightarrow S(t_n)]$, где множество $\{S(t_i)\}$ содержит состояния создаваемой системы.

Разделение процесса проектирования на этапы является типичным решением в любой современной технологии, предполагающей создание систем. В разных технологиях такой подход используется в разных формах для разных целей, в том числе, для снижения степени сложности. Таким образом, данный подход позволяет снизить сложность взаимодействия с системой в любом её состоянии $S(t_i)$.

Разделение на этапы, каждый из которых приводит к созданию соответствующего состояния $S(t_i)$, может быть детализировано, поскольку система в любом состоянии состоит из связанных частей и так далее до компонентов $\{C_j(t_i)\}$, которые соответствуют задачам проекта $\{Z_j(t_i)\}$, распределенным между членами команды. Иными словами, сложность любого компонента $\{C_j(t_i)\}$ оценивается (в первую очередь) проектировщиком, отвечающим за решение соответствующей задачи

$Z_j(t_i)$. Этот случай может быть представлен последовательностью отображений

$$DC_j(t_{i-1}) \rightarrow M_{ij} \rightarrow P_{ij} \rightarrow C(t_i), \quad (2.1)$$

где $DC_j(t_{i-1})$ – это первоначальное описание компонента $C_j(t_i)$. Эта последовательность дана только для того, чтобы показать, что подход Колмогорова к оценке сложности применим не только ко всей системе, но и к её компонентам на разных уровнях детализации. Последовательности отображений будут использоваться ниже только в общих рассуждениях.

Следовательно, мера сложности Колмогорова может быть применена к любому этапу и компонентам проектирования системы, включая этапы, на которых проектировщики работают с концептуальными представлениями систем. Однако отметим, что до сих пор точка зрения программирования на деятельность проектировщика инструментально не поддерживалась на ранних стадиях проектирования.

Вернёмся к значению слова «сложность», для того чтобы раскрыть особенность его понимания, которая будет использована далее, и приведём следующие детали его содержания:

1. Для полезного управления сложностью проектирования системы вопрос об измеряемой величине сложности не столь существен. Достаточно представления значения, которое помогает ответить на вопрос: «Как проектировщики могут уменьшить сложность?». Когда проектировщики используют меру Колмогорова, сложность уменьшается разделением М- и Р-программ на части.

2. Сложность системы является объективной характеристикой, значение которой во многом зависит от сложности построения системы, представленной программными моделями таким образом, чтобы облегчить понимание, исследование и модификацию. Основная цель таких моделей программирования (таких как Р, Р-модели) – освоить сложность и попытаться уменьшить её без попыток измерения.

3. Р-программы, которые будут рассмотрены ниже, представляют собой модели (Р-модели) коллективной человеко-компьютерной деятельности, ориентированные на решение проектных задач и их композиций в процессе концептуального проектирования системы. На данном этапе, за счёт построения и выполнения соответствующих Р-программ, команда проектировщиков создаёт концептуальный проект системы. Выбор этапа концептуального проектирования обусловлен тем, что без освоения проектировщиками сложности на данном этапе нет оснований полагать, что разработка системы будет успешной.

4. Освоение сложности основано на опыте команды проектировщиков и используемых моделях, которые должны быть представлены таким

образом, чтобы облегчить построение Р-моделей и их использование. Для повышения эффективности концептуального проектирования в достижении успеха необходимо постоянно совершенствовать коллективный опыт и систему его моделей (базу опыта). Нынешнее состояние коллективного опыта и его модели будут называться далее как доступный опыт.

5. Особенно важная часть имеющегося опыта связана с технологией разработки системы и её освоением командой проектировщиков. Использование применяемой технологии можно интерпретировать как выполнение соответствующих технологических программ (Т-программ), поддерживающих создание системы. Любая из таких Т-программ имеет определённую структуру, которую можно извлечь из технологических руководств, что помогает проектировщику понять технологию на уровне, достаточном для её использования. Отметим, что Т-программа, построенная проектировщиками для создания определённой системы, занимает центральное место в соответствующей Р-программе.

6. Приведённые выше рассуждения о сложности проектирования систем помогают нам просто уточнить нашу позицию по процессу проектирования с программной точки зрения. С этой точки зрения Р-программы включают поведение, соответствующее нормативным инструкциям, описанным в руководствах по технологии и инструментам. Эти действия выполняются проектировщиками одновременно с другими действиями компьютеризированных рабочих процессов.

7. Алгоритмические схемы рабочих процессов (соответствующие Т-программам) с их детализированными задачами помогают проектировщикам в освоении сложности некоторых процессов при проектировании систем, но не отражают сложности решения задач в предметных областях создаваемых систем. Схемы (модели) рабочих процессов являются не более чем источниками шаблонов, копии которых включаются в реальные процессы разработки систем, схемы которых зачастую существенно отличаются от стереотипных.

8. Для определённой Р-программы проектировщики обычно вынуждены строить «программные» компоненты для решения необходимых задач, которые отражают область соответствующей системы. Проектировщики создают такие компоненты в реальном времени проектирования, когда они используют и развивают доступный опыт.

Системный подход к управляемому разделению Р-процесса на части – не единственный подход, позволяющий управлять сложностью концептуального проектирования систем. Среди других подходов отметим взгляд на Р-процесс с позиции инициативы SEMAT (Software Engineering Methods and Theory), которая была заявлена в 2009 году (Jacobson, 2009) [27]. В настоящее время данная инициатива получила нормативное

обоснование (Jacobson, 2012) [73], что отражено на схеме, показанной на рис. 2.2.

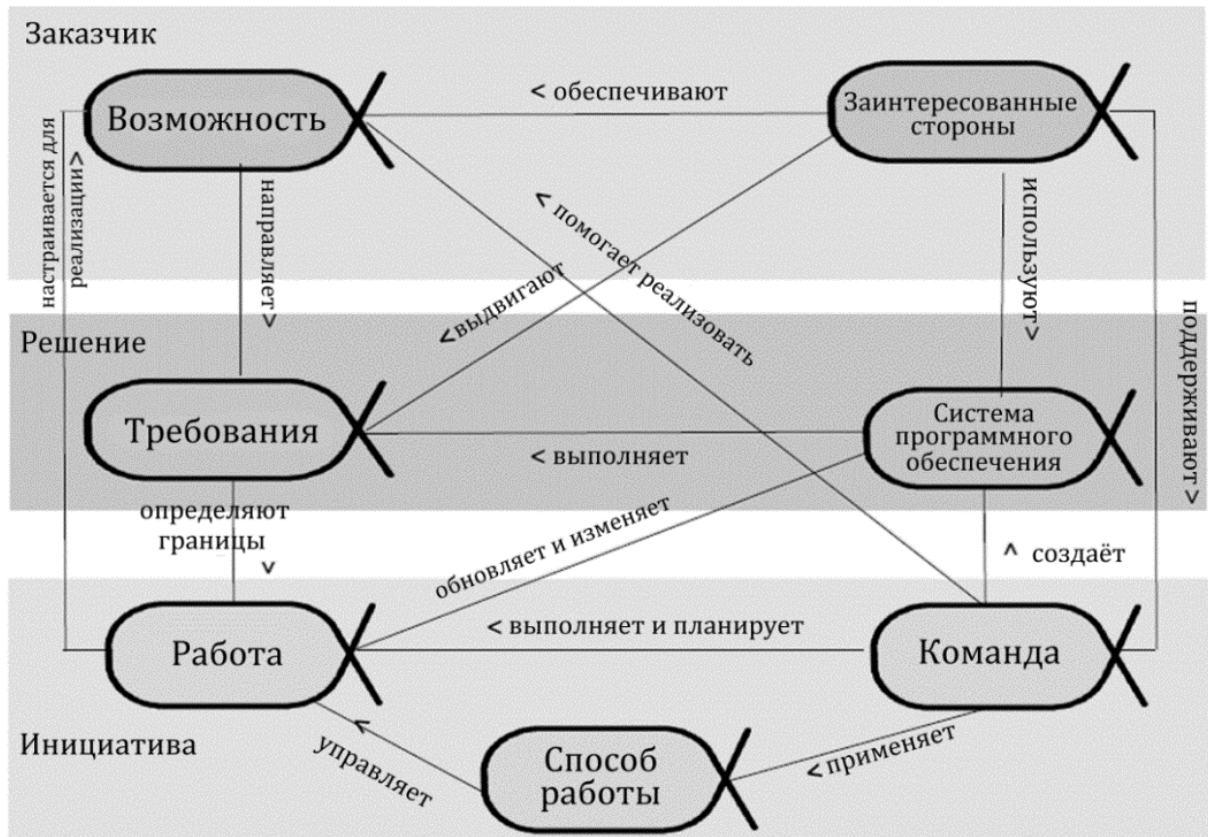


Рис. 2.2. Структура проблемных областей (Jacobson, 2012)

С использованием специализированных понятий «альфы» (Alphas – Abstract-Level Progress Health Attributes), данная структура раскрывает связанную сеть важных аспектов, которые должны найти своё воплощение в разработке программных систем. Альфы предназначены для согласованной оценки состояния соответствующих частей проблемных областей, что помогает, прежде всего, отделить друг от друга «объекты, с которыми предстоит работать».

В общем случае, все эти предметы формируются при проектировании системы и их предварительное и контролируемое разделение способствует снижению сложности для тех лиц, которые будут участвовать в процессе. Кроме того, в подходе SEMAT такой процесс реализуется методом (Way-of-Working), который сочетает ядро SEMAT с лучшими практиками, выбранными проектировщиками для реализации определённой инициативы (рис. 2.3). По определению (Jacobson et al 2012) [73], метод (Way-of-Working) – это «индивидуальный набор практик и инструментов, используемый командой для руководства и поддержки их работы».

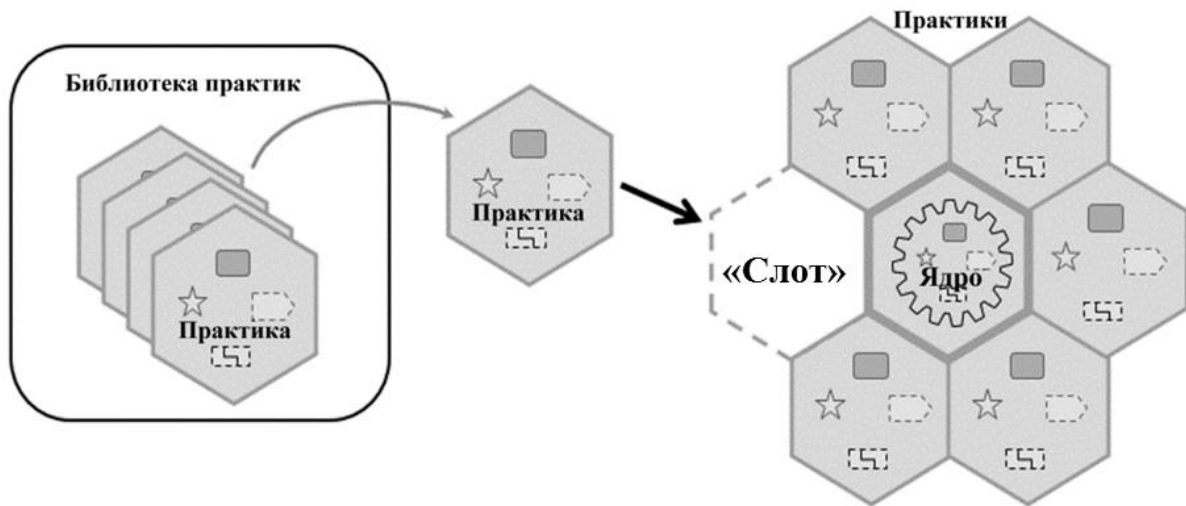


Рис. 2.3. Разработка метода работы

Подход SEMAT не только определяет представленную структуру, но и предлагает богатую библиотеку практик, которая помогает организовать и управлять деятельностью, направленной на создание программного комплекса. В дополнение к объектам, с которыми предстоит работать, нормативные документы определяют набор «дел», «необходимых навыков», включающих компетенцию, и «вспомогательных материалов». Все эти ресурсы доступны и используются в условиях, где они могут быть добавлены с помощью других необходимых ресурсов.

Отметим, что изначально подход SEMAT был разработан для программной инженерии, но его основные идеи уже нашли своё применение в системной инженерии.

Возможность наследования обусловлена схожестью действий разработчика при создании сложных систем. Более того, для команды, применяющей подход SEMAT при разработке программной части системы, возможность становится целесообразной необходимостью.

Возвращаясь к подходу SEMAT, отметим следующее:

1. В настоящее время данный подход находится на стадии профессионального становления и достигнутое им состояние может быть использовано для практического применения, которое, в свою очередь, может служить полезным источником для совершенствования подхода.

2. Альфы ядра и отношения между ними явно указывают на необходимость эффективного взаимодействия как между лицами, участвующими в процессе работы, так и каждого человека с текущим состоянием процесса проектирования и его результатов.

3. Становление подхода SEMAT является ярким примером необходимости переосмысления не только основ программной инженерии, но и основ человеко-компьютерной деятельности, в том числе и человеко-компьютерного взаимодействия.

Таким образом, в данном пособии мы ссылаемся на SEMAT, поскольку это одна из последних попыток найти более рациональные способы профессиональной деятельности в программных разработках, соответствующих сущности человеческой деятельности. Так получилось, что в своём нынешнем состоянии данный подход лишь частично охватывает следующие аспекты:

– Теоретические основы предлагаемого подхода (в глоссарии SEMAT слово «теория» отсутствует);

– Динамика активности и её «механизм» с учётом итерационного процесса и многозадачности;

– Творчески интеллектуальное поведение лиц, использующих доступный опыт при участии в процессах разработки.

Поэтому названные аспекты (и не только в контексте подхода SEMAT) указывают на важные направления совершенствования человеко-компьютерной деятельности. Эти направления приводят к необходимости инноваций в проектировании АС.

Следует также отметить, что кроме SEMAT существует множество других подходов к разработке программных систем и АС. Любой из этих подходов структурирует проблемные области в формах, снижающих сложность. Типичным способом структурирования является использование концептуальных схем, которые могут быть применены в качестве шаблонов для различных целей и, прежде всего, для поддержки понимания.

То, что было сказано о сложности, позволяет сделать следующие выводы:

1. В любом акте взаимодействия с компьютером проектировщик сталкивается с определённой степенью сложности операционной среды, отображаемой на экране монитора.

2. Для любой наблюдаемой ситуации операционной среды её разработчики должны создавать условия, при которых воспринимаемая степень сложности не должна выходить за пределы естественных ограничений разума проектировщика.

3. В тех случаях, когда сложность воспринимаемых условий находится за пределами естественных ограничений разума, было бы полезно обеспечить операционную среду уменьшением сложности.

2.2.2. Технологические аспекты человеко-компьютерных взаимодействий

Как было сказано выше, рациональное управление степенью сложности положительно влияет на эффективность взаимодействия человека с компьютером. Типичным вариантом такого управления

является разделением наблюдаемого комплекса (структуры) на части. При проектировании систем проектировщики взаимодействуют с различного рода комплексами, которые лучше классифицировать, и эта работа может быть выполнена с использованием различных точек зрения. На стратегическом уровне структуры, описанные в предыдущем подразделе, являются полезными источниками для выбора соответствующих точек зрения и структурирования.

Одним из основных принципов подхода SEMAT является целесообразное расширение ядра необходимыми практиками. Такое расширение приводит не только к онтологическим деталям, но и к расширению лексики, которая находит своё контролируемое использование в рассуждениях, требованиях и ограничениях, документах и других артефактах, включая элементы человеко-компьютерного взаимодействия.

Именно поэтому и только гипотетически мы раскрываем ряд сущностей, которые соответствуют таким проблемным областям, как «Решение» и «Инициатива». Раскрывая детали, сначала мы ссылаемся на технологию Rational Unified Process (RUP). Выбор данной технологии обусловлен принятой в ней оценкой того, насколько богата, продумана и внедрена система лучших технологических практик. В деталях мы будем использовать термины глоссария RUP без указания их строгих определений.

Интересная попытка представить RUP в форме записи SEMAT описана в статье (González-Pérez et al., 2014) [69], где авторы применили ядро RUP, которое было объединено с его базовыми практиками, что показано на рис. 2.4.



Рис. 2.4. Имитация RUP

Кроме того, эти практики были распределены авторами между областями «решение» (управление требованиями, использование компонентной архитектуры, визуальная модель программного обеспечения, проверка качества программного обеспечения) и «инициатива» (итеративная разработка программного обеспечения, контроль изменений в программном обеспечении).

Согласно глоссарию SEMAT [60], любая практика – это «повторяемый подход к выполнению чего-либо с конкретной целью. Практика описывает, как обращаться с определённым аспектом разработки программного обеспечения, включая описания всех соответствующих элементов, необходимых для выражения желаемого руководства работой, которое требуется для достижения цели практики. Практику можно определить как совокупность других практик». Данное определение приводит ко второму варианту разделения RUP на практики, который показан на рис. 2.5.



Рис. 2.5. Имитация RUP

Во втором варианте RUP представляет собой производство шести процессов (рабочие процессы, бизнес-моделирование, требования, анализ и проектирование, реализация, тестирование, применение) и трёх вспомогательных процессов (управление проектом, управление конфигурацией и изменениями, среда), которые пересекаются с четырьмя

стадиями (начальная стадия, уточнение, построение и внедрение) работ и их итераций.

Следует отметить, что действия первой композиции переплетаются со стадиями очень сложным образом, в то время как компоненты второго варианта разделяют рабочие процессы на классы и части, которые более чётко демонстрируют программный подход к снижению сложности. Для двух RUP-процессов (Требования и Управление проектами), их технологические структуры схематично показаны на рис. 2.6 (только для демонстрации сложных комплексов технологических работ).

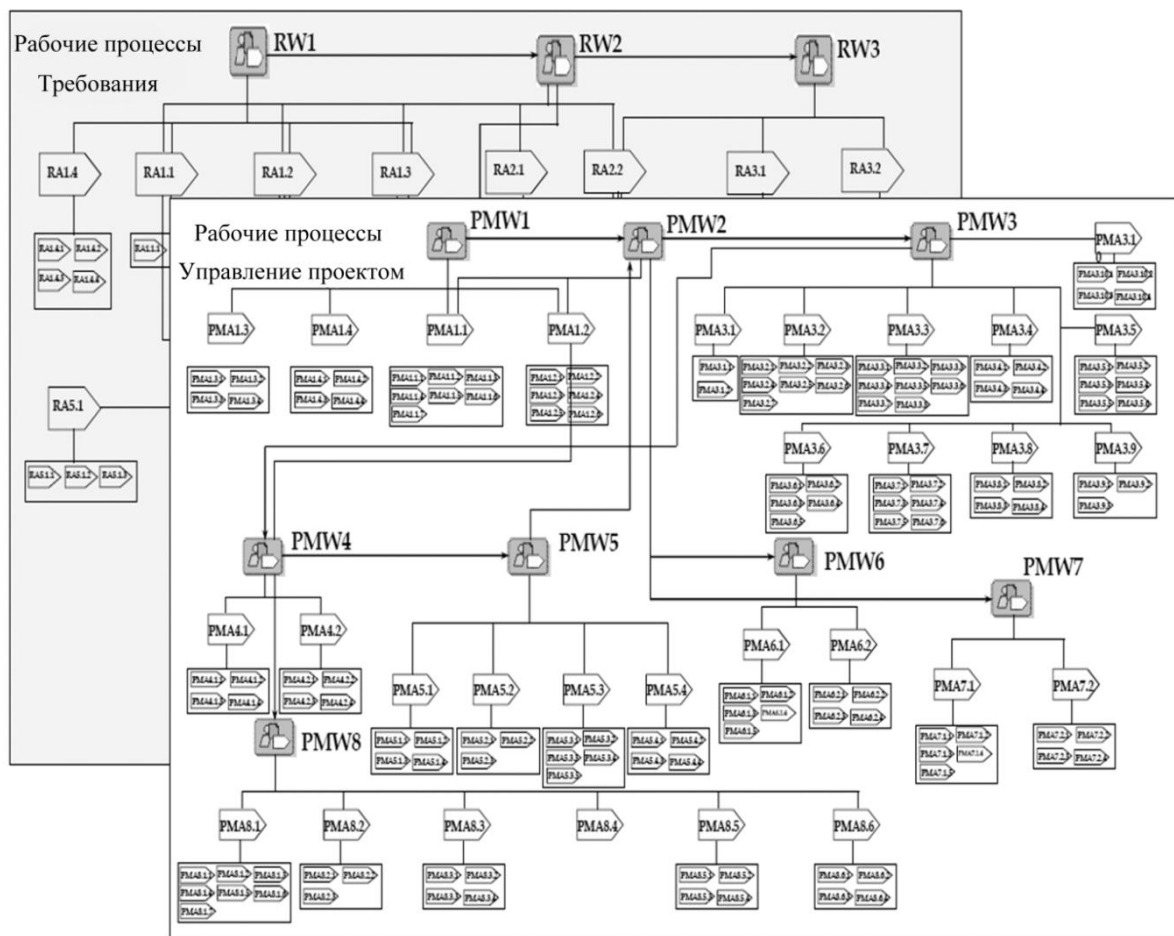


Рис. 2.6. Имитация RUP

Схемы раскрывают эти процессы на уровне рабочих процессов (RWX и PMWY) и подчинённых действий, которые изображаются пятиугольными иконками. Отметим, что RUP-гlossарий содержит следующие определения терминов, соответствующих схемам:

1. «Процесс» – это совокупность смежных видов действий, относящихся к основной «проблемной области».

2. «Рабочий процесс» – это последовательность действий, выполняемых в бизнесе, которая создаёт результат с видимой значимостью для каждого отдельного субъекта бизнеса.

3. «Деталь рабочего процесса» – это группа действий, которые выполняются в тесном взаимодействии для достижения определённого результата. Действия обычно выполняются либо параллельно, либо итеративно, при этом выходные данные одного действия служат входными данными для другого действия. Сведения о рабочих процессах используются для группировки действий, чтобы обеспечить более высокий уровень абстракции и улучшить понятность рабочих процессов.

4. «Действие» – это единица работы, которая обеспечивает значимый результат в контексте проекта. Она имеет чёткую цель, которая обычно включает в себя создание или обновление артефактов. Каждое действие назначается определённой роли. Действия могут повторяться несколько раз, особенно при выполнении в различных итерациях.

Можно отметить, что все указанные определения включают термин «действие», подчёркивая, что соответствующая единица работы выполняется проектировщиком, который играет определённую роль. В общем понимании «роль определяет поведение и обязанности человека или группы людей, работающих вместе как команда, в контексте организации разработки программного обеспечения. Роль отвечает за один или несколько артефактов и выполняет ряд действий» (Borges, Machado and Ribeiro, 2012) [58].

RUP поддерживает использование набора ролей, любая из которых приводит к определённой интерактивной модели, которая помогает проектировщику вести себя соответствующим образом. Таким образом, интерактивные схемы ролей также являются средством снижения сложности проектирования системы с программным обеспечением. Эти схемы являются хорошим классом примеров эффективного человеко-компьютерного взаимодействия.

В RUP артефакты – это ещё один класс компонентов визуализируемых моделей. Как показано на рис. 2.7, использование этих компонентов связывает рабочие процессы и действия внутри рабочих процессов.

Кроме того, иконки этих компонентов открывают интерактивный доступ к шаблонам артефактов или их визуализированным моделям, например, к необходимым UML-диаграммам.

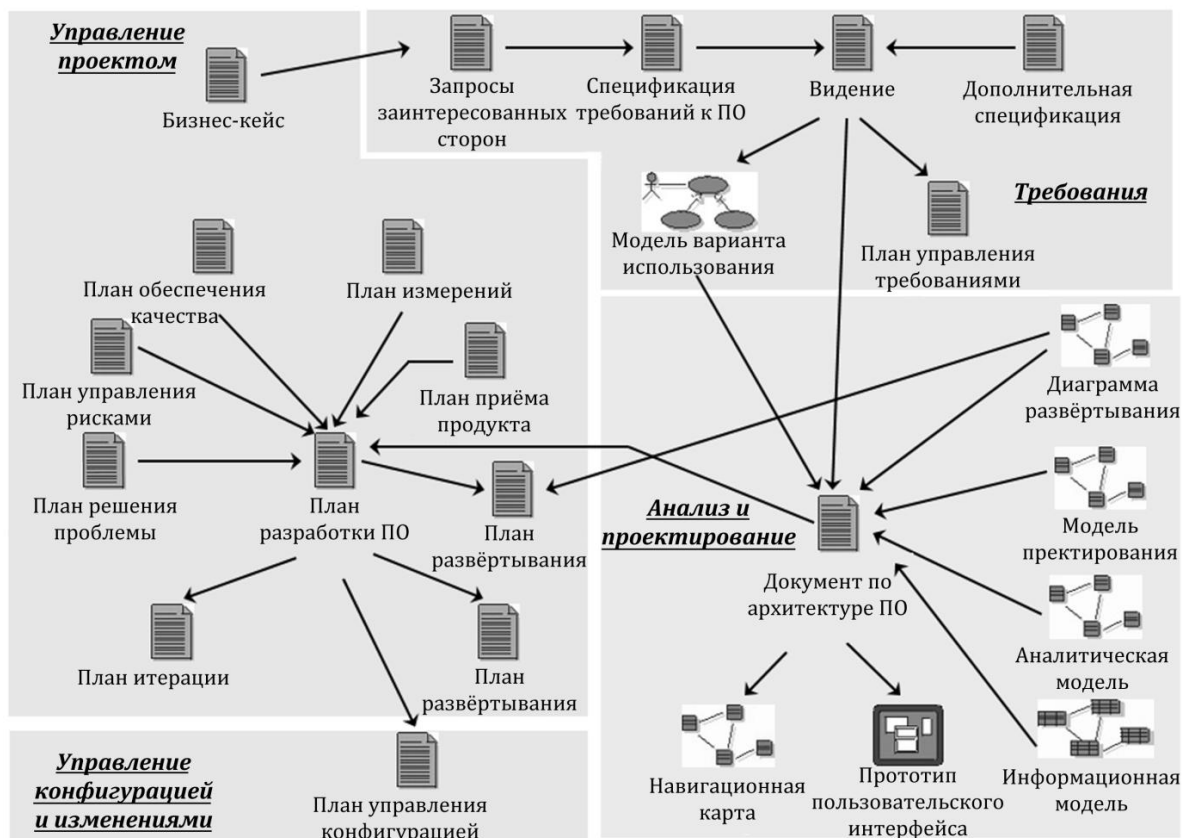


Рис. 2.7. Имитация RUP

2.2.3. Стандарты человеко-компьютерной деятельности

При разработке системы программного обеспечения (или подсистемы АС) проектировщик выполняет работу в определённом пространстве деятельности, которое в глоссарии SEMAT определяется как «поле для чего-либо, что необходимо сделать при разработке программного обеспечения». Важную часть этого пространства занимает компьютеризированная среда, с которой проектировщики взаимодействуют, используя соответствующий опыт с определёнными целями при выполнении работ в реальном времени. В любом элементе взаимодействия проектировщик воспринимает определённый взгляд на выбранное состояние компьютерной среды. В общем случае выбранное состояние сочетает состояния разрабатываемой системы и используемой части способа работы.

Первое из этих состояний отражает будущую систему, которая будет эксплуатироваться в определённой предметной области D . Разработка любой из таких систем невозможна без использования опыта E^D , соответствующего области D . Обычно разработчики системы приобретают необходимый объём опыта E^D в процессе разработки. Исходя из этого, они должны выражать приемлемую степень сложности в адекватных

требованиях к компьютеризированным интерфейсам, встроенным в систему. Такие требования должны отражать удобный доступ к необходимому опыту и его использование в реальном времени будущими пользователями. Этот опыт будет включать в себя не только опыт E^D , но и технологический опыт E_1^T , который будет необходим для использования способа работы, встроенного в систему её разработчиками.

Второе из указанных состояний соответствует применяемой технологии и инструментарию, создатели которых должны обеспечить допустимую степень сложности в любых точках потенциальных взаимодействий разработчиков с используемым способом работы. Обычно лучшие практики и соответствующие инструментальные средства разрабатываются как независимые от потенциальных областей их применения, но эти компоненты деятельностного пространства также отражают соответствующий технологический опыт E_2^T , который должен быть легко освоен и комфортно использован разработчиками будущей системы.

Таким образом, при обеспечении приемлемой степени сложности для разработчиков, которые будут взаимодействовать с компьютеризированными компонентами пространства деятельности, необходимо создать условия, при которых они будут эффективно осваивать и использовать отмеченные выше виды опыта (E^D , E_1^T и E_2^T). Иными словами, в условиях человеко-компьютерных взаимодействий разработчикам системы должны быть предоставлены средства, обеспечивающие доступ и использование необходимого опыта в режиме реального времени. Эта рекомендация, которая лучше всего понимается как требование, распространяется на создателей таких систем, как способ работы. Напомним, что в соответствии с подходом SEMAT, необходимый способ работы лучше строить разработчикам определённой системы в процессе их работы.

В любом варианте разработки технологии и соответствующего инструментария её создатели должны использовать соответствующие источники опыта E^T , среди которых можно отметить стандарты.

Для концептуального этапа проектирования важное место занимают стандарты, аккумулирующие опыт архитектурных решений. Существуют два известных стандарта, предусматривающих разработку архитектуры системы – IEEE-1471 [90] и ISO 42010 [91]. Основной идеей обоих этих стандартов является разработка архитектуры как соответствующего набора архитектурных представлений, любое из которых раскрывает систему с определённой точки зрения. Более строго, в соответствии со стандартом ISO 42010, нормативная структура архитектуры показана на рис. 2.8.

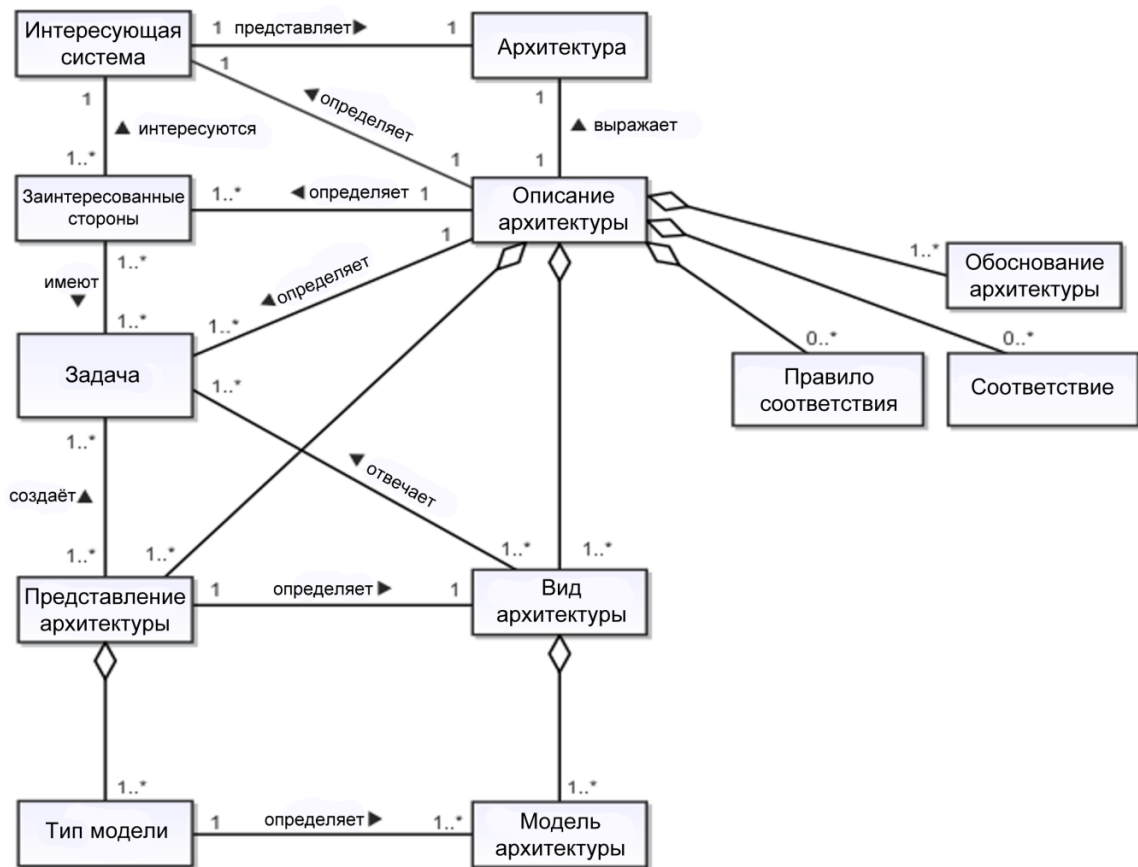


Рис. 2.8. Структура архитектуры системы

Явное построение и использование архитектуры при разработке систем неоднократно демонстрировали значимые положительные эффекты. А именно, архитектура:

1. Способствует извлечению требований и объединению их в систему;
2. Наглядно демонстрирует набор ранних проектных решений;
3. Предписывает организационную структуру системы (хорошо структурированные системы полны образцов);
4. Играет роль общей архитектурной основы для линии программных продуктов;
5. Даёт возможность рассматривать, согласовывать и предсказывать характеристики качества, в том числе результаты исследований;
6. Предоставляет информацию о распределении работ и календарных планах;
7. Даёт возможность более точно оценить стоимость и график работы;
8. Сокращает риски;
9. Является первой формой существования системы, которую можно проверить (протестировать) как целое;

10. Обеспечивает возможность передачи или повторного использования архитектурных стилей и каркасов;

11. Приносит пользу за счёт ограничения вариаций проекта или его частей;

12. Помогает в эволюционном прототипировании;

13. Составляет концептуальную целостность АС и процесс её разработки;

14. Поддерживает понимание и взаимопонимание в индивидуальной и коллективной работе;

15. Поддерживает выражение идей и аргументов при коммуникации лиц, участвующих в разработке систем;

16. Предоставляет возможность рассуждать о потенциальных изменениях в ходе развития АС, и способствовать управлению такими изменениями;

17. Может быть основой для изучения системы.

Приведённый выше список впечатляет, но что особенно важно, архитектура снижает сложность за счёт разделения возможных взаимодействий проектировщика с системой и процессом её разработки на сферы в соответствии с набором полезных точек зрения. Для каждой точки зрения архитектура открывает потенциальным пользователям доступ к соответствующей блок-схеме с нормативной семантикой.

В этом пособии в связи с его тематикой мы также отмечаем стандарт, называемый «Capability Maturity Model Integration» (или коротко СММІ) [88]. Данный стандарт разделяет технологические процессы, применяемые при разработке систем, на уровни в зависимости от их профессиональной зрелости, которая может быть измерена специалистами. Другими словами, любой процесс можно сравнить с нормативной схемой зрелости, которая в целом представлена в таблице. 2.3.

Таблица 2.3. Модель профессиональной зрелости процессов

Уровень	Фокус	Область деятельности	Результат
1	2	3	4
5 Оптимизируемый	<i>Непрерывное совершенствование процесса</i>	Организационные инновации и внедрение Анализ причин и принятие решений	Продуктивность и качество
4 Управляемый на основе количественных данных	<i>Управление на основе количественных данных</i>	Производительность организационного процесса Управление проектом на основе количественных данных	

Продолжение таблицы 2.3

1	2	3	4
3 Определённый	<i>Стандартизация процесса</i>	Разработка требований Техническое решение Комплексирование продукта Верификация Валидация Фокусирование на организационных процессах Описание организационных процессов Организационное обучение Комплексное управление проектом Управление рисками Анализ и принятие решений	Продуктивность и качество
2 Управляемый	<i>Основное управление проектом</i>	Управление требованиями Планирование проекта Мониторинг и контроль проекта Менеджмент договоров с поставщиками Измерение и анализ Обеспечение качества товаров и процессов Управление конфигурацией	
1 Начальный	<i>Компетентные люди и героизм</i>		

Данная схема отражает не только цели объединения лучших практик, но и целесообразность постоянного совершенствования как процессов разработки так и библиотек лучших практик, включая их разделы. Использование указанной модели приводит ко многим положительным результатам, одним из которых является уменьшение сложности способов работы.

Отметим, что аналогичные модели, ориентированные на пять уровней, были разработаны для различных коллекций лучших практик. Среди этих моделей мы выделяем следующие из них:

1. People Capability Maturity Model (модель зрелости возможностей персонала) – определяет профессиональную зрелость команды разработчиков и каждого члена этой команды;

2. Organizational Project Management Maturity Model (модель зрелости организационного управления проектами) – указывает пути улучшения проектного управления;

3. Business Process Maturity Model (модель зрелости бизнес-процессов) – соответствует формам и средствам совершенствования бизнес-процессов, не касаясь разработки АС;

4. Business Intelligence Maturity Model (модель зрелости Business Intelligence) – фокусирует внимание на достижении совершенства с точки зрения интеллектуализации деятельности;

5. Enterprise Architecture Maturity Model (модель зрелости архитектуры предприятия) – включает списки практик для создания архитектуры предприятия и рекомендации для её совершенствования.

2.3. Особенности деятельности человека в физическом мире и компьютерной среде

2.3.1. Эмпирическая природа человеческой деятельности в физическом мире

Основным отличием человека от других существ является использование управляемых естественных взаимодействий (N-взаимодействий) с окружающей средой. Более того, при необходимости человек может моделировать взаимодействия для ситуационного планирования с целью достижения необходимого или благоприятного эффекта.

Это различие является результатом эволюции жизни на Земле за счёт использования различных форм экспериментов в эволюционном процессе. В эволюции природа не только экспериментировала, но и кодировала результаты экспериментов для их повторного использования живыми существами, другими словами, кодировала опыт существ.

Для кодирования опыта живых существ природа «изобрела» следующие способы:

– Генетический способ, который использует кодирование опыта в хромосомной памяти для повторного использования протестированных форм жизни;

– Повторное использование на основе условных рефлексов, которые кодируются в нейронных структурах.

Чтобы поддерживать жизнь своим опытом, живые существа используют определённые органы чувств, которые позволяют снабжать их информационными потоками об условиях окружающей среды. После любого акта восприятия, введённая информация передается в нейронную память, где она обрабатывается для достижения ряда полезных эффектов.

Среди этих эффектов, но более чем в упрощённом варианте с указанием их особенностей и без пояснений, отметим следующие основные виды эффектов:

1. Органы чувств обеспечивают мониторинг условий окружающей среды, некоторые из которых как источники определённого стимула инициируют врождённые, обычно фиксированные паттерны инстинктивных реакций, закодированные в генетическом опыте живого существа.

2. При мониторинге обработка информации показывает, что воспринимаемое состояние окружающей среды соответствует определённому условному рефлексу, в соответствии с которым живое существо активизирует соответствующую реакцию, полезность которой была обнаружена этим живым существом в предыдущих восприятиях аналогичных условий окружающего мира.

Первый эффект основан на генетическом опыте, коды которого формируются в генетических структурах при очень длительной обработке воспринимаемых условий механизмами адаптации, которыми обладает любой вид живых существ. При этом соответствующие инстинктивные реакции обусловлены врождёнными биологическими факторами.

Условные рефлексы открыли возможность положительного влияния дополнительной приспособляемости на естественный отбор при изменении условий жизни существ, приобретших аналогичные рефлексы. Этот механизм выполняет свою избирательную функцию на уровне поколений живых существ без управляемой возможности переноса приобретённых условных рефлексов между живыми существами и поколениями.

У определённого живого существа в нейронных структурах, выполняющих функцию памяти при обучении, формируются единицы опыта, основанные на условных рефлексах. Эта особенность условных рефлексов находит своё применение в теории и практике искусственных нейронных сетей, основные приложения которых связаны с предметными областями «распознавание образов» и «анализ сцен», где моделированию условного рефлекса CR_i соответствует следующее логическое выражение: если <определённые условия C_i >, то <соответствующая реакция R_i >.

Для множества $\{CR_i\}$ определённый вид живых существ должен иметь не только механизмы формирования $\{CR_i\}$, но и механизмы доступа к единицам этого множества. Мощность множества и функции обоих этих механизмов существенно зависят от биологических ограничений такого рода. Именно поэтому эволюция жизни на Земле пыталась найти необходимость увеличения мощности множества $\{CR_i\}$ и совершенствования указанных механизмов ΔE_i^N .

На следующем этапе эволюции жизни, основанной на опыте, природа «изобрела» разумную форму естественного опыта E^N , единицы которого $\{\Delta E_i^N\}$ создаются человеческими существами с помощью интеллектуальной обработки условных рефлексов. Основой этой

обработки является естественный язык L^N , что привело к включению средств второй сигнальной системы во взаимодействие человека с окружающим миром.

В общем смысле язык L^N – это система знаков, каждый из которых ссылается на соответствующий объект, когда это необходимо или полезно для человека. Главное в таких ссылках то, что их можно использовать в качестве вторичных сигналов о наличии соответствующих объектов в воспринимаемой сцене (ситуации). Именно поэтому И. П. Павлов указал язык L^N в качестве второй сигнальной системы.

Таким образом, объекты окружающей среды могут восприниматься органами чувств человека в процессе, который дополнительно использует соответствующие знаки L^N . Такое включение приводит, например, к следующим полезным эффектам:

1. Выбранные знаки могут быть использованы для правильного восприятия сцены за счёт следующих действий:

- Фокусирование внимания человека на соответствующих объектах;
- Расширение чувственно воспринимаемой информации за счёт использования значений используемых знаков;
- Взаимная коррекция информационных потоков, поступающих от первичных (органов чувств) и вторичных сигнальных систем.

2. Воспринимаемые знаки или их сочетания позволяют моделировать процессы восприятия для следующих целей:

- Воображение ситуаций, недоступных для восприятия органами чувств по различным причинам;
- Имитация другого вида чувств за счёт использования приборов измерения физических величин, которые не воспринимаются органами чувств человека.

И всё же основной целью языка L^N является принципиальное расширение и совершенствование совокупности психических механизмов, обеспечивающих создание единиц $\{\Delta E_i^N\}$ в виде модифицированных условных рефлексов. Включение этого языка в процесс создания человеческого опыта привело к формированию системы интеллектуальных способностей, поддерживающих, например, следующую интеллектуальную деятельность:

- Абстрагирование как мысленное извлечение существенных характеристик из их множества, задающего определённый объект или их класс;
- Анализ как процесс разбиения сложной темы или материала на более мелкие части, чтобы получить лучшее понимание этого;

- Обобщение как перенос на более высокий уровень абстракции путём выявления общих признаков объектов в определённой проблемной области;
- Творческое воображение как изобретение полезных образов;
- Сопереживание как творческое воображение и понимание чьих-то чувств и мыслей рациональным способом;
- Оценка выражения значения или суммы числами или другими способами;
- Мышление как формирование идеи или концепта;
- Прогнозирование как объявление или оценка того, что произойдет в будущем или будет следствием чего-то;
- Осознанная мотивация создания необходимой единицы ΔE_i^N ;
- Постановка целей в формах, позволяющих оценить их достижение;
- Понимание человека в качестве адекватной оценки определённого восприятия или воображения на уровне схемы (ситуации, системы или процесса) с точки зрения его целостности.

Необходимо различать, где и как осуществляется интеллектуальная деятельность. В любом случае, происходит она в ментальном пространстве, с переносом определённых частей интеллектуальных процессов за пределы мозговых структур или нет. Указанная интеллектуальная деятельность и другие её виды постоянно совершенствуются. Основой совершенствования является определение их естественной сущности. В настоящее время существует богатейшая коллекция лучших практик, обеспечивающих использование автоматизированной интеллектуальной деятельности и даже её автоматического моделирования для применения в жизни человека.

Попытки раскрыть сущность интеллектуальной деятельности и её участие в создании новых единиц человеческого опыта имеют долгую историю. Эти попытки привели к пониманию принципиальной роли экспериментов и логики в овладении опытом E^N , приобретении его новых единиц и их постоянном совершенствовании.

Когда люди взаимодействуют с окружающим миром, они неявно или явно используют доступный опыт E^N и язык L^N и такая реальность создает предпосылки для экспериментов. В таких условиях, если взаимодействия поддерживают активность человека, которая может привести к созданию новой единицы ΔE_i^N , то такая активность может быть интерпретирована как проведение соответствующего эксперимента. Это обусловлено наличием информационных потоков, поступающих от органов чувств, преобразованием этих потоков в первичные данные, которые

интеллектуально обрабатываются для получения необходимых результатов в соответствии с целью деятельности.

Таким образом, даже в неосознанной мыслительной деятельности, приводящей к новой единице ΔE_i^N , эта деятельность может быть квалифицирована как неявное мысленное экспериментирование. Явное и преднамеренное использование языка и доступного опыта E^N в первую очередь обеспечивает управление процессом мысленного экспериментирования. Кроме того, они способствуют улучшению интеллектуальных способностей, участвующих в проведении мысленного эксперимента, что оставляет «следы» внутреннего интеллектуального процесса вне мозга. Эти следы не только отражают информацию об используемой интеллектуальной деятельности, но и могут быть применены для контролируемого воздействия на процесс мысленного экспериментирования.

Было время, когда люди поняли, что экспериментирование следует выделять как самостоятельную предметную область, особенно в науке. После этого человечество изобрело, развило, освоило, постоянно использовало, а также применяет и сейчас богатую коллекцию экспериментальных практик. Поэтому опыт экспериментирования E^N будет применяться и постоянно совершенствоваться для его интенсивного использования в человеко-компьютерной деятельности. Следует отметить, что экспериментирование – это творческий вид деятельности, и поэтому оно всегда включает в себя действия внутри мозга, которые переплетаются с действиями за пределами мозга. Поэтому любой эксперимент можно понимать как мысленный эксперимент, проведение которого поддерживается действиями вне мозга.

Как сказано выше, логика также занимает важную роль в создании и использовании единиц $\{\Delta E_i^N\}$. В этой области логика открывает следующие основные возможности интеллектуальной обработки информации:

1. Отображение на концептуальном уровне абстракций, где человек может эффективно моделировать объекты и процессы, которые размещены или происходили как за пределами мозга, так и в ментальном пространстве.

2. Осуществление вывода, доказательства и аргументации для обоснования прогноза или подтверждения гипотез или теорем.

3. Осуществление рассуждения с контролируемым использованием естественных и искусственных языков.

4. Управление интеллектуальной обработкой.

5. Программирование как естественное программирование (N-программирование) на языке L^N , а также как программирование на

искусственных алгоритмических языках (программирование как прикладная логика).

Для надёжного использования логики отображение заинтересованных компонентов деятельностного пространства должно обеспечивать необходимую степень адекватности, которая оценивается его истинностью. Кроме того, при обратном отображении описания языка в реальность результаты логической обработки должны проверяться путём проведения экспериментов в соответствующем пространстве деятельности.

Помимо поддержки и совершенствования механизмов создания единиц $\{\Delta E_i^N\}$, использование языка L^N практически снимает ограничения на мощность множества единиц опыта, в первую очередь за счёт систематизации этого множества и совершенствования механизмов доступа к его элементам. В этой ответственности L^N необходимо различать имплицитные и эксплицитные интеллектуальные механизмы упорядочения и поиска, основанные на потенциале языка.

Сущность имплицитного упорядочения и поиска обусловлена системной природой языка L^N и ассоциативными отношениями между элементами системы, прежде всего ассоциативными отношениями в лексике и её употреблении. Эти особенности L^N обеспечивают функционирование такого феномена человеческой памяти, как «запоминание».

Развитие эксплицитных систематизаций языка L^N и их использование в управляемых механизмах упорядочения и поиска приводят к существенному улучшению имплицитных взаимодействий человека с опытом.

В обоих видах взаимодействия человека с естественным опытом любой акт взаимодействия начинается с попытки автоматического доступа к опыту E^N в определённых условиях. На этом этапе любого взаимодействия соответствующие предварительные условия преобразуются в список ключей (выраженный на языке L^N), назначение которых аналогично запросу к базе данных.

Для повышения адекватности запроса список ключей можно преобразовать в соответствующий вопрос Q в виде конструкта на языке L^N . Среди возможных ответов на этот вопрос есть два строго противоположных варианта – опыт E^N включает единицу ΔE_i^N , которая либо соответствует запросу, либо не соответствует. Если запрос правильный, то причиной отрицательного ответа является отсутствие необходимой единицы ΔE_i^N , что может быть связано с существованием пробела, который в разных случаях может иметь разную природу.

2.3.2. Особенности человеческой деятельности в компьютеризированной среде

Сказанное в предыдущем подразделе верно, когда человек вовлечён в человеко-компьютерную деятельность, сочетающую действия человека с компьютерными процессами, например, при выполнении определённой работы. Эту деятельность можно трактовать как сотрудничество человека и компьютера в выполнении общей работы, эффективность и другие характеристики которой существенно зависят от того, как это сотрудничество организовано и реализовано. В этом случае за объективацию такого сотрудничества отвечают человеко-компьютерные взаимодействия, которые должны быть разработаны, запрограммированы и встроены в общий процесс.

Именно поэтому в известных учебных программах ACM SIGCHI по человеко-компьютерному взаимодействию (Hewett et al., 2002) [26] «human-computer interaction» (HCI) формально определяется как «дисциплина, связанная с проектированием, оценкой и внедрением интерактивных вычислительных систем для использования человеком и с изучением основных явлений, окружающих их». Там же, использование и контекст HCI связаны с системой обучения HCI, которая включает следующие курсы:

1. Природа HCI: (Мета-) Модели HCI.
2. Использование и контекст компьютеров: Человеческая социальная организация и работа; Области применения; Человеко-машинная адаптация.
3. Характеристики человека: Обработка информации человеком; Язык, общение и взаимодействие; Эргономика.
4. Архитектура компьютерных систем и интерфейсов: Устройства ввода и вывода; Диалоговые методы; Диалоговый жанр; Компьютерная графика; Диалоговая архитектура.
5. Процесс разработки: Подходы к проектированию; Методы реализации; Методы оценки; Кейсы и примеры систем.
6. Проект: Презентации и экспертизы проектов.

Представленная структура обучения HCI отражает многообразие знаний и опыта, относящихся к соответствующей предметной области, и это многообразие указывает на её междисциплинарный характер.

В данном пособии наши интересы касаются области HCI только с точки зрения использования доступного опыта проектировщиком, решающим задачу в рамках человеко-компьютерной деятельности. Вот почему ниже в тексте мы откроем и уточним известные детали области HCI только фрагментарно и только тогда, когда это будет необходимо для

рассуждений или решений. В этом подразделе текста он необходим для сравнения N-взаимодействий с взаимодействиями типа HCI.

Более того, при сравнении и далее мы будем рассматривать только визуальное взаимодействие человека с компьютеризированной средой. Данное ограничение выбрано в связи с наличием обширного опыта использования визуализации в различных её приложениях в человеко-компьютерной деятельности. Поэтому использование этого разнообразия достаточно для репрезентативного сравнения N-взаимодействий с человеко-компьютерными взаимодействиями.

Одним из источников систематизации информации на экране монитора является периодическая таблица визуализации (Lengler and Erpfer, 2007) [76], которая в интерактивном режиме демонстрирует сто методов. Авторы таблицы определяют метод визуализации как «систематическое, основанное на правилах, внешнее, устойчивое графическое представление, которое изображает информацию таким путём, что способствует проникновению в суть, усовершенствованию понимания или распространению опыта». Рис. 2.9 раскрывает структуру таблицы, любая ячейка которой представляет соответствующий метод.

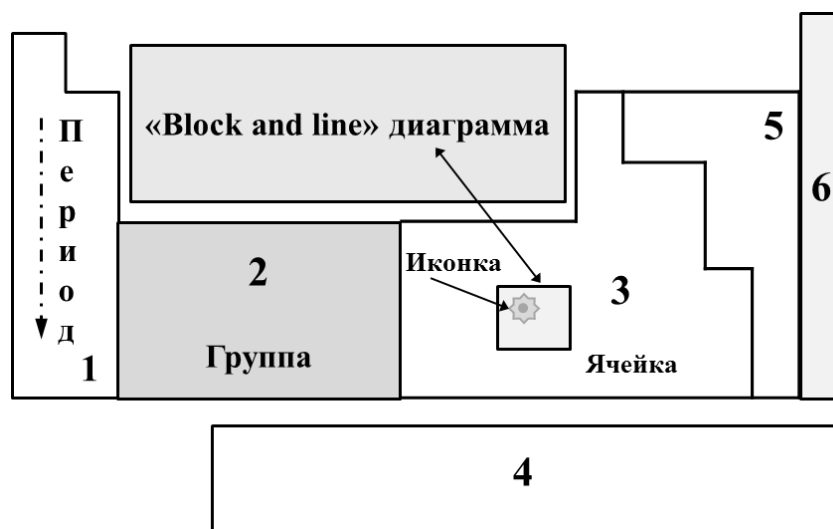


Рис. 2.9. Структура периодической таблицы

В данной таблице методы визуализации распределены по группам (1-6), соответственно периодам каждой группы. Используемые группы указывают на следующие области применения:

1. **Визуализация данных** сочетает методы визуального представления количественных данных в схематической форме.

2. **Визуализация информации**, например, семантические сети или деревья, направленные на визуальное представление данных с целью усиления восприятия.

3. **Визуализация концепций** помогает графически отобразить качественные концепты, идеи и результаты их анализа.

4. **Визуальные метафоры** похожи на шаблоны, которые передают сложные метафорические идеи.

5. **Визуализация стратегий** определяется «как систематическое использование дополнительных визуальных репрезентаций с целью улучшения анализа, разработки, формулирования, коммуникации и реализации стратегий в организациях».

6. **Комплексная визуализация** подразумевает использование нескольких методов, выбранных из вышеперечисленных групп.

Параметр «период» отражает сложность визуализации, которая выражается в любой ячейке таблицы «количеством правил, применяемых в методе и/или количеством взаимозависимостей элементов, подлежащих визуализации» (Lengler and Eppler, 2007).

Кроме того, каждая ячейка содержит определённую иконку, атрибуты которой указывают на следующие особенности методов (Lengler and Eppler, 2007).

1. *Взгляд на проблему [когда?]*: Детальный (подчёркивает конкретные элементы), Общий (картина целиком), Общий и Детальный (оба способа одновременно).

2. *Способ мышления [почему?]*: Конвергентное (уменьшает сложность) vs. Дивергентное (повышает сложность).

3. *Тип Репрезентации [что?]*: Процесс (ступенчатый циклический во времени и/или непрерывный последовательный), Структура (т.е. иерархия или причинно-следственная сеть).

Авторы периодической таблицы методов визуализации создали её в учебных целях, и, следовательно, таблица частично отражает опыт компьютерной визуализации. Для практического применения этот опыт распространяется на набор стандартов, среди которых мы выделяем семейство стандартов IDEF (Integration DEFinition) [89] и стандартный унифицированный язык моделирования (UML, последняя версия 2.5) [92]. Семейства IDEF и UML разработаны для целей визуального моделирования в области системной и программной инженерии. Поэтому стандарты глубоко продуманы и имеют строгую семантику.

Любой стандарт семейства IDEF помогает проектировщику создать блочно-линейную схему для своих приложений. Например, стандарт IDEF5 поддерживает визуальное моделирование при разработке и поддержании пригодных для использования, точных онтологий предметных областей. В качестве метода этот стандарт включает графический язык для поддержки концептуального анализа онтологии, язык структурированного текста для детальной характеристики онтологии и систематическую процедуру, которая обеспечивает руководящие

принципы для эффективного формирования онтологии. Предполагается, что проектировщик создаёт необходимые схемы, управляя ими мысленным взором в процессе визуального моделирования. Этот язык, как и другие языки семейства IDEF, не выполняется автоматически. Типичной инструментальной поддержкой IDEF-версий визуального моделирования является использование специализированных графических редакторов.

UML изначально был направлен на визуальное моделирование в программной инженерии с точки зрения перевода UML-диаграмм в выполняемый код соответствующих частей проектируемой программы. Это хорошая цель, но автоматический перевод определённого комплекса UML-диаграмм в соответствующий выполняемый код до сих пор не реализуется. Способы перевода разрабатываются и тестируются для некоторых UML-диаграмм, но не для всего разнообразия их типов. Поэтому графические редакторы также являются базовыми средствами для схематического рассуждения на основе UML.

Анализируя визуальные взаимодействия человека с компьютером, необходимо учитывать взаимодействие с интегрированными инструментальными средами, богатым источником которых являются инструментальные средства, встроенные в Microsoft Office в любых его версиях. Эти средства открывают доступ к разнообразным видам визуальных сцен, каждая из которых отражает набор ситуаций для потенциальных действий. На рис. 2.10 представлен фрагмент одной из таких сцен, используемых в Microsoft Word.

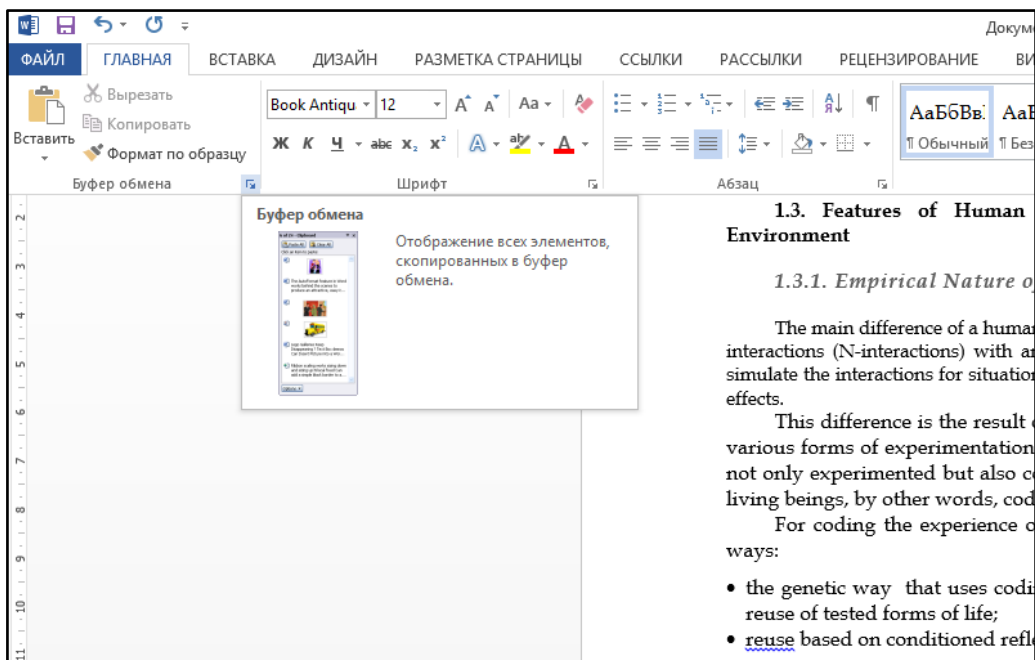


Рис. 2.10. Фрагмент визуальной сцены

Как мы видим, показанная сцена далека от привычных ситуаций, возникающих перед человеком вне компьютера. Она также далека от визуализированных образов этих ситуаций. Любая наблюдаемая сцена состоит из карты областей, помеченных словами, или используемых для ввода информации, значков с их именами или без них. Большая часть наблюдаемых компонентов поддерживает выбор действий, некоторые из которых выполняют только роль информационной помощи.

Таким образом, подобные сцены предназначены для использования их компонентов в технологических целях в условиях, когда эти компоненты отражают поверхностное содержание действий (и их групп), обычно на уровне их названий.

2.3.3. *Человеко-компьютерное взаимодействие с точки зрения сложности*

В реальности при разработке системы любой проектировщик взаимодействует с компьютеризированной средой, когда естественные и искусственные взаимодействия переплетаются в режиме реального времени в условиях, схематично показанных на рис. 2.11.



Рис. 2.11. Условия человеко-компьютерного взаимодействия

Схема указывает на часть естественных взаимодействий, которая неявно фокусируется на определённом объекте сложности, в то время как их вторая часть неявно воспринимает запрограммированные интерфейсы. Параллельно проектировщик участвует в явных искусственных

взаимодействиях с программируемыми интерфейсами, которые отражают объект сложности с технологически predetermined стороны.

Таким образом, predetermined особенность искусственных взаимодействий приводит к ограничениям в отображении объекта сложности. Кроме того, они односторонне активируют естественные взаимодействия, которые находят неявно объективированное выражение только во второй их части. Первая часть естественных взаимодействий остаётся без поддержки автоматизации, в то время как эта часть имеет значительно больший полезный потенциал, который несопоставим с возможностями, предоставляемыми второй частью.

Если в контексте проектирования систем неявные естественные взаимодействия будут автоматизированы полезным образом, то такая автоматизация поможет снизить сложность не только в predetermined ситуациях человеко-компьютерных взаимодействий. По нашему глубокому убеждению, автоматизация взаимодействия с естественным опытом является перспективным способом уменьшения сложности в режиме реального времени в непредсказуемых ситуациях и снижения негативного влияния человеческого фактора при проектировании систем.

Контрольные вопросы

1. Какие системы относятся к классу «автоматизированных систем, интенсивно использующих программное обеспечение»?
2. Какие критерии до 2015 года составляли «железный треугольник успеха» проектов по оценкам компании Standish Group?
3. Какие факторы, представленные в отчёте Haze, относятся к области человеко-компьютерного взаимодействия?
4. Каким феноменом обусловлены негативные влияния человеческого фактора на процесс проектирования систем?
5. Как интерпретация меры Колмогорова реализуется учёными и практиками в оценке сложности систем?
6. Какое место в проектировании систем занимает технология разработки?
7. Что такое уровень зрелости процесса проектирования?
8. Какие средства в процессе проектирования используются для упрощения сложности?
9. Что такое «Рабочий процесс» в соответствии в RUP-гlossарием?
10. Что представляет собой архитектура АС?
11. К каким позитивным эффектам приводит использование архитектуры?
12. Чем по своей сути является стандарт IEEE-1471-2000?
13. Сколько уровней в модели профессиональной зрелости процессов в соответствии со стандартом CMMI?

ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНЫЕ ДЕЙСТВИЯ НАД КОНЦЕПТУАЛЬНЫМИ ОБЪЕКТАМИ

3.1. Содержательно-эволюционный подход к проектированию систем

3.1.1. Теоретизация в проектировании систем

В своих первоначальных установках и даже в названии SEMAT-подход подчёркивает необходимость теоретизации основ разработки программного обеспечения. В то же время нормативный документ «Kernel and Language for Software Engineering Methods, Version 1.1» не содержит конструктивных ответов о создании и использовании соответствующих теорий при разработке программных систем.

Даже при согласовании позиций по инициативе SEMAT и на ранних этапах эволюции в SEMAT А. Кокберн критиковал этот аспект подхода (Cockburn, 2010) [81]. В последние годы предпринимались многочисленные попытки предложить варианты теоретизации. Некоторые из этих попыток и их результаты были представлены на четырёх ежегодных семинарах по общей теории программной инженерии (GTSE 2011-2015) [22, 30, 42, 43] и 5-м Международном семинаре по теоретически ориентированной программной инженерии (2016). В то же время можно констатировать, что удовлетворительная общая теория до сих пор отсутствует.

Как правило, после семинара его результаты обобщаются, и, например, в таком отчёте для второго семинара указывается, что теория должна «объяснять и предсказывать явления программной инженерии на нескольких уровнях, включая социальные процессы и технические артефакты, должна синтезировать существующие теории из программной инженерии и справочных дисциплин, должна разрабатываться итеративно, должна избегать распространённых заблуждений и атеоретических концепций, а также должна учитывать сложность явлений программной инженерии» (Johnson, 2013) [30, 43].

Доклады следующего семинара расширили возможности, которые должны найти своё воплощение в содержании и функциях GTSE, например, отчёт о результатах третьего семинара выделил работы следующих авторов:

Paul Ralph (Ralph, 2014) [37, 42], который «различал теории дисперсии (которые предсказывают зависимую переменную относительно независимых переменных) и теории процессов (которые объясняют, как происходит явление)».

Pan-Wei Ng (Ng, 2014) [42], который «предложил подход к разработке программного обеспечения, основанный на Essence. Он утверждает, что каждый программный проект уникален и чувствителен к своему контексту».

Очень часто исследователи, участвовавшие в попытках построить общую теорию, ссылались на статью (Sjoberg, 2007) [46], которая включает анализ назначений теорий и способ создания полезных теорий, ориентированных на их применение в программной инженерии. В своих исследованиях авторы данной статьи ссылались на следующую классификацию, описанную в работе (Sjoberg, 2007):

1. *Анализ*. Теории данного типа включают описания и представления типа «что есть что-либо». Также сюда включаются таксономии, классификации и онтологии.

2. *Объяснение*. Теории данного типа направлены на явное объяснение. Нетривиальным является вопрос о том, что составляет объяснение. Так или иначе, общее представление об объяснении заключается в ответе на вопрос *почему* что-либо есть или происходит (вместо вопроса *что* происходит).

3. *Прогнозирование*. Такие теории направлены на предсказание того, что произойдёт, без объяснения причины.

4. *Объяснение и прогнозирование*. Теории данного типа сочетают признаки второго и третьего типа и соответствуют тому, что многие считают «стандартной» концепцией эмпирически обоснованных теорий.

5. *Проектирование и деятельность*. Такие теории описывают «как делать» что-либо, таким образом, они предписывающие.

Кроме того, в статье (Sjoberg, 2007) [46] проанализированы следующие шаги, которые помогают построить определённую теорию:

1. Определение конструкторов теории;
2. Определение положений теории;
3. Подготовка объяснений для подтверждения теории;
4. Определение области применения теории;
5. Проверка теории с помощью эмпирических исследований.

Затем, в исходной теории для разработки на основе UML в больших проектах, для демонстрации этого способа, эти шаги были применены для построения «Исходной теории для разработки на основе UML в больших проектах». Разработанная теория была оценена авторами с использованием следующих характеристик:

– Проверяемость как степень, в которой теория построена таким образом, что эмпирическое опровержение возможно.

– Эмпирическая поддержка как степень, в которой теория поддерживается эмпирическими исследованиями, подтверждающими её обоснованность.

– Объяснительная сила как степень, в которой теория учитывает и предсказывает все известные наблюдения в пределах своей области, проста в том, что она имеет мало специальных предположений и связана с тем, что уже хорошо понято.

– Экономность как степень, до которой теория построена экономически с минимумом понятий и положений.

– Общность как широта охвата теории и степень независимости теории от конкретных установок.

– Полезность как степень, в которой теория поддерживает соответствующие области индустрии программного обеспечения.

Для наших рассуждений, представленных ниже, мы отметим ещё одну статью (Perry, 2016) [33], которая называлась «Theories, Theories Everywhere», где её автор утверждал, что общая теория программной инженерии должна иметь «две логические части: проектирование и оценка, D и E, каждая из которых является теорией T». На основе анализа этой позиции автор «очерчивает богатое разнообразие теорий, связанных с D и E, и считает их подтеориями, критичными для понимания отношений между теориями в D и E».

В спецификациях результатов анализа используется такая конструкция, как модель M, сущность которой уточняет следующее утверждение: «теория T (более или менее абстрактная сущность) овеществляется, представляется, удовлетворяется и т. д. по модели M (конкретная сущность)». В статье (Perry, 2015) [33] важное место занимает разделение теорий, соответствующих D и E, на подтеории, среди которых, например, мы выделяем следующие их множества: теории поведения и ограничений, теории структуры модели, теории удобства интерфейса, теории полезности, теории об оценке моделей и теории об оценке теорий.

3.1.2. Становление теории

Понятие «теория» в моделировании научной деятельности соответствует системе знаний, объединяющей совокупность общих идей, направленных на упорядочение фактов и поддержку процессов объяснения или понимания в определённой сфере действительности.

Конкретная теория их совокупностей может обладать и другими свойствами, например, обладать способностью оценивать состояние дел или генерировать прогнозы, и это приводит к различению видов теорий от описательных до дедуктивно-аксиоматических типов. Именно поэтому, в зависимости от уровня теории, она даёт возможность поддерживать соответствующий тип процессов в человеко-компьютерных средах: формальные, полуформализованные или эвристические,

неформализованные. В каждом процессе определённая нагрузка падает от теории на компоненты компьютера или человека.

Теория вносит порядок и согласованность в отображаемый мир. Этот порядок имеет логический характер. Теория бесполезна без её интерпретации в предметной области. Многие проблемы, связанные с построением и использованием теории, решаются её языком. В связи с этим с использованием теории связана методология знаковых вариантов моделирования исследуемых объектов. Одна из особенностей теории проявляется в том, что она обеспечивает систему понятий для их использования исследователем, который будет формулировать вопросы и план проведения эксперимента в человеко-компьютерной среде. Теория, образуя некую концептуальную систему, определяет, что следует и что не следует классифицировать как факт. Это стимулирует осмысленный поиск.

Теория никогда не появляется в своей окончательной форме без предыдущих попыток её описания. Незавершённая, несовершенная теория вовсе не редкость. Процесс создания теории, как и процесс её проверки, никогда не заканчивается и требует творчества и воображения. Лучше придерживаться теории, которая имеет слабые места, чем оставаться без теории.

В нашем исследовании и практике определённой предметной области W мы используем схему становления теории $Th(t)$, которая показана на рис. 3.1, где динамика $Th(t)$ объединяет процессы и расширения теории, происходящие в ряде фаз теории.

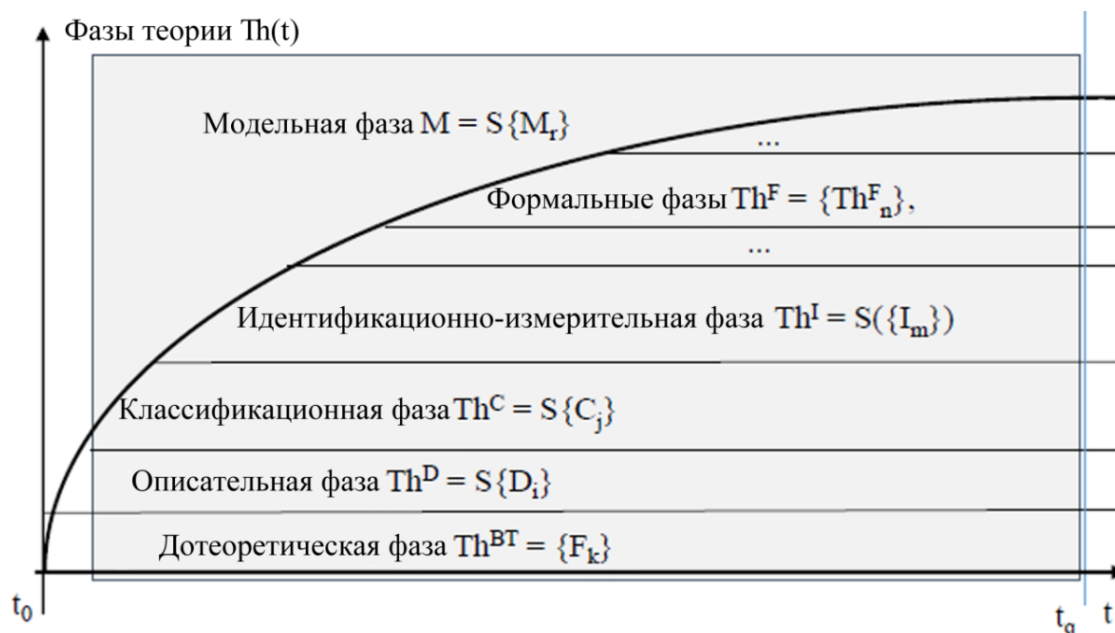


Рис. 3.1. Развитие фазовой структуры теории

1. **Дотеоретическая фаза** включает в себя набор фактов $F(t) = \{F_k\}$, относящихся к соответствующей предметной области, а скорее к свойствам её компонентов $\{B_p\}$ и отношений между ними.

2. В **описательной фазе** из зарегистрированных фактов конструируются тексты $T(t) = \{T_i\}$, связывающие факты в описании сущностей, выделенных в предметной области, и эти конструкции уже вносят определённый порядок (системность) в совокупности фактов.

3. При построении теории особое внимание уделяется словарию и особенно развитию его части, представляющей систему понятий $S(\{N_p\})$, семантическое значение и порядок которых определяют соответствующий набор классификаций $C(t) = \{K_m\}$. Разработка системы $S(\{N_p\})$ на основе $\{K_m\}$ является очень важной фазой становления теории. На рис. 3.1, эта часть теории отмечена как **классификационная фаза**.

4. В становление теории и её использование **идентификационно-измерительная фаза** вводит возможность эмпирической интерпретации теоретических конструкций, в частности, возможность контролировать адекватность любого концептуального образования или понятия при их использовании в выбранном фрагменте описания или факта. Типичным подходом к идентификации является распознавание образов с использованием соответствующих средств, которые приводят к именам классов. Измерение помогает определить значения атрибутов для выделенных сущностей теоретизированной реальности.

5. **Формальная фаза** обычно выражается одной или несколькими формальными теориями $ThF(t) = \{ThF_k\}$, систематизирующими материал предыдущих фаз и переносящими решение задачи с уровня манипуляций с сущностями $\{U_r\}$ предметной области PO_s на уровень манипуляций с символическими конструкциями теории. Такой переход открывает возможность использовать вывод для предсказаний и применять доказательства для проверки предсказаний.

6. Любая теория создаётся для её использования, очень важным видом которого являются модели. Модели являются посредниками между теорией и реальностью, когда проектировщики взаимодействуют с ними. Поэтому модели образуют полезную область теоретических приложений, которая показана на рис. 3.1 как **модельная фаза**. Модели выходят за рамки теории и служат не только для её применения, но и для её развития. Место моделей в человеческой деятельности показано на рис. 3.2.

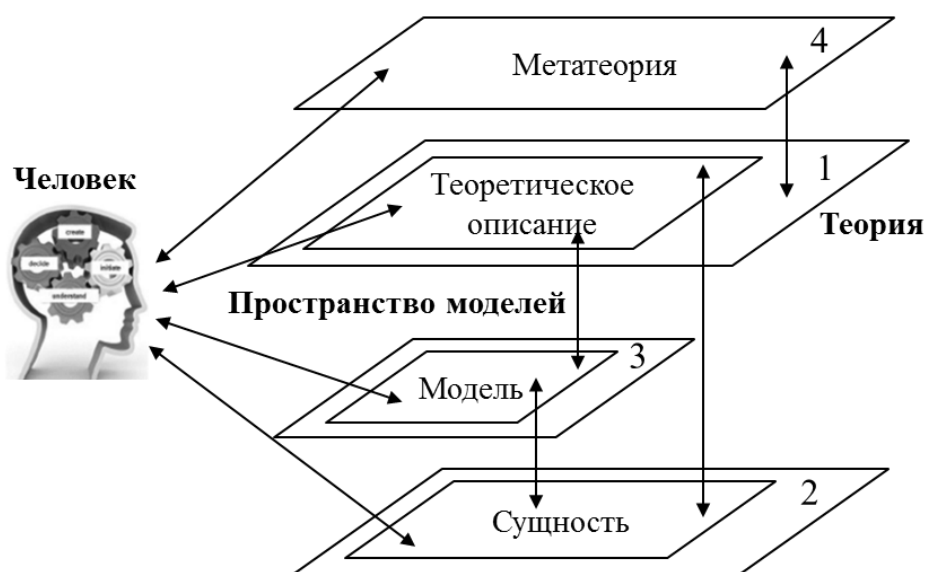


Рис. 3.2. Развитие фазовой структуры теории

Таким образом, на текущем этапе своего становления теория Th может быть представлена следующим выражением:

$$Th(t) = [D_s(t) \cup Th^T_s(t) \cup Th^K_s(t) \cup Th^I_s(t) \cup \{Th^F_{sk}(t)\}] \leftrightarrow M_{sq}(t), \quad (3.1)$$

где знак « \leftrightarrow » указывает на связь между теорией и моделями, которые теория помогла построить.

Кроме того, необходимо отметить, что переход от стадии к стадии может привести к необходимости определения полезных абстракций для уже существующих теоретических конструктов, систематизация которых помогает построить соответствующую метатеорию, что также показано на рис. 3.2. В этом случае теория включает в себя метатеоретические компоненты, которые являются особым вкладом в развитие Th и его систематизацию.

В методологии науки термин «метатеория» понимается как теория логико-семиотических свойств отображаемой теории. В научной практике метатеории помогают изучать: свойство логической непротиворечивости теории, синтаксическую и семантическую целостность; логическую зависимость или независимость предложений и другие теоретические особенности.

Теория бесполезна без неявных или явных интерпретаций её конструктов, когда они используются для достижения определённой цели. В любом её варианте интерпретация предполагает переход от конструкта C к соответствующей сущности в реальности, например, к определённому объекту Ob определённой предметной области. Такие переходы возможны тогда, когда ранее между конструктами теории (или метатеории) и

объектами $\{Ob\}$ соответствующей предметной области определялось адекватное отображение R . Подобное положение дел показано образно на рис. 3.3.

Следует отметить, что результаты интерпретаций, например, приписывание свойств $\{g_c\}$ для конституирования $\{B_v\}$ и отношения $\{r_d\}$ между ними, находят своё выражение в формах предикаций, используемых в текстовых конструктах $\{C_r\}$ и $\{C^M_r\}$ теории и метатеории соответственно. Конечно, обеспечение правильности определённого присваивания может быть только с использованием идентификации или измерения подходящими средствами $\{I_m\}$ и $\{I^M_m\}$.

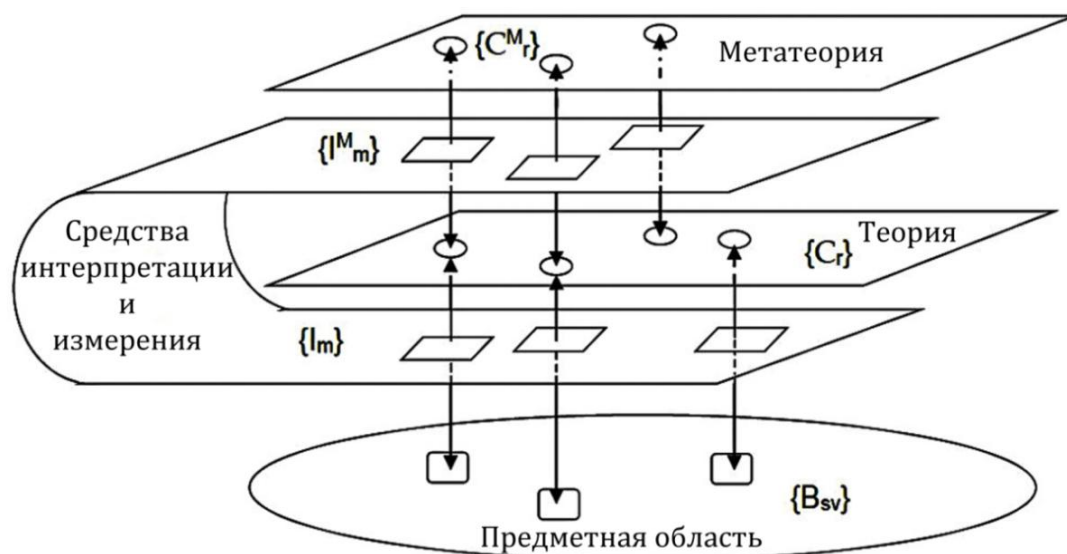


Рис. 3.3. Интерпретация конструктов теории

Следует отметить, что обычно отображение находит своё выражение в предикативных формах, связывающих объекты Ob с их свойствами $\{g_c\}$ или с отношениями $\{r_d\}$ между ними. Конечно, обеспечение правильности такой привязки может быть достигнуто только с использованием идентификации или измерения подходящими средствами $\{I_m\}$ и $\{I^M_m\}$.

В разных ситуациях различные средства идентификации и измерения обеспечивают осуществление предикации. Эти средства включают в себя как человеческие способности, основанные на естественных чувствах (например, воображение), так и различные искусственные устройства и практики. Специализированные средства (различные датчики, измерительные приборы, устройства и механизмы распознавания изображений) значительно расширяют потенциал предикативных атрибуций при создании теорий и их использовании.

К месту отметить, что адекватность предикации, её точность и качество принципиально зависят от средств идентификации и измерения, а также от действий и методов их использования. Одним из наиболее

полезных способов прогнозирования является «эксперимент». В принципе, можно предположить, что каждый акт предикации является результатом проведённого эксперимента, зрелость которого определяется схемой выполненных действий.

Среди различных видов экспериментов центральное место занимают научные эксперименты, имеющие долгую историю, в ходе которой их схемы постоянно совершенствовались. Ещё раз отметим, что эксперименты являются средством идентификации и измерения.

Если для конструкта C_r теории $Th(t)$ был проведен необходимый эксперимент, направленный на проверку предикации, то, другими словами, было бы подтверждено соответствие:

$$C_r \overset{\pi}{\leftrightarrow} B_{sv}, \quad (3.2)$$

тогда этот результат (где π указывает на предикативное отношение) может многократно использоваться при соответствующих условиях, когда это будет необходимо.

В общем случае результат предикации для конструкта C_r может быть получен с помощью ряда различных экспериментов, и эта особенность позволяет квалифицировать соответствие

$$\{C_r\} \overset{\pi}{\leftrightarrow} \{B_{sv}\}, \quad (3.3)$$

как многозначное отображение конструктов $\{B_{sv}\}$ на конструкте $\{C_r\}$ теории.

Перейдём к деталям отображений, которые регистрируют соответствие между конструктами предметной области, её теориями $Th(t)$ и метатеорией $ThM(t)$ для $Th(t)$. В целом, множество конструктов в предметной области включает объекты (естественные и искусственные), связанные множества объектов в определённых условиях, процессы и их комбинации, а также действующих лиц, участвующих в процессе.

Как было сказано выше, предикативное отображение любого объекта Ob_t в теории $Th(t)$ заключается в его спецификации как теоретического конструкта C_r , то есть в описании его свойств и отношений с другими компонентами предметной области. В теории основными единицами любого описания (основанного на предикации) являются следующие два типа простых предложений (Simple sentences (SS)), выраженных с использованием форм Бэкуса-Наура (БНФ):

$$\begin{aligned} SS^g &:= g(< Ob >) < As > \\ SS^f &:= < r > (< Ob_1 >, < Ob_2 >) < As >, \end{aligned} \quad (3.4)$$

где предикаты g обозначают свойство объекта Ob в первом предложении, r представляет отношение между Ob_1 и Ob_2 (второе предложение), As выражает ассоциативный компонент, соответствующий контексту предикации.

Используя эти основные единицы предикации, предикативное представление объекта Ob может быть представлено выражением:

$$C_r = \pi(Ob) = S(\{SS_{ti}^g\}, (\{SS_{ij}^r\}, t), \quad (3.5)$$

в котором S фиксирует объединение простых предикатов в одну единицу C_r , а время t отражает динамику становления конструкта C_r . Эти разложения позволяют описывать любой компонент в области теории как разложение на основе простых предикатов.

Выражение (3.5) показывает, что в QA-интерпретации следует различать простые и сложные предикативные представления, которые состоят из простых предикатов, объединённых в единое целое таким образом, чтобы они успешно применялись при решении задач соответствующей предметной области.

3.1.3. Содержательно эволюционный подход к становлению теории

Создание проекта $P(t)$ любой АС реализуется в процессе решения множества разнообразных и разнородных, но связанных между собой задач, формирующих определённую предметную область SASIS для совместной профессиональной деятельности.

В последнее время для создания АС разработчики чаще всего используют итерационный процесс проектирования, в ходе которого разрабатывают версию для версии проекта, каждая из которых повышает уровень детализации и находит своё описание в системе нормативных документов проектной документации $PD(t)$. На ранних стадиях проектирования документация имеет текстовую форму $S(\{T_i\}, t)$, которая дополнительно включает графические блоки (обычно в блочно-линейном виде) и таблицы. На последующих этапах программные коды и технологические артефакты включаются в проектную документацию. Общая схема становления проектной документации приведена на рис. 3.4.

Формирование – это динамический процесс, в ходе которого документация $PD(t)$ расширяется за счёт включения информационных блоков, унаследованных из следующих источников:

- Необходимые знания из предметной области;
- Артефакты технологий (например, нормативные текстовые шаблоны и диаграммы);

– Соответствующие и необходимые стандарты.

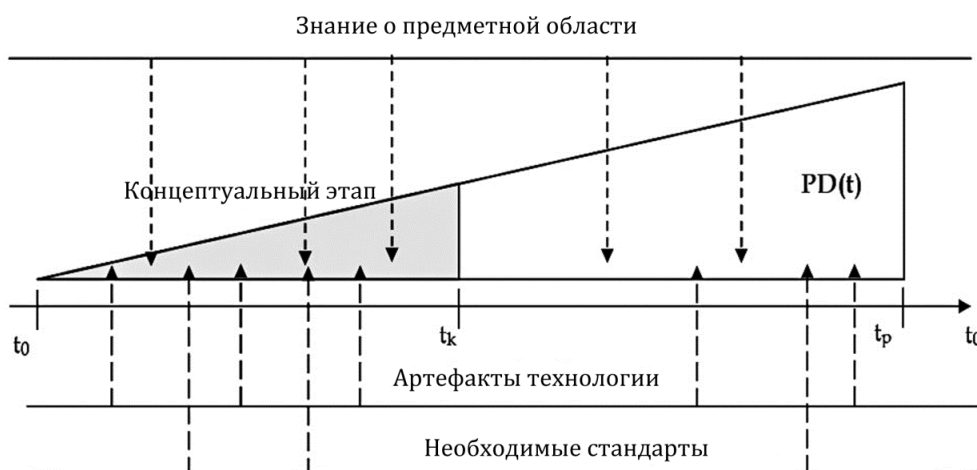


Рис.3.4. Формирование проектной документации

При разработке документации особое внимание уделяется систематизации текстов $S(\{T_i\}, t)$, достижение которой регламентируется огромным количеством профессиональных стандартов. Поэтому ничто не мешает связать каждый проект $P(t)$ с его теорией $Th^P(t)$. Следует отметить, что развитие современных АС и особенно их семей является наукоёмкой деятельностью, включающей научную составляющую. Кроме того, необходимо отметить, что при разработке новой АС и отсутствии теории её проекта целесообразно ориентироваться на формирование теории, показанной на рис. 3.4.

В самом общем смысле любая теория – это система предложений, объединённых в текстовые единицы, отношения между которыми определяют структуру теории. Для использования теории характерно совмещение её теоретических конструктов с конструктами соответствующей метатеории, подходящими графическими единицами и таблицами. Аналогичная комбинация строится на концептуальном этапе проектирования АС и соответствующих состояний $PD(t)$, что можно понимать как объединение $Th^P(t)$, его метатеории $Th^{PM}(t)$, текстовых моделей (используемых текстовых шаблонов и таблиц) и графических моделей. В текущем состоянии $PD(t)$ объединение обеспечивается использованием естественно профессионального языка $L^P(t)$ проекта, и этот язык также развивается в ходе проектирования.

Возникает вопрос: «Что значит определять и обеспечивать развитие проекта $PD(t)$ в его динамике?»

Согласно подходу SEMAT, любая система является результатом работы, выполненной командой разработчиков, взаимодействующих с заинтересованными сторонами. В свою очередь, работа определяется как

система действий, выполнение каждого из которых приводит к соответствующему «продукту труда». В управлении проектами и программировании такие действия обычно называются задачами.

Как естественно искусственные сущности, задачи имеют следующие особенности:

1. Они ориентированы на достижение определённых целей, которое должно быть подтверждено полученными результатами и их проверкой.

2. В общем смысле, жизненный цикл задачи, как и проблемы, включает в себя творческие действия, которые помогают заполнить определённые пробелы. В этих случаях задача должна быть решена, как правило, таким образом, чтобы было возможно повторное использование этого решения в будущем.

3. Описание задачи, обеспечивающее её повторное использование, можно интерпретировать как модель соответствующего прецедента, имитирующего единицу опыта.

Сказанное о задачах позволяет ответить на поставленный выше вопрос следующим утверждением: решения плановых и ситуационных задач обеспечивают динамику развития как AC, так и PD(t). Далее мы будем часто использовать термин «задача» в значении, определённом выше.

Таким образом, любая решённая задача Z_i оставляет текстовые и другие следы в PD(t) и его компоненте $Th^P(t)$ как системе $S(\{T_i\}, t)$. Предположим, что текстовая часть следов T_i входит в $S(\{T_i\}, t_j)$ как её инкремент $\Delta S(Z_i, t_j)$ в момент времени t_j . Тогда эта система текстов может быть выражена в виде

$$S(\{T_i\}, t_j) = S(T_0, t_0) \cup (U_i \Delta S(Z_i, t_j)), \quad (3.6)$$

где T_0 – это текстовое описание, с которого проектировщики начинают свою работу над проектом P(t) в момент времени t_0 , $\{Z_i\}$ – это набор решённых задач, а t_j указывает на текущий момент времени.

Поскольку выражение соответствует текстовой структуре теории $Th^P(t)$, его можно представить в виде:

$$Th^P(t_j) = Th^P(t_0) \cup (U_i \Delta Th^P(Z_i, t_j)), \quad (3.7)$$

где Z_i можно интерпретировать как определённое правило в теории $Th^P(t)$. Теория $Th^P(t_j)$ развивается с помощью набора правил $\{Z_i\}$, и ни одно из этих правил не должно вносить никаких противоречий в теорию.

Отсутствие противоречий возможно тогда, когда для любого

$\Delta Th^P(Z_i, t_j)$ его включение в теорию не изменяет прежней структуры и содержания теории $Th^P(t_{j-1})$. Если это так, то по типологии теорий теорию $Th^P(t)$ можно квалифицировать как теорию содержательно эволюционного типа.

Любая теория такого типа развивается в соответствии с принципом аддитивности как система $s(\{T\}, t)$ текстов T , каждый из которых встраивается в $s(\{T\}, t)$, начиная с начального состояния $s(T_0, t_0)$, только при наличии достаточных обоснований для встраивания.

Принцип аддитивности выражает соответствующую последовательность в его применении к созданию теории $Th(t)$. Материализация этого принципа соответствует методу пошаговой детализации, который широко используется в программировании и практике проектирования АС. Следует отметить, что уточнение теоретического конструкта C_r возможно при использовании следующих правил проверки:

1. Уточнить некоторый компонент конструкта C_r , указав структуру этого компонента или изменив его структуру, чтобы не изменилась функциональная сущность компонента (уточнение в «глубину»);

2. Добавить новый конструкт C_p или изменить его прежний вариант путём сохранения спецификаций предыдущего состояния теории (уточнение в «ширину»).

Таким образом, перед встраиванием любого инкремента $\Delta Th^P(Z_i, t_j)$ в текущее состояние теории $Th^P(t)$, инкремент необходимо проверить на его соответствие принципу аддитивности. В практике проектирования систем существуют случаи, когда проверка $\Delta Th^P(Z_i, t_j)$ указывает на нарушение этого принципа, но решение задачи соответствия Z следует включать в проект. Подобные случаи обычно обрабатываются специализированными рабочими процессами «Управление изменениями». После обработки документации, соответствующая теория будет без противоречий.

В заключение данного подраздела следует отметить, что фазовую структуру становления теории целесообразно использовать для создания содержательно эволюционной теории $Th^P(t)$, или, что то же самое, теории концептуального проектирования системы. Центральное место в этой теории занимает фаза описания, возникшая на основе дотеоретической фазы. Потенциал $Th^P(t)$ может быть существенно увеличен, если будут дополнительно разработаны другие фазы становления теории.

3.2. Экспериментирование в человеческой деятельности

3.2.1. Особенности экспериментальной деятельности

В конце пункта 3.1.1 мы ссылались на статью (Perry, 2014) [20], автор которой подчеркнул необходимость построения теорий D и E, отражающих развитие системы и оценку как процесса разработки, так и её результатов. В программной инженерии существует богатый опыт оценивания, который также должен быть переосмыслен как способы теоретизации. Перспективным направлением такого переосмысления является ориентация на способы оценки в науке, особенно способы научного эксперимента, адаптированные под особенности человеко-компьютерной деятельности.

Неявные эксперименты повседневной жизни людей являются корнями научного экспериментирования. Типичными причинами такого рода экспериментов являются предсказание результатов задуманных действий или определение того, какое действие или комплекс действий может привести к задуманным результатам.

История научного экспериментирования началась с позиции Фрэнсиса Бэкона (1561–1626), который утверждал, что теоретические дедуктивные утверждения должны обязательно проверяться на практике. В последующие столетия и до сих пор научные эксперименты были и остаются незаменимым средством для прогресса науки и практики.

В своём общем понимании, научный эксперимент – это метод эмпирического исследования, при котором учёный воздействует на изучаемый объект с помощью специальных материальных средств (экспериментальных установок и приборов) с целью получения необходимой информации о свойствах и особенностях этих объектов или явлений.

Приведём некоторые определения эксперимента:

1. Dictionary.com [65],

1.1. Тест, испытание или пробная процедура; действие или операция с целью открытия чего-то неизвестного или проверки принципа, предположения и т.д.

1.2. Тест или исследование, для предоставления доказательства за или против гипотезы: научный эксперимент.

1.3. Акт проведения такого расследования или теста.

2. Businessdictionary [60]:

Метод исследования для проверки различных предположений (гипотез) методом проб и ошибок в условиях, установленных и контролируемых исследователем. Во время эксперимента одно или несколько условий (называемых независимыми переменными) могут изменяться организованным образом, и влияние этих изменений на

связанные условия (называемые зависимыми переменными) измеряется, записывается, проверяется и анализируется для получения вывода.

3. Oxford Dictionary [77]:

Научная процедура, предпринятая для того, чтобы сделать открытие, проверить гипотезу или продемонстрировать известный факт.

4. Free Dictionary [83]:

Тест в контролируемых условиях, который делается для демонстрации известной истины, проверки обоснованности гипотезы или определения эффективности чего-то ранее не испытанного.

5. Merriam-Webster [85]:

Операция или процедура, выполняемая в контролируемых условиях с целью обнаружения неизвестного эффекта или закона, проверки или установления гипотезы или демонстрации известного закона.

Представленные определения подсказывают, что эксперименты являются подходящим средством для оценки теорий D-типа. Поэтому в данном разделе мы более подробно раскрываем такой вид человеческой деятельности.

Любое экспериментирование имеет цель, в соответствии с которой учёные выделяют следующие типы экспериментов:

1. Преобразующий (творческий) эксперимент требует активного изменения структуры и функции исследуемого объекта, например, в соответствии с заявленной гипотезой. Такого рода эксперименты могут быть направлены на образование новых связей и отношений между компонентами объекта или между исследуемым объектом и другими объектами. В соответствии с открытой тенденцией развития объекта исследования, исследователь намеренно создаёт условия, которые должны способствовать формированию новых свойств и качеств объекта.

2. Констатирующий (декларирующий) эксперимент используется для проверки любого исходного предположения. В ходе данного эксперимента исследователь констатирует наличие некоторой связи между воздействием на объект исследования и результатами, подтверждающими существование тех или иных фактов.

3. Контрольный эксперимент решает задачу обеспечения контроля над исследуемым объектом, управление объектом с использованием воздействующих факторов при исследовании изменения его состояния в зависимости от воздействия.

4. Поисковый эксперимент проводится в том случае, если классификация факторов, влияющих на изучаемое явление вследствие отсутствия достаточных предварительных (априорных) данных сложно. В результате поискового эксперимента устанавливается значимость факторов, а несущественные факторы исключаются. Зачастую это лишь начальный этап в серии экспериментальных исследований.

5. Решающий эксперимент проверяет обоснованность основных положений фундаментальных теорий в том случае, когда две или несколько гипотез одинаково согласуются со многими явлениями. Эта последовательность приводит к трудностям, которые рассматриваются как правильная гипотеза. Решающий эксперимент выявляет факты, которые согласуются с одной из гипотез и противоречат другой.

Выделенные цели являются типичными для проектирования системы, так что эксперименты указанных видов могут быть полезны не только для оценки теоретических конструкций, но и для решения задач проекта.

Любой эксперимент начинается с вопроса, ответ на который должен быть получен в процессе экспериментов, которые будут проведены с помощью экспериментальной установки в определённых условиях. Поэтому при проектировании системы следует также разработать задуманный эксперимент для его эффективного проведения. Таким образом, намерение провести какой-либо полезный эксперимент будет предварительно включать:

1. Задумывание (изобретение) эксперимента с выбором его вида;
2. Формулировка задачи экспериментирования и концептуальное решение этой задачи;
3. Разработка соответствующей установки с учётом условий проведения экспериментов;
4. Разработка плана и техники проведения экспериментов.

Любая из этих работ существенно зависит от объектов, с которыми будет проводиться эксперимент. При проектировании АС пространство экспериментирования представляет собой компьютеризированную среду, с которой взаимодействуют проектировщики, решая поставленные задачи или задачи, возникающие ситуационно.

Различие между объектами, используемыми в эксперименте, позволяет различать мысленные и компьютерные эксперименты. Объектами ментальных (мысленных) экспериментов являются мысленные модели объектов или явлений (чувственные образы, образные и символические модели, символические модели). Иногда этот термин можно использовать для обозначения мысленного эксперимента: идеализированного или воображаемого эксперимента. Мысленный эксперимент – это форма мыслительной деятельности познающего субъекта, в ходе которой в воображении воспроизводится реальная структура эксперимента.

В компьютерных экспериментах проектировщики работают с объектами, которые являются программируемыми моделями компьютерной среды. Главное отличие компьютерного эксперимента от ментального в том, что этот эксперимент является формой объективной

связи сознания с внешним миром, в то время как мысленный эксперимент является специфической формой теоретической деятельности субъекта.

Следует отметить, что современное состояние искусственного интеллекта является богатейшим источником средств и методов моделирования мысленных сущностей и механизмов, и этот источник открывает возможности для управляемой автоматизации мысленного экспериментирования.

После подготовки эксперимента его можно проводить в соответствии с разработанным планом, используя настройки и технику проведения экспериментов. Эта часть процесса эксперимента включает следующие шаги:

1. Определите результаты и оцените их достоверность.
2. Определить, подтверждают или опровергают ли результаты заявленную гипотезу.
3. Описать проведённый эксперимент в том виде, который поможет его повторно использовать любому проектировщику команды, разрабатывающей систему.
4. Перейти от эмпирического исследования к логическому обобщению, анализу и теоретической обработке полученного фактического материала.

Завершая этот подраздел, отметим, что если параллельно с созданием теории $Th^P(t)$ проектировщики строят соответствующую теорию экспериментирования, то эта работа приводит к расширению теории $Th^P(t)$ на уровне идентифицируемой измерительной фазы.

3.2.2. Разработка концепций и экспериментирование

В этой книге среди направлений, которые фокусируют свои интересы на экспериментировании с концептуальными объектами, мы выделяем исследования и практики, которые касаются предметной области «разработка концепций и экспериментирование (CD&E)».

Основные идеи, содержащиеся в направлении CD&E, раскрываются в таких публикациях его основателей Дэвида С. Альбертса и Ричарда Э. Хейса, как «Campaigns of Experimentation» и «Understanding Command and Control» [79].

Содержание CD&E раскрывает следующие определения:

1. Термин «разработка концепций и экспериментирование обозначает метод, который позволяет нам путём эксперимента предсказать, способны ли определённые концепции, теоретические конструкторы, подсистемы или системы отвечать требованиям, предъявляемым процессом трансформации, и могут ли они быть конструктивно интегрированы в общую систему» [59].

2. CD&E – это научно-ориентированная процедурная модель, предназначенная для оптимизации возможностей вооружённых сил на всех уровнях военных действий и миссий, особенно на фоне проблем безопасности и политики в 21 веке. CD&E использует множество различных научных процедур, методов и инструментов. CD&E помогает вооружённым силам быстро и целенаправленно определять, анализировать и ликвидировать пробелы в потенциале. Это касается, в частности, концептуального проектирования, материалов, подготовки, планирования миссий или структуры командования [71].

3. CD&E – «это применение структуры и методов экспериментальной науки к задаче развития будущего военного потенциала. CD&E – это перспективный процесс разработки и оценки новых концепций, прежде чем выделять значительные ресурсы. CD&E – это процесс определения наилучшего решения не только с технической точки зрения, но и для возможных решений проблем, связанных с доктриной, организацией, обучением и материалами для достижения значительных успехов в будущих операциях. CD&E – это способ думать о своём будущем, прежде чем тратить деньги» [87].

4. CD&E – «метод, который позволяет нам исследовать и прогнозировать, путём экспериментов, будут ли новые концепции, которые могут повлиять на людей, организацию, процесс и системы, способствовать целям трансформации и будут ли они вписываться в более широкий контекст, или, точнее, соответствовать ему».

В CD&E термин «концепция» соответствует «общему предлагаемому решению, которое охватывает всеобъемлющее и твёрдое представление о том, как может быть решена проблема или как можно воспользоваться возможностью с целью выполнения требования о потенциале в данном контексте. Концепция в соответствии с этим определением включает описание проблемы, которую необходимо решить (проблемы или возможности), почему она должна быть решена (требование к возможности) и как она должна быть решена (основная идея)» [61].

В схематичной форме сущность CD&E продемонстрирована на рис. 3.5 из книги (Alberts and Hayes, 2005) [79].

Любая реализация CD&E приводит к определённому проекту, реализация которого должна обеспечивать итеративное переплетение разработки концепции и экспериментирования. Схема такой работы показана на рис. 3.6.

Итеративное переплетение, управляемое результатами экспериментов и обратной связью, продолжается до тех пор, пока не достигнет желаемой способности. Таким образом, в этом процессе за каждым шагом экспериментирования следует шаг, на котором продолжается оформление

концепции, корректировка и уточнение по всем выбранным направлениям развития.

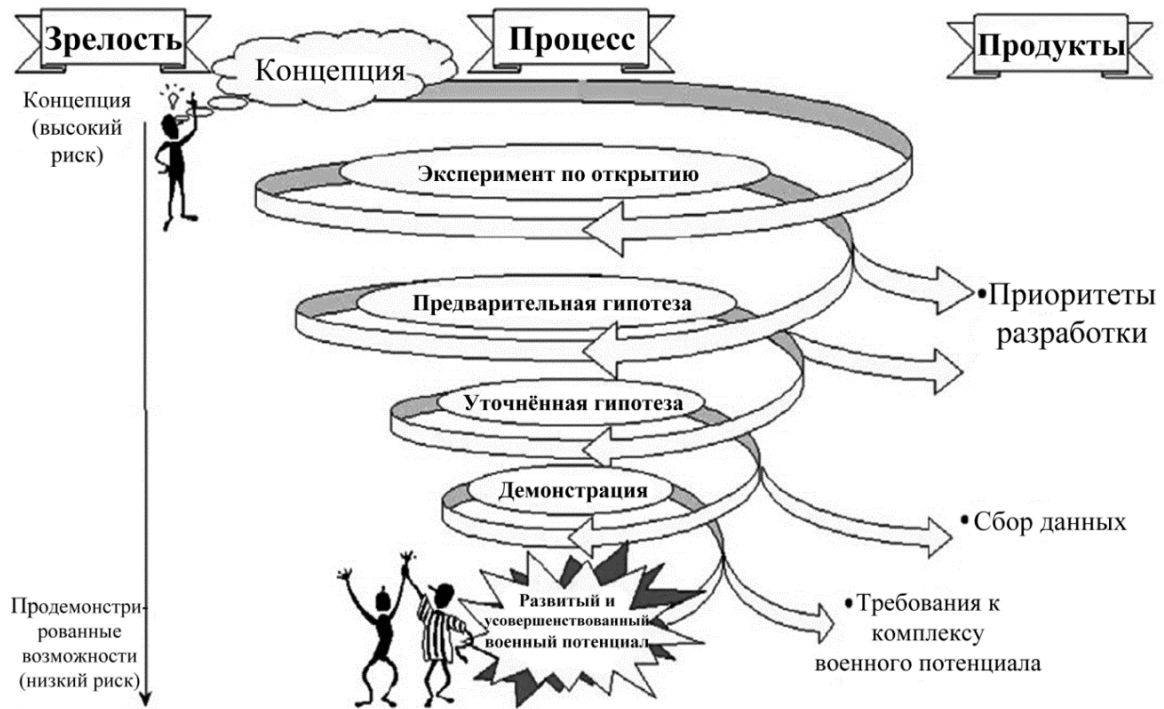


Рис. 3.5. Сущность CD&E

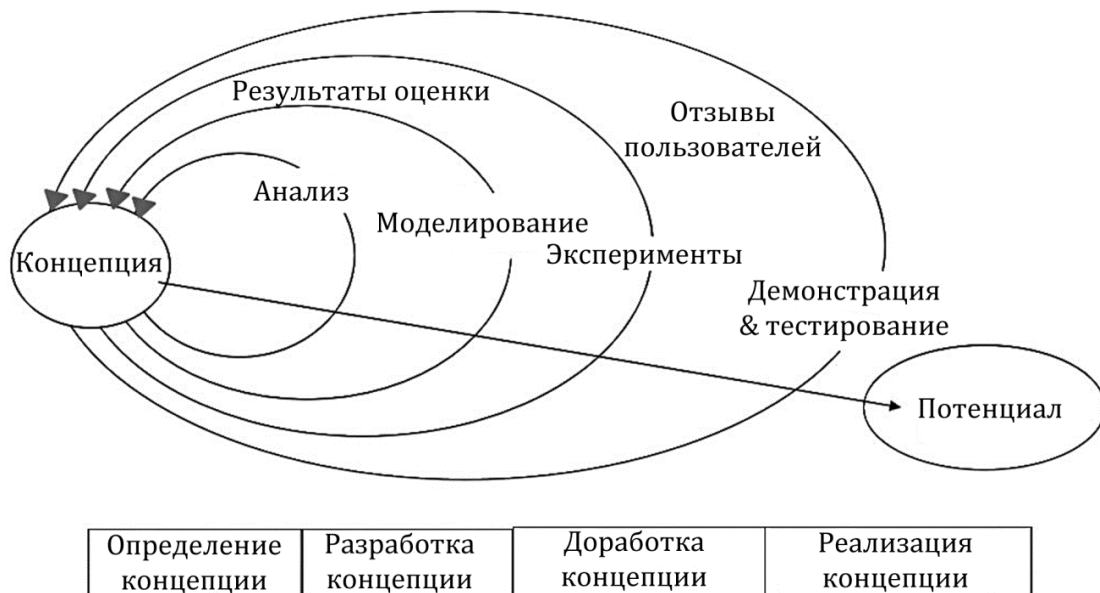


Рис. 3.6. Итеративный процесс

Любой используемый эксперимент имеет собственный жизненный цикл, типичная структура которого показана на рис. 3.7, который заимствован из книги (Alberts and Hayes, 2005). Эта схема подчёркивает,

что на основе используемых знаний необходимо выбрать или построить соответствующую модель, поскольку процесс CD&E ориентирован на модель. Используемая модель помогает создать выполненный прототип, который будет протестирован в эксперименте.

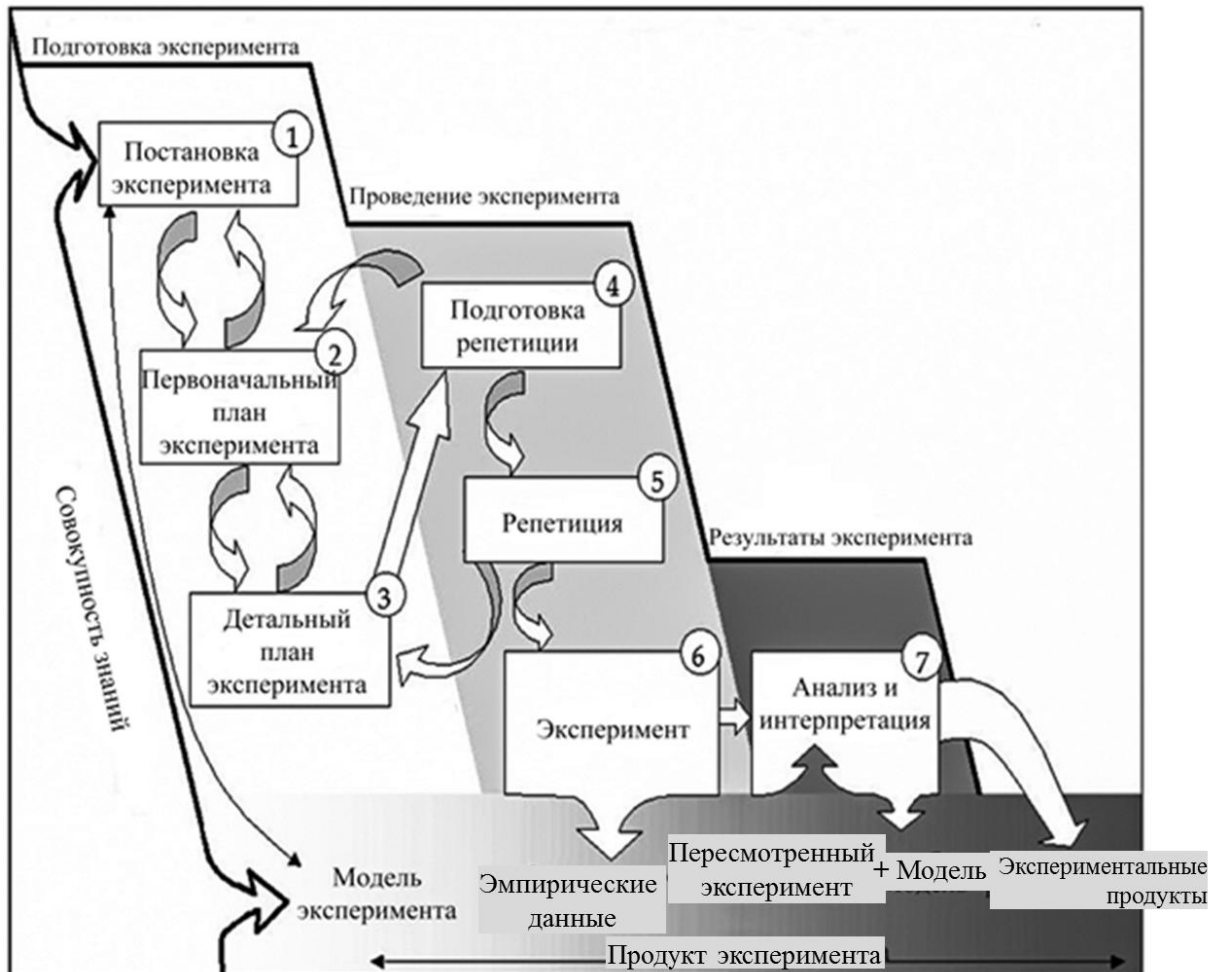


Рис. 3.7. Жизненный цикл эксперимента

Отметим, что термины, используемые в блочно-линейных схемах на рис. 3.5–3.7, проясняют суть как CD&E процессов, так и место и роль экспериментов в разработке концептов. Схемы помогают понять применение экспериментов в концептуальном пространстве в любых приложениях, включая область разработки АС. Эти схемы помогут понять наши версии эксперимента, которые мы предлагаем ниже для их использования в концептуальном проектировании систем.

На практике любой реализованный процесс состоит из рабочих процессов, каждый из которых объединяет в себе ряд техник. Любое используемое сочетание процесса можно оценить с точки зрения его зрелости. Существует разработанная структура (Wiel et al., 2010) [86], позволяющая оценить уровень зрелости с помощью следующих метрик:

1. Первый уровень. Введение: сформулирована принципиальная идея.

2. Второй уровень. Эксплуатационная валидность: заказчик выражает потребность и предъявляет требования к разработке принципиальной идеи.

3. Третий уровень. Применимость: положительная идентификация пустоты понятия и исследование применимости понятия в пределах операционной среды.

4. Четвертый уровень. Идентификация: компонентное концептуальное решение, способное удовлетворить заявленные требования.

5. Пятый уровень. Осуществимость: аналитическое доказательство того, что концептуальное решение действует так, как требуется в моделируемой среде.

6. Шестой уровень. Прототипирование: доказательство принципа построения интегрированных концептуальных решений.

7. Седьмой уровень. Проверка прототипов: решения могут работать в лабораторных условиях.

8. Восьмой уровень. Валидация решений: решения работают в оперативной среде и проверяются с помощью военной приёмки.

9. Девятый уровень. Интеграция решения: разработанное блочное решение интегрировано в полное решение, удовлетворяющее требованиям по спектру DOTMLPFI: Doctrine (Доктрина), Organisation (Организация), Training (Обучение), Materiel (Материал), Leadership (Лидерство), Personnel (Персонал), Facilities (Услуги), Interoperability (Совместимость).

10. Десятый уровень. Демонстрация: полная реализация во всех областях DOTMLPFI.

Чтобы гарантировать, что проект CD&E может быть завершён и что результирующие продукты могут быть одобрены – на каком бы уровне – единственная необходимость для проекта заключается в обеспечении достижения определённого уровня зрелости. Проект может даже заранее объявить, на какой уровень они нацелены.

Следует отметить, что мы провели анализ отчётов [84] подготовленных в нидерландской организации прикладных научных исследований (один из лидеров разработки приложений CD&E). Исследования, представленные в докладах, ориентируются на устоявшуюся практику CD&E. Таким образом, отмеченная в предыдущих параграфах научная новизна распространяется на сравнение с результатами исследований и организации, а также следующих организаций: Defence Research and Development Canada (Канада) [64] и шведское агентство оборонных исследований (Швеция) [82], доклады о которых также были рассмотрены.

3.2.3. Дизайн-мышление

Ещё одной предметной областью научно-исследовательской и опытно-конструкторской работы, которая фокусирует свои интересы на экспериментировании с концептуальными объектами, является «дизайн-мышление». Эта область имеет много граней, особенности которых можно раскрыть следующим набором определений:

1. «Особенность дизайн-мышления заключается в том, что рабочие процессы проектировщиков могут помочь нам систематически извлекать, обучать, изучать и применять эти антропоцентрические методы для решения проблем творческим и инновационным способом – в наших проектах, в нашем бизнесе, в наших странах, в нашей жизни» [72].

2. «Дизайн-мышление – это ориентированный на человека подход к инновациям, который опирается на инструментарий проектировщика для интеграции потребностей людей, возможностей технологий и требований к успеху бизнеса» [75].

3. «Дизайн-мышление можно описать как дисциплину, которая использует восприимчивость проектировщика и методы, чтобы сопоставить потребности людей с тем, что технологически осуществимо и что жизнеспособная бизнес-стратегия может преобразовать в потребительскую ценность и рыночные возможности» [75].

4. Дизайн-мышление использует процессный подход к решению задач, и, как любой процесс оно включает в себя ряд шагов, которые выполняются в определённом порядке для достижения цели. В этом случае цель состоит в том, чтобы определить решение, которое способно привести к успеху, может быть выполнено своевременно и, вероятно, будет принято всеми заинтересованными сторонами [80].

5. Дизайн-мышление – это итеративный подход к решению проблем, который намеренно ищет людей с разными перспективами, знаниями, навыками и опытом и заставляет их работать вместе, чтобы создать практическое решение реальной проблемы [80].

6. Дизайн-мышление – это ориентированный на человека, итеративный процесс решения проблем через открытие, создания идей и экспериментов, который использует различные методы проектирования, чтобы получить представление и получить инновационные решения практически для любого типа организационных или бизнес-задач. Этот метод сочетает в себе как аналитический, так и творческий подходы к генерации решений [67].

7. Дизайн-мышление – это дисциплина переосмысления и создания устойчивых процессов, систем, структур и бизнес-моделей на основе эмпатического и прогностического гуманистического подхода с использованием современных блестящих технологий и

междисциплинарного мышления, которое учитывает неудовлетворённые, желаемые и непредвиденные потребности всех заинтересованных сторон.

8. Дизайн-мышление – это подход к инновациям, который может быть применён во всех сферах человеческой деятельности. Дизайн-мышление относится не к формальному пошаговому процессу, а к структуре и мышлению. Он сосредоточен на действии, ориентированном на человека, точке зрения и способе непрерывного экспериментирования. Основная идея заключается в том, что, глубоко понимая потребности клиентов, появятся возможности для инноваций. Эти идеи могут быть дополнительно уточнены с помощью быстрых прототипов и итераций, чтобы привести к прорывным результатам [63].

Таким образом, в более общем плане дизайн-мышление – это методология решения инженерных, деловых и других задач, основанная на творческом, а не аналитическом подходе. В отличие от аналитического мышления, дизайн-мышление – это творческий процесс, в котором иногда самые неожиданные идеи приводят к лучшему решению выявленной проблемы.

В этом процессе проектировщик использует мышление, сочетающее следующие его стороны: логическую и образную; рациональную и эмоциональную; интуитивную и конструктивную; аналитическую и синтетическую. Биполярная ориентация дизайн-мышления обладает рядом присущих ей свойств:

1. Интуитивное и творческое понимание предмета дизайна и целостности видения;
2. Способность работать с деталями, не теряя из виду целое;
3. Умение работать с неразвитыми и неосознанными проблемами, а также с непрограммируемыми данными, требующими интуитивного «схватывания», а не логического построения;
4. Согласованность и единство аналитических и синтетических компонентов приводят к акту синтеза;
5. Способность предсказывать;
6. Соотнесение творческого воображения с реальностью;
7. Умение визуализировать абстрактные понятия и идеи;
8. Умение работать в команде.

Процесс дизайн-мышления схематично показан на рис. 3.8

Данная схема представляет типичное разделение процесса на пять этапов, которые соответствуют следующим действиям:

1. Empathize («сопереживание, эмпатия»), которое фокусируется на выявлении и понимании явных и неявных потребностей, которые имеют потенциальные клиенты или пользователи инновационных проектных решений.

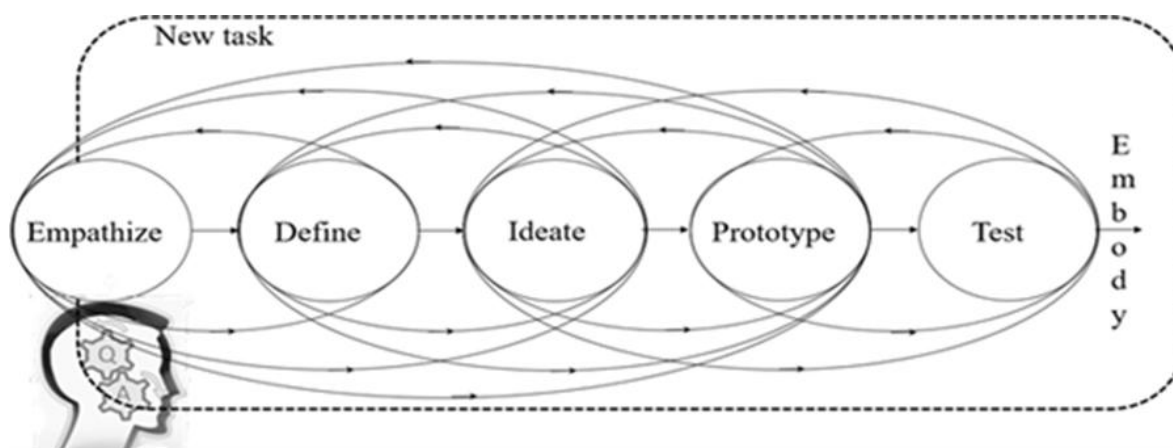


Рис. 3.8. Процесс дизайн-мышления

2. Define («определение»), которое определяет новые возможности, рассматривая вещи по-разному с точки зрения потребностей и понимания пользователей.

3. Ideate («идея»), которое направлено на генерирование и изучение идей и решений с учётом различных точек зрения за короткий промежуток времени.

4. Prototype («прототип»), которое обеспечивает визуальное представление одной или нескольких выбранных идей в демонстративных целях.

5. Test («тестирование»), основная цель которого – оценить найденное решение и определить, как пользователь понимает и использует соответствующие идеи.

Все эти действия переплетаются в пошаговом процессе, который повторяется за несколько итераций с использованием обратных связей. Этот процесс управляется вопросами и ответами, проверяемыми на их соответствие целевым потребностям.

3.2.4. Особенности мысленного экспериментирования

Реальный эксперимент обычно имеет ограниченные масштабы. Иногда он ограничивается по экономическим причинам или из-за своей сложности. Зачастую материальный эксперимент не даёт желаемого результата, поскольку его масштаб ограничен уровнем развития знаний и технологий. Просто мысленный эксперимент, в котором логическое мышление и творческое воображение исследователя сочетаются с экспериментальным и теоретическим материалом, позволяет оттолкнуться от реальности и пойти дальше – понять и исследовать то, что раньше казалось неразрешимой загадкой. Во всех случаях, когда для познания наиболее глубокой сущности необходим эксперимент, исследователь

старается провести мысленный эксперимент специально для высокой степени абстрагирования от реальных условий.

Мысленные эксперименты создаются не совсем произвольно. Это мыслительные операции, отвечающие определённым требованиям и принципам доказанной научной теории. Как и в любой другой теоретической конструкции, в мысленном эксперименте все операции должны подчиняться некоторым правилам, вытекающим из знания объективных законов науки. Это требование обеспечивает высокую степень достоверности знаний, полученных в процессе исследования.

Мысленный эксперимент – это эксперимент в области сознания, в котором ведущая роль принадлежит мышлению. Это определяется его субъективной природой. Однако тот факт, что мысленный эксперимент реализуется на уровне сознания, говорит о том, что его содержание объективно. На практике некоторые части ментального эксперимента проводятся за пределами сознания, что может быть представлено схемой, показанной на рис. 3.9.

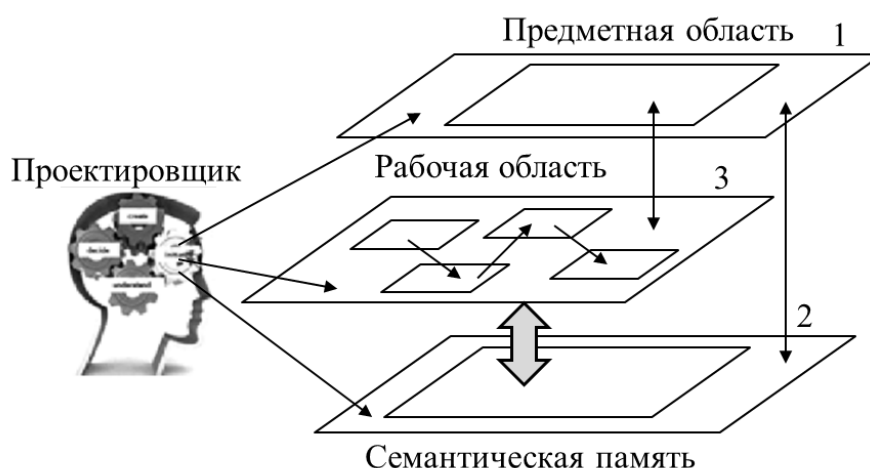


Рис. 3.9. Условия мысленного экспериментирования

На данной схеме проектировщик помогает мысленным операциям эксперимента путём отображения объектов предметной области на рабочей области, где объекты используются в качестве концептуальных моделей с описаниями в соответствующей семантической области.

Оценивая мысленный эксперимент, нельзя рассматривать его как законченное знание; в данном случае он выступает как простая иллюстрация. Также его содержание нельзя сводить только к размышлению о планировании материального эксперимента (хотя ему всегда предшествует материальный эксперимент). Мысленный эксперимент – это скорее продолжение и обобщение, схематизация последнего, а не наоборот.

Ценность мысленного эксперимента, во-первых, в том, что он позволяет исследовать ситуацию практически неосуществимую, хотя и возможную в принципе. Во-вторых, в некоторых случаях это позволяет создавать знания и производить проверку их истинности, не прибегая к материальным экспериментам. Однако мысленный эксперимент является как прямым, так и модельным, и эти его стороны в конечном итоге требуют практической проверки результатов. Если в материальном эксперименте его процесс уже служит подтверждением истинности посылок, то же самое нельзя сказать и о мысленном эксперименте: окончательная оценка мысленного эксперимента может быть получена только в процессе проверки результатов на практике.

Подводя итог, можно описать мысленный эксперимент как эвристическую операцию со следующими особенностями:

1. Это когнитивный процесс, который принимает структуру реального эксперимента;

2. Вся цепочка рассуждений реализуется на основе зрительных образов;

3. Мысленное экспериментирование ассоциируется с процессом идеализации;

4. По своей логической структуре мысленный эксперимент является гипотетико-дедуктивной конструкцией;

5. Мысленный эксперимент обычно обусловлен задачей, которая решается в ходе исследования;

6. Мысленное экспериментирование основано на разработке схемы мысленного действия, направленной на обработку исходной информации;

7. Мысленный эксперимент сочетает в себе силу формального вывода с экспериментальной достоверностью.

Как обобщается в (Sjoberg, 2008) [46], мысленные эксперименты, что касается их практического применения, создаются для того, чтобы:

1. Экстраполировать за пределы (или интерполировать внутри) границ уже установленного факта;

2. Предсказывать и прогнозировать неопределённое и непознаваемое будущее;

3. Объяснять прошлое;

4. Ретродикция, постдикция и ретроспектива неопределённого и непознаваемого прошлого;

5. Облегчение принятия решений, и выбора, в т. ч. выбора стратегии;

6. Решать проблемы и генерировать идеи;

7. Перемещение текущих (часто неразрешимых) проблем в другое, более полезное и более продуктивное проблемное пространство (например, см. функциональная фиксированность);

8. Приписывать причинно-следственную связь, предотвратимость, вину и ответственность за конкретные результаты;
9. Оценка виновности и компенсационного ущерба в социально-правовом контексте;
10. Обеспечить повторение прошлых успехов;
11. Изучить степень, в которой прошлые события могли произойти по-другому;
12. Обеспечить (в будущем) предотвращение прошлых неудач.

Таким образом, мысленный эксперимент – это форма мышления, объективно возникающая в результате активного воздействия человека на природу. Специфика этой формы заключается в том, что абстрактное и конкретное, рациональное, понятийное и чувственно-визуальное составляют её диалектическое единство. Мысленный эксперимент является эффективным средством получения новых знаний о мире.

3.3. Вопросно-ответный подход к разработке теории проекта

3.3.1. Вопросно-ответный анализ проектных задач

В пункте 3.1.3 мы уточнили роль объектов типа задач как в разработке проектной документации и теории проекта, так и в разработке проекта конкретной АС. Значение таких объектов позволяет интерпретировать проект в целом и с точки зрения конкретной задачи. В этой книге такую трактовку проекта мы будем называть «главной задачей Z^* проекта». Это упростит как наши рассуждения, так и используемые символичные обозначения.

Следует отметить, что с этого «момента» текста мы начинаем рассуждения, которые представят наш подход к использованию доступного опыта в разработке систем (систем программного обеспечения или АС). Напомним, что наши интересы сосредоточены только на концептуальном этапе проектирования.

Как было сказано выше, становление проектной документации $S(T_0, t_0)$ начинается с текста T_0 , содержание которого также определяет первый конструкт теории $Th^P(t_0)$. Поскольку мы глубоко убеждены, что задачи являются причинами разработки $Th^P(t)$, то можно предположить, что основная задача Z^* проекта инициирует запись текста T_0 , который будет использоваться в качестве начальной основы $Th^P(t)$. Кроме того, этот текст должен выражать начальное утверждение $St(Z^*, t_0)$ задачи Z^* . Другими словами, текст T_0 создаётся вне теории $Th^P(t_0)$, но этот текст должен использоваться в качестве исходной (информационной) основы $Th^P(t_0)$, с которой теория начинает своё становление путём пошаговой детализации (см. подраздел 3.1.3). Предположим, что текст T_0 создаётся

проектировщиком, который использовал шаги Empathize, Define и Ideate подхода дизайн-мышление.

Данное информационное основание должно удовлетворять следующим требованиям:

1. **Краткий текст.** Высказывание должно восприниматься как понятная целостность, когда на нём сосредоточено внимание. Поскольку «область» фокусировки ограничена человеческими возможностями, текст $S(Z^*, t_0)$ должен быть как можно короче.

2. **Как можно более абстрактно.** Утверждение $St(Z^*, t_0)$ является начальной версией динамической конструкции $St(Z^*, t)$, которая будет развиваться шаг за шагом (или состояние $S(Z^*, t_k)$ за состоянием $S(Z^*, t_{k+1})$) в течение жизненного цикла задачи Z^* . Любой шаг уменьшает определённый «объём» неопределённости $\nabla U(Z^*, t_k)$, существовавшей в $S(Z^*, t_k)$, путём включения в него следующего приращения $\Delta S(Z^*, t_k)$. Последовательность приращений и их содержание отражает последовательность действий по принятию решений и их результаты, каждое из которых может привести к ограничениям на будущие действия. Соответственно, в начальном состоянии $S(Z^*, t_0)$ утверждения неопределённость $\nabla U(Z^*, t_0)$ должна быть настолько неопределённой, насколько это возможно. Поэтому текст $S(Z^*, t_0)$ должен быть сформулирован как можно более абстрактно.

3. **Уровень абстракции в достаточной мере.** В соответствии с предыдущим пунктом текст $S(Z^*, t_0)$ с соответствующей ему неопределённостью $\nabla U(Z^*, t_0)$ является исходным источником для формирования последовательности приращений $\Delta S(Z^*, t_1), \Delta S(Z^*, t_2), \dots, \Delta S(Z^*, t_{k-1})$, которые развивают исходный текст до необходимого состояния $S(Z^*, t_k)$. Для такой генерации текст может быть интерпретирован как источник исходных аксиом, которые включены в этот текст. Этим аксиом должно быть достаточно для вывода состояния $S(Z^*, t_k)$.

В описанном варианте разработки $St(Z^*, t)$ указанные требования привели нас к шаблону начального состояния $St(Z^*, t_0)$, текст которого объединяет следующие три предложения:

1. Первое предложение отражает ориентацию его содержания на цель задачи Z^* . Это предложение включает в себя указатели не только на цель, но и на возможность её достижения. Сопровождаемая неопределённость может быть скрыта или зарегистрирована соответствующими знаками.

2. Второе предложение предлагает особенность идеи, которая может помочь решить эту задачу. Функция неявно указывает способ достижения цели.

3. Третье предложение касается среды (с её ограничениями), в которой команда проектировщиков будет работать с задачей Z^* .

Таким образом, жизненный цикл теории $Th^P(t)$ начинается с состояния $Th^P(t_0)$, которое состоит из трёх предложений текста $S(Z^*, t_0)$, каждое из которых имеет определённую функцию. Этот текст соответствует исходной постановке $S(Z^*, t_0)$ задачи Z^* , работа с которой будет направлена на проектирование соответствующей системы.

Существует способ расширить утверждение $S(Z^*, t_0)$ путём его анализа на основе доступного опыта. Суть этого способа образно показана на рис. 3.10.

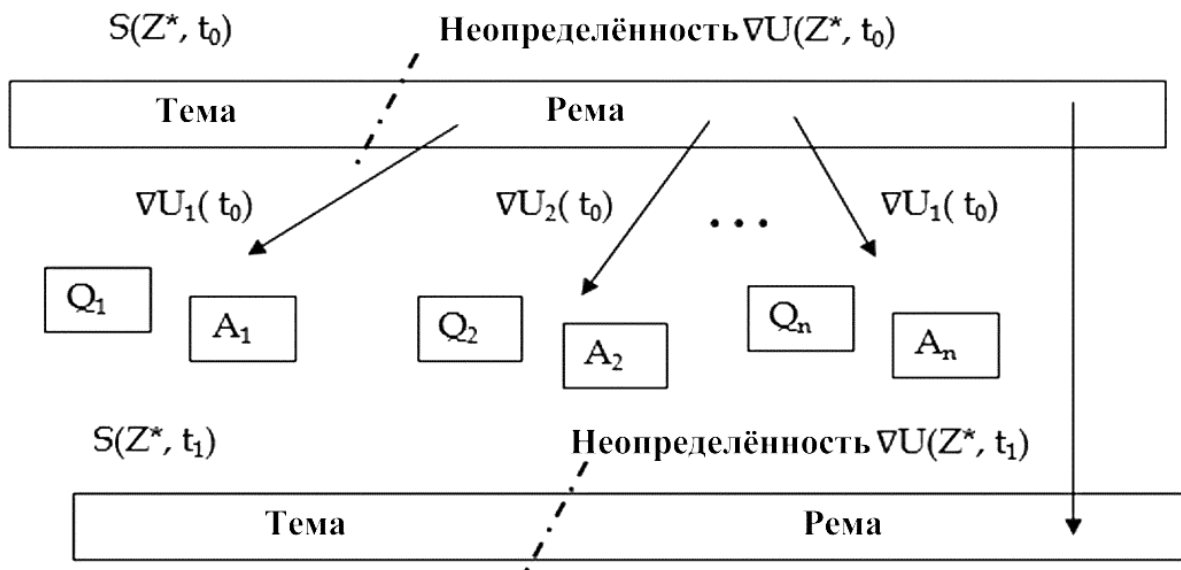


Рис. 3.10. Процесс анализа вопросов и ответов

Данный рисунок демонстрирует контролируемое уменьшение неопределённости $\nabla U(Z^*, t_0)$ утверждения $S(Z^*, t_0)$ с помощью использования вопросов-ответов при фактическом разделении его предложений. Во взаимодействиях с текстом фактическое разделение как коммуникативный механизм разбивает любое предложение на две части – тему и рему, где «тема» является предсказуемой отправной точкой интересного коммуникативного, а «рема» указывает новую информацию, которая должна получить дополнительную ценность в ответе на соответствующий вопрос. Таким образом, результат перехода из состояния $S(Z^*, t_0)$ в состояние $S(Z^*, t_1)$ может быть представлен конструкцией, показанной на рис. 3.11.

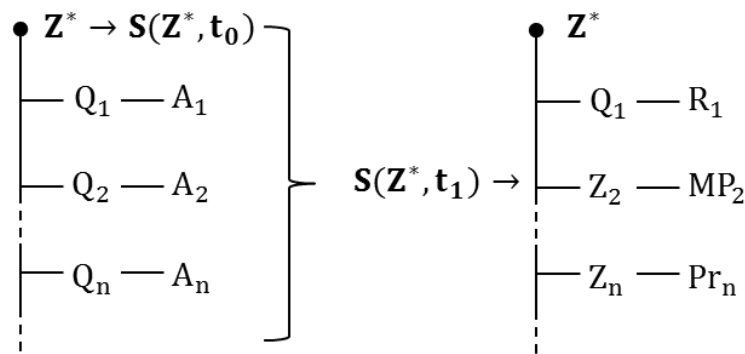


Рис. 3.11. Переход между отображениями

В коммуникативной практике конструкты Q_x и A_x могут иметь разные типы и цели, например:

1. Если ответ A_x на вопрос Q_x существует, и человек должен извлечь значение ответа из соответствующей онтологии базы данных или других информационных источников, вопрос Q_x можно квалифицировать как «Запрос» с символическим обозначением «Q».

2. Если ответ A_x отсутствует и может быть получен только после решения определённой задачи, тогда вопрос Q_x можно квалифицировать как «Задачу».

3. Если построение вопроса A_x привело к определённому требованию, то этот тип ответов можно назвать «Требованием».

Чтобы указать на разнообразие типов вопросов и ответов, на рисунке 3.11 обозначены узлы графа, который показывает результаты вопросно-ответного анализа $S(Z^*, t_0)$, где R_1 – требование, Z_2 и Z_n представляют задачи, MP – модель прецедента, а Pr – прототип.

В соответствии с пошаговой детализацией этот способ уменьшения неопределённости может быть продолжен для каждого узла графа, сгенерированного с использованием вопросно-ответного анализа. Применяя этот способ, проектировщики должны учитывать, что использование этого анализа похоже при любой задаче. Следовательно, шаг за шагом можно разработать графические структуры вопросно-ответного анализа, уменьшая неопределённость до соответствующей степени.

В наших исследованиях и на практике мы разработали вопросно-ответный подход (QA-подход) для проектирования АС [35, 36, 37, 38, 39, 40]. В этом подразделе мы рассматриваем отношения этого QA-подхода с нашей версией теоретизации проектов систем.

Ядром этого подхода является вопросно-ответный анализ (QA-анализ), некоторые особенности которого были представлены выше. При применении QA-подхода к задачам проекта результаты QA-анализа

регистрируются в графовых структурах, которые схематично показаны на рис. 3.12.

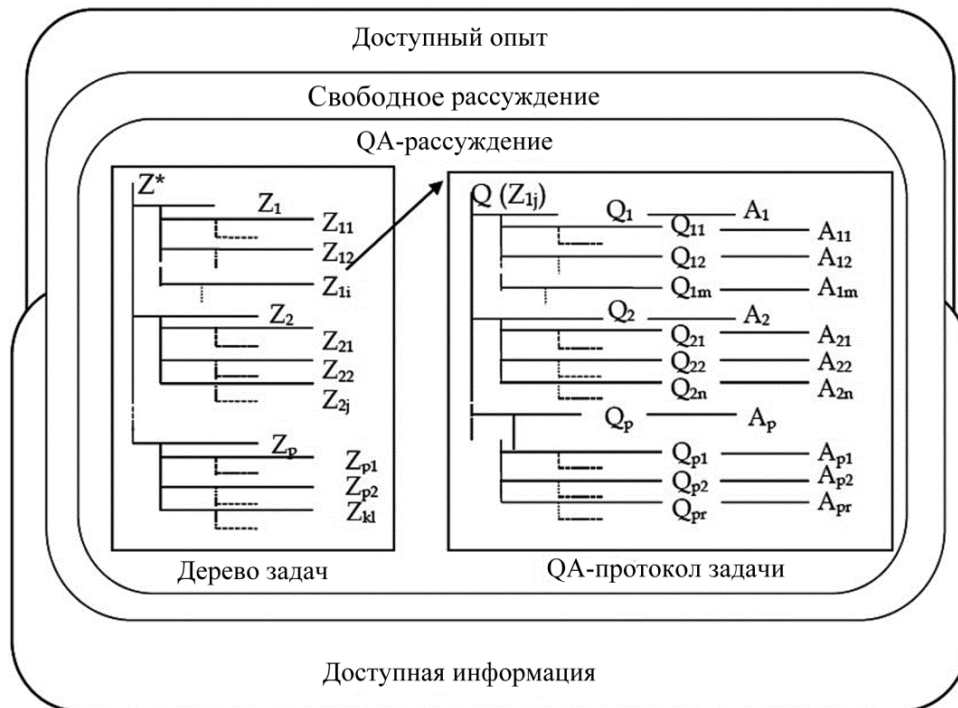


Рис. 3.12. Типичный результат вопросно-ответного анализа

Схема подчёркивает, что эти вопросно-ответные структуры (QA7-структуры) порождаются на основе естественных рассуждений (свободных рассуждений) проектировщиков, когда они конструктивно применяют вопросно-ответные рассуждения (QA-рассуждения) во время решения проектных задач. В этой работе проектировщики оперативно используют доступный опыт и соответствующую информацию.

Выясним суть указанных QA-структур. Для определённого проекта P они представляют его дерево задач $TT(Z^*, t)$ в текущий момент времени t . Другими словами, артефакт $TT(Z^*, t)$ отражает состояние проекта P в реальном времени с точки зрения структуры его задачи. Этот артефакт имеет следующие особенности:

1. Артефакт помещается в семантическую память инструментария WIQA (Working In Question Answer).

2. Для любой задачи из $TT(Z^*, t)$ отображение включает в себя вопросно-ответный протокол (QA-протокол, $QA(t)$), который кодирует QA-рассуждение, формируемое в ходе решения задачи.

3. Все узлы $TT(Z^*, t)$ визуальны проектировщикам.

4. Ячейки этой памяти помогают адекватно выразить семантику любой задачи проекта и любого компонента его QA-протокола.

5. Важное место среди атрибутов любой ячейки занимает поле, предназначенное для словесного описания загруженной единицы (описание для любого типа вопроса или ответа).

6. У проектировщиков есть возможность связать группу узлов в структуре $TT(Z^*, t)$ необходимым отношением.

Существует две основные точки зрения на артефакт $TT(Z^*, t)$:

– В соответствии с первым представлением, артефакт имеет иерархическую структуру, которая координируется с использованием пошаговой детализации.

– Вторая точка зрения отражает чистую структуру $TT(Z^*, t)$ в условиях, когда разработчики установили некоторые полезные отношения между узлами артефакта.

Таким образом, в пошаговом концептуальном проектировании дерево задач накапливает встроенные и используемые описания вопросов и ответов любых типов. Кроме того, проектировщики имеют возможность для проверки, исправления, обработки, изменения и использования любого из этих описаний, если это будет необходимо или полезно.

Одним из полезных применений этого информационного источника является разработка проектной документации, в которой типично размещать многократно используемые текстовые единицы в предопределённых положениях документов в их системе.

В разделе 3.1.3 мы описали систему $P^D(t)$ в её связи с теорией проекта $Th^P(t)$. Любые проектные документы и их системы должны быть построены как модели теории $Th^P(t)$. И, следовательно, артефакт $TT(Z^*, t)$ также может быть использован в качестве информационного источника для создания теории проекта. Более того, конструкты теории должны быть извлечены из $TT(Z^*, t)$ и после этого их необходимо включить в текущее состояние $Th^P(t)$. Только после этого проектировщики могут использовать такие конструкты в соответствующих документах. Эта последовательность соответствует следующей последовательности:

$$TT(Z^*, t) \rightarrow Th^P(t) \rightarrow P^D(t). \quad (3.8)$$

В этой последовательности первый переход предполагает, что словесные компоненты $TT(Z^*, t)$, которые будут включены в теорию $Th^P(t)$, должны быть тщательно обработаны, чтобы соответствовать требованиям теории. Любой из этих компонентов и его полезные комбинации с другими конструктами теории должны соответствовать текущему состоянию её фаз (описательная, классификационная и идентификационно-измерительная фазы).

Такая обработка неявно показана на рис. 3.13, схема которого раскрывает, что в условиях становления теории $Th^P(t)$ лучше использовать

онтологию (реализующую фазу классификации) и средства для проведения концептуальных экспериментов (поддерживающих работу проектировщиков на уровне идентификационно-измерительной фазы).

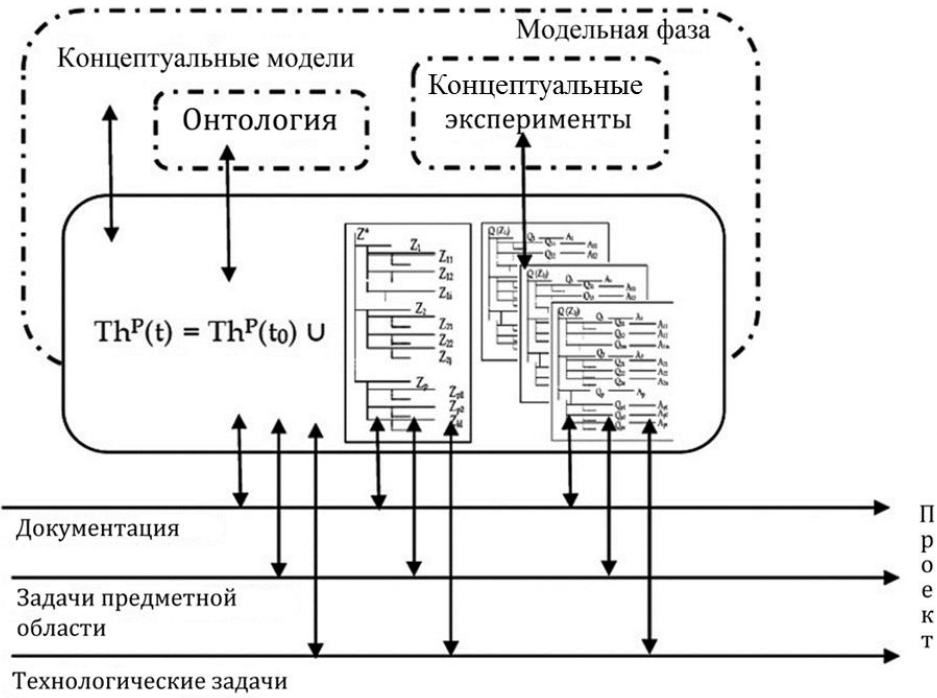


Рис. 3.13. Теория проекта в концептуальном проектировании

Схема также демонстрирует, что с помощью концептуальных моделей (модельная фаза) проектировщики могут использовать теорию для взаимодействия с документацией (задачами документации), а также с задачами предметной области и используемой технологии.

Один из положительных эффектов от использования теории $Th^P(t)$ был показан выше. Это контролируемое управление неопределённостью при поэтапном решении задач проекта. Этот тип управления положительно развивает потенциал управления проектами. Объединение теории с онтологией, которая развивается параллельно, и координация с теорией открывает возможность для использования контролируемой лексики в спецификациях, документации, коммуникации и рассуждениях проектировщиков. Теория также способна внести положительный вклад в повторное использование решаемых задач.

3.3.2. Координация дизайн-мышления и QA-анализа

Как отмечено в подразделе 3.1.3, работа с проектными задачами является источником разработки теории проекта $Th^P(t)$. При использовании вопросно-ответного подхода эта теория строится в процессе решения системы задач, начиная с корневой задачи Z^* , решение

которой регистрируется в артефакте $TT(Z^*, t)$. В этом случае одним из перспективных способов начать работу с корневой задачей является использование подхода, основанного на дизайн-мышлении (DT-подход).

Это утверждение основано на описании характера проектирования, представленного в (Dorst, 2010) [20], где К. Дорст предложил следующее начальное состояние применения QA-подхода:

$$??U \xrightarrow{???W} ?V \quad (3.9)$$

Данное выражение отражает ситуацию S , в которой проектировщик решил начать работу с новым проектом (новая корневая задача Z^*). Давайте проясним наше понимание этого выражения:

1. Существует цель G достичь определённого значения для потенциальных клиентов, и в текущий момент времени это значение V выражается с некоторой неопределённостью ($?V$).

2. Условия U , при которых цель может быть достигнута, воспринимаются неопределённо, и они могут быть выражены с существенной неопределённостью $??U$, для уменьшения которой условия могут быть построены и перестроены во время проектирования АС.

3. Конструкт $???W$ неизвестен, и его нельзя квалифицировать как проблему разрыва между U и V с учётом их неопределённостей.

4. Наличие цели G , выраженное конструктом ($?V$), открывает возможность для интерпретации ситуации S как индикатора задачи Z^* которая должна быть решена.

5. Для уменьшения неопределённости в выражении (3.9), необходимо применять дизайн-мышление, основанное на QA-рассуждении.

Эта позиция привела нас к объединению подхода «дизайн-мышление» с вопросно-ответным подходом, которое может быть выражено следующим выводом:

$$?^U U(Z^*) \xrightarrow{?^W W(Z^*)} ?^V V(Z^*), \quad (3.10)$$

где символы $?^U, ?^V$ и $?^W$ указывают на применение QA-подхода к компонентам U, V и W задачи Z^* . В описываемом случае этот подход реализуется после формулирования первоначальной постановки задачи Z_i и согласования с процессом дизайн-мышления.

Кроме того, использование QA-подхода для задачи Z^* будет порождать подчинённые задачи как узлы дерева $TT(t)$, и для любой новой задачи Z_i также применяется все сказанное выше для задачи Z^* .

Таким образом, на концептуальном этапе проектирования АС решение корневой задачи Z^* проекта будет продолжаться до момента

времени t_k , когда будут решены все подчинённые задачи. Таким образом, до этого момента постановка задачи Z^* и постановка любой подчинённой задачи могут быть изменены. Поэтому на самом деле проектировщики будут знать, какую задачу Z^* они концептуально решали, только в конце концептуального этапа. Таким образом, дерево задач $TT(t)$ является полезным артефактом, который адекватно отражает процесс концептуального проектирования АС.

3.3.3. Вопросно-ответная структура теории проекта

В предыдущем подразделе мы представили и раскрыли некоторые детали идеи теоретизации проекта на его концептуальной стадии. Эта идея имеет следующие основные особенности:

1. Теория проекта $Th^P(t)$ должна создаваться как теория содержательно эволюционного типа.

2. Становление теории должно начинаться с её первоначального основания в форме текстового описания, которое является как можно более кратким и абстрактным, но суть теории должна быть выражена в достаточной мере.

3. Основная цель разработки теории состоит в том, чтобы уменьшить неопределённость в концептуальном решении задач проекта путём использования пошаговой детализации совместно с QA-анализом при его применении к постановкам задач.

4. Проектировщики развивают теорию, когда они взаимодействуют с доступным опытом и используют соответствующую информацию в реальном времени концептуального проектирования, когда в дополнение к традиционным формам взаимодействия человека с компьютером они вынуждены использовать основанные на опыте формы HCI.

Как уже говорилось выше, в своём применении теория является источником ряда положительных эффектов, которые влияют на компоненты процесса проектирования и их результаты. Это достигается, когда описанный подход к разработке таких теорий становится более реальным. В исследовании, которым занимаются авторы, и практике этот подход был усовершенствован и получил название «Вопросно-ответный подход к концептуальному проектированию». Материализация этого подхода привела к разработке инструментария WIQA в нескольких версиях, которые поддерживают концептуальное проектирование систем.

При разработке этого инструментария его центральные задачи были направлены на усовершенствование типичных конструкторов теории. Поэтому набор базовых атрибутов может быть расширен проектировщиком с использованием механизма «Дополнительные атрибуты» и их композиций с учётом отображения на реальность.

Инструментарий был реализован таким образом, что класс типичных конструкторов включает в себя вопросы и ответы различных типов. В свою очередь, типы были разделены на два подкласса – базовые типы и типы, определённые проектировщиками. При назначении атрибутов для основных типов мы использовали сущность их референтов.

Мы понимаем такую сущность как «вопрос» как естественно-искусственный феномен, который возникает в голове у определённого человека, который явно или неявно пытается применить естественный опыт. Чтобы контролировать доступ к опыту, человек выражает это явление с помощью соответствующих знаков естественного языка. Таким образом, словесное выражение «вопроса» – это не более чем его знаковая модель.

Поэтому в спецификации такого конструктора, как вопрос Q, мы используем набор базовых атрибутов, который включает следующие элементы:

1. Символьное выражение «вопроса», зарегистрированное на языке проекта;
2. Создатель знакового выражения;
3. Время создания или последнего изменения выражения;
4. Идентификатор, который соответствует типу вопроса;
5. Идентификатор конструктора, определяющий его место в теории;
6. Адрес конструктора, определяющий её место в семантической памяти инструментария;
7. Атрибут, определяющий подчинённость между смежными вопросами;
8. Ссылка на необходимую графическую схему или изображение;
9. Ссылка на соответствующий материал в дотеоретической фазе.

Принимая во внимание взаимную взаимодополняемость вопроса и соответствующего ответа, ответ как конструктор теории имеет те же основные атрибуты. Однако любой ответ выражает реакцию на соответствующий вопрос, и эта реакция может коррелировать с любым типом ответов. Вот почему набор базовых атрибутов может быть расширен проектировщиком с использованием механизма «Дополнительные атрибуты». Например, если ответ соответствует определённому действию, то базовые атрибуты расширяются характеристиками этого действия. Во время регистрации любого вопроса или ответа или позже, когда это будет необходимо, проектировщик имеет возможность назначить этот конструктор соответствующего типа, который соответствует функции конструктора. Например, определённый вопрос Q можно квалифицировать как задачу Z после оценки возможности ответа. В этом случае вопрос Q будет иметь две точки его размещения в теории – первую в месте начальной регистрации, а вторую в соответствующей точке дерева задач.

Для связывания этих мест как Q , так и соответствующие Z помечены уникальным дополнительным атрибутом.

Важными компонентами любой теории являются суждения, типичными формами которых являются предложения (положения) на языке теории. Следует отметить, что теоретически $Th^P(t)$ знаковая модель любого вопроса или ответа имеет текстовую форму, включающую одно или несколько предложений. Например, вопрос может включать в себя предварительные условия, но даже вопрос состоит из одного предложения, его выражение включает часть возможного ответа.

Однако теоретически мы можем квалифицировать в качестве типичного предложения пару взаимно дополняющих друг друга вопросов и ответов. Такое суждение мы можем понимать как простую вопросно-ответную конструкцию (QA-конструкция) теории $Th^P(t)$. Сложные QA-конструкции состоят из необходимой композиции простых QA-конструкций.

Когда применяется теория $Th^P(t)$, её конструкт любого типа становится объектом, который используется в определённом приложении. Таким образом, в приложениях теории проектировщики могут использовать и обрабатывать Q-объекты, A-объекты и QA-объекты любого типа в различном составе и даже с объектами, выходящими за рамки теории $Th^P(t)$. В приложениях объекты из теории могут быть использованы в качестве моделей. Среди таких моделей важное место занимают QA-протоколы, отражающие концептуальные решения соответствующих задач. Поэтому такие типы моделей мы назвали «QA-модель задачи».

Ещё одним важным приложением теории $Th^P(t)$ является разработка соответствующей онтологии. Кроме того, это приложение в качестве классификационной фазы включено в $Th^P(t)$. Вот почему онтология создаётся на основе тех же типов конструктов, что и описательная фаза теории $Th^P(t)$. В онтологии любой концепт (термин, понятие) может рассматриваться как неявное (свёрточное) предложение или их группа.

В то же время онтология имеет полезные и важные приложения не только в теории, но и за её пределами. Например, онтология может применяться при извлечении ответов из изучаемых предложений. Отсутствие используемых слов в предложении в онтологии может быть истолковано как причины для вопросов о функциях или значениях этих слов в словесной единице дотеоретической фазы.

Онтология также полезна при экспериментальной проверке знаковых моделей вопросов и ответов. В этом случае, если проектировщик проводит мысленный эксперимент, часть которого реализуется с помощью моделей концептов вне мозга, то взаимодействие с онтологией может улучшить любой этап эксперимента, начиная со стадии его подготовки.

3.3.4. Концептуальное экспериментирование в концептуальном пространстве

Проектирование систем в условиях разработки и использования теории $Th^P(t)$ открывает возможности для целенаправленных мысленных экспериментов. Прежде всего, любой конструкт C_T теории должен быть проверен на его предварительную понятность. Затем создатель конструкта должен установить причины и следствия, вызванные включением конструкта в теорию. После этого включение конструкта должно быть выражено на уровне причинно-следственной связи. И, что является обязательным, включение конструкта в теорию должно быть проверено на соответствие принципу аддитивности.

Как сказано выше в разделе 3.3.3, мысленные эксперименты имеют более широкую область ответственности, которая может найти своё применение в различных вариантах в модельной фазе теории.

В соответствии с QA-подходом типичными конструктами теории являются вопросы и ответы, выраженные в текстовых формах. При проектировании системы соответствие любого вопроса Q и ответа A можно интерпретировать как следующее причинно-следственное отношение:

$$Q \rightarrow A, \quad (3.11)$$

которое должно удовлетворять логическому принципу достаточности. В простейшем случае это отношение должно сопровождать достаточное количество аргументов $\{Arg_i\}$. Таким образом, проверка отношения может быть достигнута путём использования адекватного мысленного эксперимента, после которого принятое отношение будет иметь вид:

$$Q_i \rightarrow (\{Arg_k\} \rightarrow A_i). \quad (3.12)$$

В свою очередь, есть некоторые умственные работы, которые должны быть выполнены для формулировки ответа A . Эти работы состоят из следующих пунктов:

1. Первоначальная формулировка ответа A ;
2. Управление лексикой текстового выражения A ;
3. Управление предикатами, используемыми в ответе;
4. Предварительное понимание текстового выражения ответа;
5. Управление лексикой A после понимания;
6. Выявление аргументов;
7. Контроль достаточности аргументации;
8. Контроль понятности ответа в его текущей формулировке.

Если сравнить эти работы со списком приложений мысленных экспериментов, которые были описаны в подразделе 3.3.3, то можно увидеть, что мысленный эксперимент может быть применён для любой из этих работ.

Следует отметить, что в соответствующей задаче ответ A может выполнять определённую роль, например, он может быть требованием для соответствующей задачи или определённым ограничением. При разработке системы ответ может выражать принцип, настройку, мотив, цель, фактор, условие, следствие, результат, аргумент или любую другую роль. Таким образом, для повышения качества аргументированного ответа A перечисленные работы применяются к любому аргументу импликации $(\{Arg_k\} \rightarrow A_i)$.

В соответствии с QA-подходом текстовое выражение любого ответа A является источником неопределённостей, которые должны быть уменьшены путём использования QA-анализа. Такое уменьшение неопределённости предполагает формирование набора вопросов $\{Q_{ij}\}$, которые в общем случае требуют выполнения следующих работ:

1. Обнаружение вопроса;
2. Определение вопроса;
3. Формулировка вопроса.

Цели обнаружения и определения вопроса могут быть достигнуты, когда проектировщик выполняет формулировку ответа A_i , прежде всего во время следующих работ:

1. Управление лексикой текстового выражения A ;
2. Контроль предикатов, используемых в ответе;
3. Предварительное понимание текстового выражения ответа;
4. Управление лексикой A после понимания;
5. Выявление аргументов;
6. Контроль понятности ответа в его текущей формулировке.

Формулировка любого вопроса аналогична формулировке ответа. Таким образом, все работы, которые обеспечивают формулировку ответа, также применяются к формулировке вопроса, и все виды мысленных экспериментов, которые поддерживают формулировку ответа, могут быть полезны при формулировке вопроса.

Таким образом, мысленные эксперименты имеют многочисленные и разнообразные применения в становлении теории $Th(t)$ и её приложений. В любом случае при таком экспериментировании любой проектировщик проводит любой мысленный эксперимент в концептуальной среде, которая схематично показана на рис. 3.14.

Эта схема демонстрирует, что определённые компоненты эксперимента находятся за пределами мозга. Внешняя концептуальная среда включает в себя следующие компоненты:



Рис. 3.14. Мысленный эксперимент в концептуальной среде

1. Экспериментальная установка, функции которой выполняет многократно используемый комплекс действий, распределённых между проектировщиком и компьютером.

2. Объект эксперимента, который может быть конструктом теории или определённой моделью приложения теории.

3. Условия эксперимента, указывающие на концептуальное окружение проведения эксперимента.

4. Концептуальная среда, в которой внешние компоненты, вовлечённые в мысленный эксперимент, существуют в компьютеризированных формах.

5. Проектировщик, который является активной силой эксперимента. Он (проектировщик) взаимодействует с концептуальной средой и её компонентами с помощью средств, которые помогают психическим процессам, используемым в концептуальном эксперименте. Более того, проектировщик участвует во внешних действиях и управляет их эффективным исполнением.

Таким образом, в вопросно-ответном подходе процесс мысленного экспериментирования частично автоматизирован, поэтому мы назвали такую версию мысленного экспериментирования «концептуальным экспериментированием» (Соснин, 2015) [47].

Главной особенностью концептуальных экспериментов является необходимость взаимодействия проектировщика с доступным опытом в режиме реального времени. Эти взаимодействия основаны на естественных вопросно-ответных рассуждениях, которые переплетаются с

вопросами-ответами за пределами структур мозга. Описанную специфику взаимодействия человека с компьютеризированной средой можно квалифицировать как человеко-компьютерное взаимодействие на основе опыта.

Кроме того, специфическими особенностями концептуального экспериментирования являются использование отображений сущностей, участвующих в любом концептуальном эксперименте, в семантической памяти типа вопрос-ответ, и псевдокодированное программирование действий проектировщика, который выполняет их как интеллектуальный процессор (Соснин, 2013) [9].

3.3.5. Теоретически экспериментальный взгляд на проект

Выше мы попытались обосновать целесообразность использования теоретизации проектов на основе вопросно-ответного подхода. Наши рассуждения привели нас к становлению теории $Th^P(t)$, фазовая структура которой представлена на рис. 3.15.

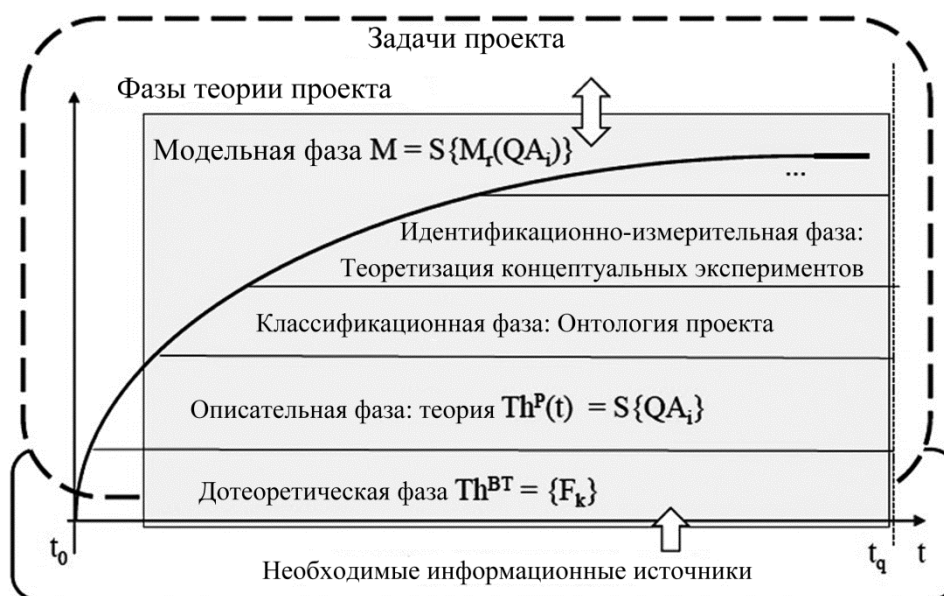


Рис. 3.15. Фазовая структура теории проекта

Схема указывает, что фазовая структура объединяет артефакт $Th^P(t)$ с соответствующей онтологией и теоретизацией ряда концептуальных экспериментов, которые были проведены для обоснования правильности конструкторов теории $Th^P(t)$.

Конструкторы теории визуально наблюдаемы в их системе $S\{QA(t), t\}$, которая включает в себя дерево задач и QA-протоколы для этих задач. Визуально эта система представляет собой совокупность сетей, типичный узел которых показан на рис. 3.16.

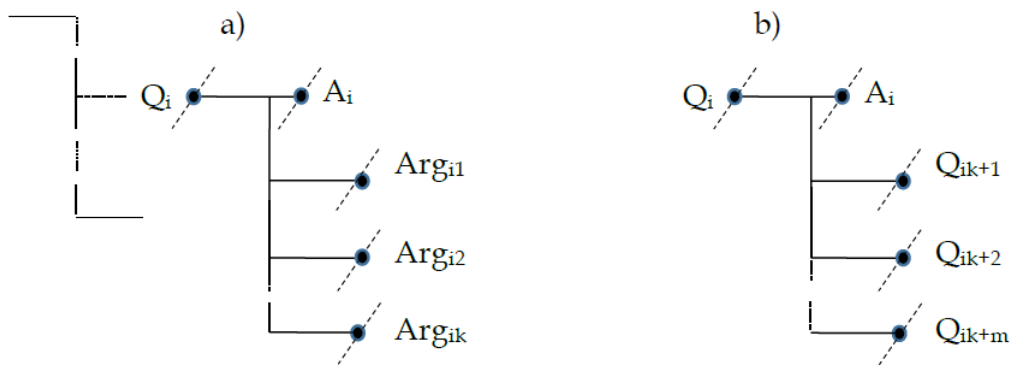


Рис. 3.16. Типичная структура узлов

Согласно четвёртому закону классической логики, любой ответ, входящий в структуру теории, должен быть достаточно обоснованным. На рисунке 3.16 левая структура узлов отражает случай, когда обоснование ответа A_i даёт использование списка аргументов $\{Arg_{ik}\}$. Правильная структура соответствует случаю, когда, помимо аргументации, ответ был источником более детального опроса.

Как сказано выше, ответ может иметь вид следующего открытого списка {принцип, утверждение, настройка, требование, ограничение, мотив, цель, фактор, условие, следствие, результат, аргумент, предупреждение, совет, комментарий...}. Кроме того, любой узел в этой системе имеет не только иерархическую связь со своим контекстом, но и другие типы системных связей, обусловленные, например, используемой онтологией. Следовательно, артефакты $Th^P(t)$ визуальны доступны в виде семантической сети, которая явно и неявно (через посредников) связана с теоретическими моделями, применяемыми при решении задач проекта.

Теория $Th^P(t)$ в сочетании с соответствующей онтологией, концептуальными экспериментами и моделями открывает новую возможность для улучшения концептуального проектирования. Эти возможности обусловлены использованием основанных на опыте человеко-компьютерных взаимодействий, основной особенностью которых является ответ на вопрос о структурах и условиях, описанных выше.

Как было отмечено, вопросы представляют собой естественно искусственные явления, которые возникают в определённых условиях и вызывают определённые реакции. Поэтому любой вопрос должен быть обнаружен, идентифицирован, зарегистрирован, описан, уточнён, проверен и применён. Задать вопрос – значит построить адекватный ответ. Эта работа может быть реализована проектировщиком в одном из трёх миров: искусственный, искусственно естественный мир или мир природы. Эта сторона вопросов схематично показана на рис. 3.17.



Рис. 3.17. Типы вопросов

В процессе проектирования определённой системы задачи, стоящие за вопросами, помогают создать теорию $Th^P(t)$ с помощью соответствующих ответов, каждый из которых должен быть построен. Среди этих ответов мы выделяем те, которые имеют поведенческий характер или другими словами ответы в форме поведенческих действий. Такие ответы имеют два концептуальных варианта их регистрации в проекте. Первый вариант представляет собой описание конструкта теории $Th^P(t)$, в то время как второй вариант является руководством для построения ответа. Второй вариант можно интерпретировать как некое правило (в теории $Th^P(t)$), которое указывает на вывод ответа на соответствующий вопрос. Вторая интерпретация руководства для генерации ответа – это его модель (модель соответствующего конструкта $Th^P(t)$), которую проектировщики могут многократно использовать при проектировании. В концептуальных экспериментах такие руководства отвечают требованиям, установленным для экспериментов.

Аналогичные рассуждения мы можем провести для ответа любого типа, который соотносится с соответствующим вопросом определённого типа. Это рассуждение приводит к дополнительности <СУЩНОСТЬ - её ЗНАЧЕНИЕ>, что соответствует неявному вопросу о значении сущности. На практике, например, для физических переменных, явный переход к значению соответствует операционным определениям соответствующих переменных. В программировании дополнительность раскрывается через «программную переменную и её значение».

Следует отметить, что при проектировании системы необходимость ответов на вопросы в концептуальных формах возникает в разных случаях. Вот почему концептуальные эксперименты могут быть полезны не только для создания теории $Th^P(t)$. Концептуальные эксперименты особенно важны для разработки и тестирования поведенческих действий

проектировщиков, например, когда они решают задачи проекта с учётом их повторного использования.

3.3.6. Взаимодействие на основе опыта

Чтобы усовершенствовать QA-подход, мы разработали специализированную инструментально-технологическую среду WIQA (Working In Questions and Answers – Работа в вопросах и ответах), использование которой поддерживается рабочими процессами «Взаимодействие с опытом». При концептуальном проектировании систем любой проектировщик применяет эти рабочие процессы в сочетании с другими технологическими процессами в условиях, которые схематично показаны на рис. 3.18.

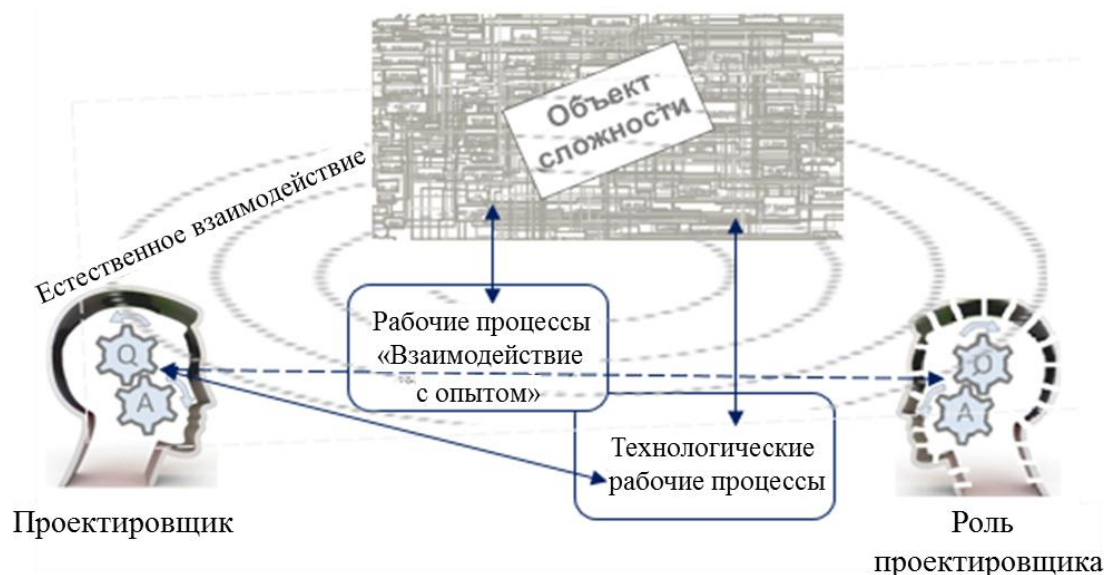


Рис. 3.18. Условия взаимодействия

Эта схема образно раскрывает нашу идею автоматизации естественного взаимодействия в снижении сложности, о которой мы рассказали в конце раздела 2 данной главы. В нём была подчёркнута необходимость поиска такого способа автоматизации естественных взаимодействий с объектом сложности и запрограммированных интерфейсов, который может быть использован в непредсказуемых ситуациях.

Смысл нашей идеи автоматизации включает в себя следующее:

1. В любой момент неявного взаимодействия с объектом сложности и неявного или явного восприятия состояния запрограммированных интерфейсов проектировщик может нарушить технологический процесс, если разум проектировщика сигнализирует о появлении вопроса.

2. Предположим, что обнаруженный вопрос Q соответствует теории и проектировщик мысленно решает отреагировать на него.

3. Это решение станет поводом для активации рабочих процессов «взаимодействия с опытом», которые имеют собственные запрограммированные интерфейсы, разработанные для автоматизации естественных взаимодействий с естественным опытом.

4. В QA-подходе, в современном состоянии его реализации, эти интерфейсы провоцируют управляемые воздействия на диалогическое структурирование вербальных отображений в сознании, мысленном воображении и мысленном эксперименте, задуманном проектировщиком.

5. Указанные влияния мы интерпретируем как управляемые искусственные взаимодействия проектировщика с тем же проектировщиком, который выполняет специализированную роль (интеллектуальный процессор, QA-процессор).

6. Этот вид искусственных взаимодействий переплетается с соответствующими естественными взаимодействиями, и такой интегральный процесс мы определяем как «основанные на опыте взаимодействия человека и компьютера», которые, в свою очередь, могут сочетаться с HCI других видов.

Контрольные вопросы

1. Что такое концептуальное экспериментирование?
2. Какие положительные эффекты в проектировании систем даёт подход SEMAT?
3. Какие фазы включает в себя теория в моделировании научной деятельности?
4. Какие типы экспериментов выделяют учёные?
5. В каких направлениях раскрывается сущность «разработка концепций и экспериментирование»?
6. Для чего создаётся мысленный эксперимент?
7. Какие этапы включает в себя жизненный цикл эксперимента?
8. Что такое дизайн-мышление?
9. Какими свойствами обладает биполярная ориентация дизайн-мышления?
10. Значение каких объектов позволяет интерпретировать проект в целом и с точки зрения конкретной задачи.
11. Каково назначение QA протокола в вопросно-ответном анализе?
12. Какие базовый атрибуты содержит конструкт «вопрос»?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основная литература:

1. Козонов, Д. В. Основы визуального моделирования / Д. В. Кознов. – М. : Интернет Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. – 248 с.
2. Леффингуэлл, Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход / Д. Леффингуэлл. – М. : «Вильямс», 2002. – 448 с.
3. Липаев, В. В. Программная инженерия. Методологические основы : учебник / В. В. Липаев ; Гос. ун-т – Высшая школа экономики. – М. : ТЕИС, 2006. – 608 с.
4. Рамбо, Д. UML : спец. справочник / Рамбо Дж., Якобсон А., Буч Г. – Санкт-Петербург : Питер, 2002. – 656 с.
5. Розенберг, В. Применение объектного моделирования с использованием UML и анализ прецедентов / В. Розенберг, К. Скотт. – М. : Изд-во ДМК, 2002. – 160 с.
6. Соммервилл, И. Инженерия программного обеспечения / И. Соммервилл. – М. : Вильямс, 2002. – 623 с.
7. Соснин, П. И. Вопросно-ответное моделирование в разработке автоматизированных систем / П. И. Соснин. – Ульяновск : УлГТУ, 2007. – 333 с.
8. Соснин, П. И. Архитектурное моделирование автоматизированных систем : учебное пособие / П. И. Соснин. – Ульяновск : УлГТУ, 2007. – 146 с.
9. Соснин, П. И. Вопросно-ответное программирование человеко-компьютерной деятельности / П.И. Соснин. – Ульяновск : УлГТУ. 2010. – 240 с.
10. Соснин П.И. Человеко-компьютерная диалогика / П. И. Соснин. – Ульяновск : УлГТУ, 2001. – 285 с.
11. Фаулер, Орлов, Якобсон UML. Классика CS. – 2-е изд. – СПб. : Питер, 2005. – 736 с.
12. Якобсон, А. Унифицированный процесс разработки программного обеспечения / А. Якобсон; пер. с англ. – СПб. : Питер, 2002. (Для профессионалов). – 492 с.

Дополнительная литература:

13. Basili, V. R., Lindvall, M., Costa, P. (2001) Implementing the experience factory concepts as a set of experience bases. In Proceedings of the

2001 International Conference on Software Engineering & Knowledge Engineering, pp. 102-109.

14. Bass, L., Ivers, J., Klein, M. & Merson, P. (2005). Reasoning Frameworks, Software Engineering Institute, *Tech. Rep. Carnegie Mellon University*, Pittsburgh, PA, CMU/SEI-2005-TR-007.

15. Booch, G. & Brown, A. W. (2003). Collaborative development environments. In M. Zelkowitz (Ed.), *Advances in computers*, 59, San Diego, CA: Academic Press.

16. Bullinger, H.-J., Warschat, J., Schumacher, O., Slama, A., Ohlhausen, P.: Ontology-Based Project Management for Acceleration of Innovation Project, Lecture Notes in Computer Science, Vol. 3379, pp. 280-288 (2005).

17. Cares, C., Franch, X., Mayol, E. (2006) Perspectives about paradigms in software engineering, in Proc. 2nd International workshop on Philosophical Foundations on Information Systems Engineering, pp. 737-744.

18. Chandrasekaran, B., Josephson, J. R., Benjamins, V. R (1998) Ontology of tasks and methods, In Proceedings of the Workshop on Applications of Ontologies and Problem-Solving Methods, held in conjunction with ECAI'98, Brighton, UK, pp. 31-43 (1998).

19. Charette, R.N. (2005). Why software falls. *IEEE Spectrum*, 4(9), pp. 36-43.

20. Dorst K. The Nature of Design Thinking, in DTRS8 Interpreting Design Thinking, Design Thinking Research Symposium Proceedings, 2010, pp. 131-139.

21. Eden, A. H., Turner, R. (2007) Problems in the Ontology of Computer Programs, *Applied Ontology*, Vol. 2, No. 1, Amsterdam, IOS Press, pp. 13-36.

22. Exman, I., Perry, D.E., Barn, B., Ralph, P. (2016). Separability principles for a general theory of software engineering: Report on the GTSE 2015 workshop. *SIGSOFT Software Engineering Notes* (41:1), p. 25-27.

23. Guarino, N. (2009) The Ontological Level: Revisiting 30 years of knowledge representation. conceptual modelling: foundations and applications, *Essays in Honor of John Mylopoulos*, Springer-Verlag, pp. 52-67.

24. Guarino, N., Oberle, D., Staab, S. (2009). What is an Ontology? In S. Staab and R. Studer (Ed.), *Handbook on Ontologies*, Second Edition. International handbooks on information systems. Springer Verlag, pp. 1-17.

25. Henninger, S. (2003) Tool Support for Experience-based Software Development Methodologies, *Advances in Computers*, vol. 59, 2003, 29-82.

26. Hewett, T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., & Verplank, W. (2002). ACM SIGCHI Curricula for Human Computer Interaction. ACM Technical Report. P. 162.

27. Jacobson, I., Meyer, B., Soley R. (2009). The SEMAT Initiative: A Call for Action. *Dr. Dobb's Journal*, 2009.

28. Jeffery, D.R., & Scott, L. (2002) Has twenty-five Years of Empirical Software Engineering Made a Difference, In Proceedings of the 2nd Asia-Pacific Software Engineering Conference, pp.539-549.
29. Johnson P., Ekstedt M. and I. Jacobson, Where's the Theory for Software Engineering?, IEEE Software, vol. 29, no. 5, pp. 94–96, 2012.
30. Johnson, P., Ralph, P., Goedicke, M., Ng, P.-W., Stol, K.-J., Smolander, K., Exman, I., Perry, D. 2013. Report on the second SEMAT workshop on a general theory of software engineering (GTSE 2013). SIGSOFT Software Engineering Notes (38:5), p. 47-50.
31. Karray, F., Alemzadeh, M., Saleh, J. A. & Arab, M. N. (2008) Human-Computer Interaction: Overview on State of the Art Smart sensing and intelligent systems. vol. 1, no.1, pp. 138-159
32. Pew, R. W. (2007). Some history of human performance models. In W. Gray (Ed.), Integrated models of cognitive systems. New York: Cambridge University Press, pp. 29–47.
33. Perry, D. (2016). Theories, theories everywhere. USA, IEEE/ACM 5th International Workshop on Theory-Oriented Software Engineering (TOSE).
34. Ralph, P. (2011) Toward a theory of debiasing software development. In Proceedings of the SIGSAND/PLAIS Euro-Symposium, Springer LNBIP 93, pp 92–105.
35. Ralph, P. (2012) Sensemaking-Coevolution-Implementation Theory: a model of the software engineering process in practice. The 2012 SEMAT Workshop on a General Theory of Software Engineering, Stockholm, Sweden, November.
36. Ralph, P. (2013). Possible core theories for software engineering. The SEMAT Workshop on a General Theory of Software Engineering, International Conference on Software Engineering, San Francisco, CA, USA, May.
37. Ralph, P. (2014). Evaluating process theories in software engineering. The 3rd SEMAT Workshop on General Theories of Software Engineering, International Conference on Software Engineering, Hyderabad, India, June: ACM.
38. Ralph, P. (2015) Developing and evaluating software engineering process theories. In Proceedings of the International Conference on Software Engineering. Florence, Italy: IEEE, May.
39. Ralph, P. (2015) The Sensemaking-Coevolution-Implementation theory of software design. Science of Computer Programming (101), p. 21-41.
40. Ralph, P. (2016) Software engineering process theory: A multi-method comparison of Sensemaking-Coevolution-Implementation Theory and Function-Behavior-Structure Theory, Information and Software Technology (70), p. 232-250.

41. Ralph, P., Chiasson, M., & Kelley, H. (2016) Social theory for software engineering research. Proceedings of the International Conference on Evaluation and Assessment in Software Engineering. Limerick, Ireland: ACM, June.
42. Ralph, P., Exman, I., Ng, P.-W., Johnson, P., Goedicke, M., Kocatas, A. T., & Yan, K. L. (2014). How to develop a general theory of software engineering: Report on the GTSE 2014 workshop. SIGSOFT Software Engineering Notes (39:6), p. 23-25.
43. Ralph, P., Johnson, P., & Jordan, H. (2013) Report on the first SEMAT workshop on a general theory of software engineering (GTSE 2012). SIGSOFT Software Engineering Notes (38:3), p. 26-28.
44. Sedano, T., Ralph, P. & Péraire, C. (2017) Lessons learned from an extended participant observation grounded theory study. Proceedings of the 5th International Workshop on Conducting Empirical Studies in Industry. Buenos Aires, Argentina: IEEE, May.
45. Simperl, E., Michel, M & Burger, T. (2010) Achieving maturity: the state of practice in ontology engineering, International Journal of Computer Science and Applications & Technomathematics Research Foundation, Vol. 7 No. 1, pp. 45 - 65, (2010).
46. Sjøberg, D.I.K., Dybå, T. Anda, B.C.D. and Hannay, J.E. Building Theories in Software Engineering. To appear in F. Shull, J. Singer, D.I.K. Sjøberg (eds.), Advanced Topics in Empirical Software Engineering, Springer-Verlag, 2007
47. Sosnin P, Conceptual Experiments in Automated Designing. Chapter in the book «Projective Processes and Neuroscience in Art and Design», Eds.: Rachel Zuanon, IGI-Global, 2016, pp. 155-181.
48. Sosnin P., Substantially Evolutionary Theorizing in Designing Software-Intensive Systems, Information 2018, 9(4), 91, 29 pages.
49. Sosnin P. (2017). Question-Answer Analysis in Design Thinking at the Conceptual Stage of Developing a System with a Software, In Proceedings of the 8th International Conference on Information, Intelligence, Systems and Applications.
50. Sosnin P. Experience-Based Human-Computer Interactions: Emerging Research and Opportunities, IGI-Global, 2018, 294 pages.
51. Sosnin P. and Galochkin M. (2017) Way of Coordination of Visual Modeling and Mental Imagery in Conceptual Solution of Project Task, Advances in Artificial Intelligence: From Theory to Practice, LNCS, volume 10350, 2017, pp. 635-638.
52. Sosnin P and Pushkareva A. Ontological Controlling the Lexical Items in Conceptual Solution of Project Tasks LNCS, volume 10409, 2017, pp. 31-46.
53. Southekal, P.H. & Levin, G. (2011) Formulation and Empirical Validation of a GQM Based Measurement Framework, In Proceedings of 11th

International Symposium on Empirical Software Engineering and Measurement, 2011, pp. 404-413

54. Standish Group, Chaos Reports (1994-2016)
<http://www.standishgroup.com>

55. Stol, K., Ralph, P. & Fitzgerald, B. (2016) Grounded theory research in software engineering: A critical review and guidelines Proceedings of the 2016 International Conference on Software Engineering. Austin, Texas: ACM, May.

56. Xu, S. & Rajlich, V. (2005) Dialog-Based Protocol: An Empirical Research Method for Cognitive Activity in Software Engineering. In Proceedings of the 2005 ACM/IEEE International Symposium on Empirical Software Engineering, pp. 397-406.

57. Webber, B. & Webb, N. (2010) Question Answering. In Clark, Fox and Lappin (eds.): Handbook of Computational Linguistics and Natural Language Processing. Blackwells.

Электронные ресурсы:

58. Borges, P., Machado, R.J., Riberio, P. (2012) Mapping RUP Roles to small software development teams. Proceedings of International conference on software and system process, p. 190-199 : [Электронный ресурс]. – Режим доступа:

http://www3.dsi.uminho.pt/rmac/privatefiles/papers/2012_SWQD_BorgesMonteiroMachado-springer.pdf (дата обращения: 20.04.2020)

59. Bundeswehr [сайт]

<https://www.bundeswehr.de/de/organisation/ausruestung-baainbw/ruestungsprojekte/cd-e>

60. Businessdictionary : [Электронный ресурс]. – Режим доступа: <http://www.businessdictionary.com/definition/experiment.html> (дата обращения: 20.04.2020)

61. CD&E Method Description // The Swedish armed forces : [Электронный ресурс]. – Режим доступа: http://blogg.forsvarsmakten.se/utvecklingsbloggen/files/2012/06/CDE_Method_Description_I-1_0_INT.pdf (дата обращения: 20.04.2020)

62. Cilb, T. A Conceptual Glossary for Systems Engineering: Define the Concept, don't quibble about the terms : [Электронный ресурс]. – Режим доступа: http://semat.org/documents/20181/27952/conceptual_glossary.pdf/741a889f-332c-48fe-9af0-62334eeaf767 (дата обращения: 20.04.2020)

63. Courage, C. Reweaving corporate DNA: building a culture of design thinking an CITRIX : [Электронный ресурс]. – Режим доступа: <https://www.managementexchange.com/story/reweaving-corporate-dna-building-culture-design-thinking-citrix> (дата обращения: 20.04.2020)

64. Defence Research and Development Canada : [Электронный ресурс]. – Режим доступа: <http://www.drdc-rddc.gc.ca/en/index.page> (дата обращения: 20.04.2020)
65. Dictionary.com: онлайн словари и переводчики : [Электронный ресурс]. – Режим доступа: <http://www.dictionary.com/browse/experiment> (дата обращения: 20.04.2020)
66. El Emam, K., Koru, A. G. A Replicated Survey of IT Software Project Failures : [Электронный ресурс]. – Режим доступа: https://ruor.uottawa.ca/bitstream/10393/12988/1/El_Emam_Khaled_2008_A_replicated_survey_of_IT_software.pdf
67. Flevy Marketplace. Design thinking : [Электронный ресурс]. – Режим доступа: <http://flevy.com/browse/business-document/design-thinking-1980> (дата обращения: 20.04.2020)
68. Glass, R. The Standish Report: Does It Really Describe a Software Crisis? : [Электронный ресурс]. – Режим доступа: <http://www.umsl.edu/~sauterv/analysis/Standish/glass.pdf>
69. Gonzalez-Perez, C., Henderson-Sellers, B.: Modelling software development methodologies: a conceptual foundation : [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/222042895_Modelling_software_development_methodologies_A_conceptual_foundation (дата обращения: 20.04.2020)
70. Haze. – The Standish Group. (2015) : [Электронный ресурс]. – Режим доступа: https://www.standishgroup.com/sample_research_files/Haze4.pdf (дата обращения: 20.04.2020)
71. IABG Group. Concept Development & Experimentation : [Электронный ресурс]. – Режим доступа: <http://www.iabg.de/en/business-fields/defence-security/services-solutions/capability-management/concept-development-experimentation/> (дата обращения: 20.04.2020)
72. Interaction design foundation. Design thinking : [Электронный ресурс]. – Режим доступа: <https://www.interaction-design.org/literature/topics/design-thinking> (дата обращения: 20.04.2020)
73. Jacobson, I., Ng P. W., McMahon, P. E., Spence, I. Svante Lidman S. The Essence of Software Engineering: The SEMAT Kernel <https://queue.acm.org/detail.cfm?id=2389616>
74. Jorgensen, M., Molokken-Ostvold K. How large are software cost overruns? critical comments on the standish group's chaos reports. <http://www.umsl.edu/~sauterv/analysis/Standish/standish-IST.pdf>
75. Kelley, D. Design thinking : [Электронный ресурс]. – Режим доступа: <http://www.ideo.com/pages/design-thinking> (дата обращения: 20.04.2020)
76. Lengler, R., Eppler, M. J. Towards A Periodic Table of Visualization Methods for Management : [Электронный ресурс]. – Режим доступа:

https://www.visual-literacy.org/periodic_table/periodic_table.pdf (дата обращения: 20.04.2020)

77. Lexico : лексический словарь : [Электронный ресурс]. – Режим доступа: <https://www.lexico.com/definition/experiment> (дата обращения: 20.04.2020)

78. Li, M., Vitanui, P.M.B. (2008). An introduction to Kolmogorov complexity and its application series : text in computer science (3rd ed.). Springer

79. NATO concept development and experimentation (CD&E) process : [Электронный ресурс]. – Режим доступа: http://www.act.nato.int/images/stories/events/2011/cde/tr_mcm0056.pdf (дата обращения: 20.04.2020)

80. Rouse, M. Design thinking : [Электронный ресурс]. – Режим доступа: <http://whatis.techtarget.com/definition/design-thinking> (дата обращения: 20.04.2020)

81. SEMAT [сайт]. : [Электронный ресурс]. – Режим доступа: <http://semat.org/blog/-/blogs/some-critiques-of-the-semat-initiative>

82. Swedish Defence Research Agency : [Электронный ресурс]. – Режим доступа: <https://www.foi.se/en/foi.html> (дата обращения: 20.04.2020)

83. TheFreeDictionary.com: онлайн словари и переводчики : [Электронный ресурс]. – Режим доступа:

84. TNO organization. Annual report : [Электронный ресурс]. – Режим доступа: <https://www.tno.nl/en/about-tno/more-about-our-work/annual-reports/> (дата обращения: 20.04.2020)

85. Webster's Online Dictionary : [Электронный ресурс]. – Режим доступа: <https://www.merriam-webster.com/dictionary/experiment> (дата обращения: 20.04.2020)

86. Wiel, W. M., Hasberg, M. P., Weima, I., Huiskamp, W. Concept Maturity Levels Bringing structure to the CD&E process : [Электронный ресурс]. – Режим доступа: <https://repository.tudelft.nl/view/tno/uuid:b8d2ec1f-f2aa-47a3-b7b3-c37162bb9f34> (дата обращения: 20.04.2020)

87. Wikipedia. Concept development and experimentation : [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Concept_development_and_experimentation (дата обращения: 20.04.2020)

Стандарты и методологии:

88. CMMI Institute [сайт]: [Электронный ресурс]. – Режим доступа: <https://cmmiinstitute.com> (дата обращения: 20.04.2020)

89. IDEF [сайт]: [Электронный ресурс]. – Режим доступа: <http://idef.ru> (дата обращения: 20.04.2020)

90. IEEE Std 1471-2000. 1471-2000 - IEEE Recommended Practice for Architectural Description for Software-Intensive Systems. Institute of Electrical

and Electronics engineers, Sept. 200 <http://cabibbo.dia.uniroma3.it/ids/altrui/ieee1471.pdf>

91. ISO/IEC/IEEE. 2011. Systems and software engineering - Architecture description. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 42010. http://sewiki.ru/ISO/IEC/IEEE_42010

92. UML [сайт]: [Электронный ресурс]. – Режим доступа: <https://www.uml.org> (дата обращения: 20.04.2020)

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АС	– Автоматизированная Система
ИИ	– Искусственный Интеллект
CD&E	– Concept development and experimentation
СММІ	– Capability Maturity Model Integration
GTSE	– General Theory of Software Engineering
HCI	– Human Computer Interaction
IDEF	– Integrated DEFinition
IEEE	– Institute of Electrical and Electronics Engineers
ISO	– International Organization for Standardization
QA	– Question-Answer
RUP	– Rational Unified Process
SEMAT	– Software Engineering Method and Theory
SIS	– Software Intensive System
UML	– Unified Modeling Language
WIQA	– Working In Questions and Answers

Учебное издание

СОСНИН Пётр Иванович
ВАЛЮХ Вероника Валерьевна

**ЧЕЛОВЕКО-КОМПЬЮТЕРНОЕ
ВЗАИМОДЕЙСТВИЕ**

Учебное пособие

Редактор М. В. Теленкова

ЛР №020640 от 22.10.97

Подписано в печать 1.06.2020. Формат 60×84/16.

Усл. печ. л. 6,98. Тираж 70 экз. Заказ 385. ЭИ № 1530.

Ульяновский государственный технический университет

432027, г. Ульяновск, ул. Северный Венец, д. 32

ИПК «Венец» УлГТУ, 432027, г. Ульяновск, ул. Северный Венец, д. 32