

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

Коллективная монография
под ред. академика РАН С.Н. Васильева

Москва
«Машиностроение»
2010

УДК 004.8
ББК 32.813
И 73

Рецензенты:

чл.-кор. РАН, д-р техн. наук, проф. Себряков Г.Г.
д-р техн. наук, проф. Красильщиков М.Н.

Интеллектуальные системы управления / под ред. акад. С.Н. Васильева.
– М.: Машиностроение, 2010. – 544 с.

И 73 Рассмотрены актуальные проблемы построения интеллектуальных систем управления сложными техническими и организационными объектами. Изложены современные подходы к анализу, синтезу и технической реализации данного класса систем с использованием методов искусственного интеллекта и современных информационных технологий управления и принятия решений.

Для научных работников и специалистов в области создания и эксплуатации интеллектуальных систем управления.

Табл.: 92. Ил.: 350. Библиогр.: 487 назв.

ISBN № 978-5-217-03467-3

Научный редактор академик РАН С.Н. Васильев

Редакционная коллегия:

д-р техн. наук, проф. Р.А. Бадамшин
д-р техн. наук, проф. С.С. Валеев
д-р техн. наук, проф. В.И. Васильев
д-р техн. наук, проф. Х. Вёрн (Германия)
д-р техн. наук, проф. Б.М. Готлиб
д-р техн. наук, проф. М.Б. Гузаиров
д-р техн. наук, проф. Б.Г. Ильясов
чл.-кор. РАН, проф. И.А.Каляев
д-р техн. наук, проф. В.В. Кульба
чл.-кор. РАН, проф. Д.А. Новиков
д-р техн. наук, проф. В.Ю. Рутковский
чл.-кор. РАН, проф. Е.Д. Теряев

УДК 004.8
ББК 32.813



Монография подготовлена в рамках европейской программы TEMPUS
ISBN № 978-5-217-03467-3 © Издательство «Машиностроение», 2010
© Уфимский государственный авиационный технический университет, 2010

УДК 004

С.Г. МАСЛОВ

Ижевский государственный технический университет

СИСТЕМНЫЙ ПОДХОД К ФОРМИРОВАНИЮ СРЕДЫ ПРОГРАММИРОВАНИЯ

В статье развивается системный подход для формирования среды программирования. В рассматриваемом случае программа (информационный артефакт) является продуктом симбиоза между человеком, компьютером и реальностью, который строится из элементов слоев описаний среды программирования (системы понятий, символов, знаков, ощущений). Особенностью подхода является использование понятий симбиоз, «инвариант», степень свободы и управление. Мы стремимся обеспечить непрерывность технологического процесса преобразования идеи в программу (или в общем случае в ИТ-систему), а также обеспечить поддержку эффективности мышления человека в этом процессе с помощью компьютерных средств и когнитивных знаний.

Ключевые слова: системный подход; среда программирования; симбиоз; ИТ-система; ИТ-сфера; инвариант; степень свободы; управление; внешний и внутренний, естественный и искусственный компьютеринг.

ВВЕДЕНИЕ

Современное состояние ИТ-сферы характеризуется явно избыточным разнообразием в создании ИТ-систем, в котором перемешаны языки моделирования, проектирования и программирования, библиотеки модулей и паттернов, приложения и платформы, стили и парадигмы, корпоративные и индивидуальные решения и т.п. Это создает иллюзию оригинальности, гибкости и адекватности применяемых средств, при скрытом дублировании, преобладании формы над содержанием, утрате механизмов эффективного накопления знаний и информации. Естественно, что процент неудачных проектов не уменьшается, например, в США остается на уровне 25%, или приблизительно в 40 млрд. долларов потерь ежегодно [1]. Фактически в ИТ-сфере и в ее разработке сформировался новый кризис, характеризующийся хаотичностью создания, низкой эффективностью потребления и использования информации, когда при изобилии информации и знаний нет нужных, достоверных и своевременных. Кроме того, сложилась парадоксальная ситуация, когда ИТ-специалисты более всего заняты информатизацией всего и всех что их окружает, и менее всего заняты информатизацией своей собственной профессиональной деятельности, действуя по остаточному принципу и довольствуясь в основном эмпирическим уровнем осознания проблем. Однако ситуация постепенно осознается и меняется. Основные усилия исследователей направляются на выявление фундаментальных основ и принципов в создании ИТ-сферы, которые переносят акцент со статических описаний (например, базы данных) на процессные динамические описания.

Данная работа посвящена выявлению фундамента системных представлений для описания среды программирования в широком смысле, т.е. к

описанию информационных процессов и объектов от возникновения идеи до ее реализации (или «материализации»), а также к квантованию и согласованию обработки информации человеком и компьютером, включая когнитивные представления. Основные усилия направлены на выявление слоев дескриптивно-конструктивного описания ИТ-систем.

1. О ПРАКТИКЕ СИСТЕМНОГО ПОДХОДА В ИТ-СФЕРЕ

Системный характер описания и разработки в ИТ-сфере считается настолько естественным, что его часто просто не упоминают. Если обратиться к истории и практике, то в системное русло попадают попытки создания универсальных языков программирования (Алгол-68, Ада), универсальных технологий программирования (R-технология, ESPRIT RAISE, Σ -программирование, ...) и средств проектирования (IDEF, UML, ...), разработка профилей, регламентов, служб и сервисов (ITPL) и другие. Очень часто системному анализу подвергается предметная область, для которой создаются программные средства моделирования. В этом случае средства программирования часто уходят на второй план, а основное внимание уделяется математическому инструментарию, позволяющему создавать системные (обобщенные) модели, которые каждый раз конкретизируются под возникающую задачу. Это можно наблюдать в матричном или тензорном анализе, в тензорном методе двойственных сетей [2]. Другим способом реализации наоборот является создание частной модели в рамках некоторого универсального математического пакета (Mathematica, Maple, MathCad, ...), в котором систематизи-

рованы средства решения классов математических задач на компьютере. Построение систем на базе конструктивной или комбинаторной логики приводит к доказательному программированию и компьютингу, а также формируют единую фундаментальную основу для описания предметной области и средств программирования [3]. Сейчас очень популярны различные онтологические средства, наиболее часто применяемые для задач организации эффективного поиска и описания различных предметных областей (особенно, в веб-среде), перехода к программированию на основе онтологий [4]. В последнее время в область информационных технологий активно встраиваются знания когнитивных наук (знания о психофизиологических, нейрофизиологических характеристиках человека, знания об уровнях когнитивной организации и регуляции интеллекта). Иногда создаются концепции, охватывающие профессиональную деятельность и формы представления знаний и информации, как например, в s-моделировании [5] или в проекте РАСПАС [6].

Несмотря на такой широкий спектр систематизации, остается ощущение незаконченности, фрагментарности, некоторой односторонности (излишней объективности или субъективности), нет непрерывности технологических процессов. Главное нет резонансного процесса синтеза мышления человека и работы компьютера (синтеза естественного и искусственного компьютинга), нет понимания, нет осознанной глубины знаний, нет гармонии осознанного и неосознанного, «костного и живого», «телесного и бестелесного» миров.

Важная концепция для ИТ-сферы с системной точки зрения формируется на основе ЛТ-языка, которая акцентирует внимание на процессных и инвариантных описаниях, измерениях, и в основе которой лежат правила и законы физического мира [7].

Таким образом, чтобы получить необходимый импульс развития и сформировать результативную среду программирования, необходимо двигаться в направлении разрешения перечисленных «нет».

2. ПОСТАНОВКА ЗАДАЧИ

Необходимо определить основные контуры среды программирования, которая позволяет:

- обеспечить непрерывный процесс движения информации от идеи до ее «материализации»;
- обеспечить симбиоз человека с естественной и искусственной средой компьютинга;
- повысить эффективность профессиональной дескриптивно-конструктивной деятельности человека в ИТ-сфере;
- сбалансировать эмпирический и теоретический уровни программирования, снижая трудоемкость и повышая качество процесса программирования.

3. МЕТОД СИСТЕМНОГО ОПИСАНИЯ И ИССЛЕДОВАНИЯ

Решение поставленной задачи будем рассматривать в конструктивном стиле, используя явно и неявно группу вопросов и ответов: «что – объект», «кто – субъект», «какой – свойства», «как – правила», «чем – инструмент», «где – пространство», «когда – время», «сколько – ценность (цена)», «почему – причина», «зачем – цель» [7]. Возможные конструктивные переходы ($=$, $\rightarrow$) при преобразовании некоторых исходных объектов в результирующие опишем в виде следующей конструкции [8]:

$$p:(x:X \rightarrow y:Y(x), N(x,y,p)):c$$

где

X – исходная область объектов;

x – исходная точка отправления;

$Y(x)$ – область результата (иногда зависящая от точки отправления);

p – метод или правило, способствующее переходу $x:X \rightarrow Y(x)$ и $p \in P$;

c – метод или факторы, препятствующие переходу $x:X \rightarrow Y(x)$ и $c \in C$;

N – возможные типы неудач перехода.

Можно выделить следующие типы неудач перехода $x:X \rightarrow Y(x)$:

- N_A – объект вообще не создан или все ресурсы израсходованы;
- N_B – объект создан, но не того качества;
- N_C – создан объект требуемого качества, но неэффективным образом;
- N_D – создан объект требуемого качества, но система ценностей и потребностей субъекта изменилась.

Если под задачей понимать определение неизвестных компонент конструктивного перехода, то фактической средой реализации будет выделенная часть общей среды реализации (E), которая является организованной совокупностью информации и знаний о X , Y , P , C . Каждый конструктивный переход или его компоненту на метафорическом уровне можно представить в виде ствола дерева (или нейрона), у которого есть корни (из чего он построен или откуда принимает информацию) и крона (где он используется или куда выдает свою информацию).

3.1. Направления взаимодействия субъекта с объектом

Рассмотрим «жизненную» ситуацию (S^j), в которой оказывается субъект (Sb) при взаимодействии с воображаемым (im) или реальным (re), искусственным или естественным объектом (Ob , например, ИТ-системой) в рамках некоторой среды (E). Выделим основные конструктивные переходы в системе знаний (K):

Когнитивный: $(Ob \rightarrow K_{Ob})$:

$K_{НАБЛЮДАТЕЛЬ}: Sb$;

Конструктивный: $(K_E \rightarrow K_{Ob})$:

$K_{КОНСТРУТОР}: Sb$;

Креативный: $(K_{Sb} \rightarrow Sb)$: $K_{СОЗДАТЕЛЬ}: Sb$;

Конативный: $(K_{Sb} \rightarrow Sb)$: $K_{НАВИГАТОР}: Sb$;

Коммуникативный:

$(K_{Sb,Ob,E} \rightarrow K'_{Sb,Ob,E})$:

$K_{СОУЧАСТНИК}: Sb$;

Рефлексивный: $(K_{Sb,Ob,E} \rightarrow Ob, K_{Sb,Ob,E})$:

$K_{АДАПТОР}: Sb$.

Необходимо заметить, что здесь возможны переходы между реальным и абстрактным мирами. Кроме того, в качестве основного препятствующего фактора часто выступают ресурсы.

3.2. Общая ситуация конструктивной деятельности

Описанные выше направления и компоненты взаимодействия субъекта и объекта конкретизируются в рамках описания общей конструктивной ситуации (S^d) в следующую совокупность конструктивных переходов.

Все начинается с формулирования цели (G), которая может быть легко достижима или порождает проблему (несовместимость желаемого и возможного).

Целеполагание: $(K_{ЦЕННОСТИ, ПОТРЕБНОСТИ}: E \rightarrow G = K_{Ob}): K_{E, Sb, Ob}$;

Проектирование: $(G$

$\rightarrow$ Проект: $K'_{Ob}): K_{E, Ob}$;

Планирование: $(K_{Об, ИНСТРУМЕНТ}: E$

$\rightarrow$ План: $K''_{Ob}): K_{Sb, Ob}$;

Реализация: $(K'_{Ob} \rightarrow Ob): C$;

Оценка: $(Ob, G \rightarrow K_{Ob, P}): C$;

Принятие решений:

$(K_{Ob, P} \& \text{Вера} \& \text{Воля} \&$

$\rightarrow$ Решение: $K_{Sb}): Sb$.

Важно отметить, что хорошее решение есть взаимодействие интуиции, идеализации, формализации и деформализации. Когда этого не хватает, включается вера, воля и осознанная ответственность за принимаемое решение.

3.3. Слои системного описания и уровни управления

Конструктивные переходы ситуаций S^I и S^{II} непосредственно к успеху не приведут, если знание о их компонентах будут формироваться хаотическим образом. Нужна такая процедура декомпозиции (анализа), которая предусматривает наиболее естественную интеграцию (синтез). Для этого случая выделяются следующие слои описания системы: морфологический, функциональный, атрибутивный, либернетический и генетический. Генетический и либернетический слои объединяются в процессы управления жизненным циклом системы. Метод включает параметрический, структурный, базисный и рефлексивный уровни управления. Рассмотрим их более подробно с точки зрения среды программирования [9].

Морфологическое описание (M), определяющее состав систем, соответствует синтаксису языка. Однако синтаксис, претендуя на правильно построенные конструкции, часто приводит к «невыполнимым» конструкциям, несовершенство которых проявляется в поведении.

Анализ: $(Ob \rightarrow (x: \text{переменные}), (e: \text{элементы}), (r: \text{связи}), (s: \text{структура}): M): C$;

Функциональное описание (P), определяющее динамическое поведение и преобразование систем, соответствует различным семантикам, например, денотационной (исчисление конверсии объектов, динамика их агрегирования и детализации), аксиоматической (аксиомы с содержательной точки зрения делятся на постоянные (законы) и переменные (условия), правила вывода), функциональной и другие.

Анализ: $(Ob \rightarrow \text{правила, преобразования}): P): C$;

Атрибутивное описание, определяющее качественное и количественное проявление систем (Ac, Aq), соответствует логической семантике, описывающей свойства устройства и поведения систем, в частности, их надежность, защищенность, сложные и эргономические аспекты, которые могут быть использованы в процессе построения, выбора и эксплуатации систем.

Анализ: $(M, P \rightarrow Ac, Aq): C$;

Либернетическое описание не имеет явного соответствия в языковой деятельности. Оно концентрирует внимание на инвариантных и вариативных, проекционных и интеграционных особенностях конструкций и поведения системы. Важной

особенностью такого описания является переход от базовых к обобщенным представлениям (координатам) с помощью программ управления, необходимых и достаточных для достижения целей.

В либернетическом описании важно понятие «степени свободы» (F) как независимой изменчивости в описании системы, действующей в формуле [10]: {свободы, организаторы} $\rightarrow$ управление, в которой организаторы фиксируют, редуцируют и генерируют необходимые и достаточные степени свободы, реализуя некоторые целевые акции и реакции.

Анализ: $(M, P \rightarrow F):C;$

Генетическое описание, обнаруживая и преодолевая препятствия и противоречия в построении систем, является основой выявления и концентрации опыта в фиксации устойчивых структур и поведения систем (U).

Анализ: $(M, P, F, C \rightarrow U):C'.$

Если выделены степени свобод, то можно рассмотреть уровни управления. Каждый уровень включает редукторы и генераторы степеней свобод.

Параметрический уровень включает фиксации и регуляции:

$$x_\alpha \rightarrow const, \quad const \rightarrow x_\alpha,$$

а также обратную связь (реакции и рефлексии):

$$x_\alpha \uparrow \rightarrow \downarrow x_\beta, \quad x_\alpha \downarrow \rightarrow \uparrow x_\beta \text{ (отрицательная),}$$

$$x_\alpha \uparrow \rightarrow \uparrow x_\beta, \quad x_\alpha \downarrow \rightarrow \downarrow x_\beta \text{ (положительная).}$$

Структурный уровень связывает группы переменных, принадлежащие элементам (ξ_{im} , i – номер элемента (e_i), m – номер переменной в элементе) или связям (ζ_{kn} , k – номер связи (Γ_k), номер переменной в связи). Можно «рубить» $e_n \rightarrow e_i$, e_j или «склеивать» элементы $e_i, e_j \rightarrow e_n$, изменяя состав переменных. Можно добавлять или удалять связи (Γ_v), вводя синергии и инварианты или освобождая ранее связанные переменные. Интересно, что для достижения цели требуется именно динамический способ изменения активных и пассивных переменных.

Базисный уровень, сохраняя форму, включает изменение характеристик среды реализации (субстанции), например, заменяя механическую среду вычислений на электронную или химическую. Это осуществляется путем изменения базиса переменных:

$$x_i \rightarrow y_k,$$

или изменение интервалов существования переменных:

$$x_i \in [a, b]_i \rightarrow x_i \in [c, d]_i.$$

Динамически может меняться точка зрения:

$$u_i = T_{ip} \bar{u}_p, \quad u_i \in Ox_i, \quad \bar{u}_p \in Az_p$$

или функциональное насыщение (преобразование):

$$H : (F : (u_i \rightarrow u_k) : C^I \rightarrow$$

$$G : (u_i \rightarrow u_k) : C^{II}) : C^{III}$$

Рефлексивный уровень позволяет перевести объекты среды из пассивного состояния в активное, т.е. использовать знание о среде и самих себе, что позволяет организовывать «цепные реакции» для эффективного достижения целей: собственных локальных для объекта и глобальных достигаемых совместно с субъектом.

Примером рефлексии могут служить конструктивные электронные издания, которые активно перестраивают свой материал под решаемую проблему и уровень знаний субъекта.

Важно, что в рамках всех четырех уровней организации степеней свобод отдается приоритет программному управлению, а не простому механизму «стимул – реакция».

3.4. Стратегии компьютеринга

Качественное описание систем естественно предполагает активное использование компьютеринга, «оживляющего» потоки информации и знаний. В рассматриваемом случае компьютеринг является обобщенным процессом, который, в частности, включает вычисления, вывод, поиск, преобразования, навигацию и т.д. Он оперирует качественными и количественными данными.

Идею симбиоза человека с внешней средой компьютеринга, реализуемого на основе описываемого метода системного анализа (и синтеза) и закона природы, можно отобразить в виде следующей графической метафоры (см. рис.1.).

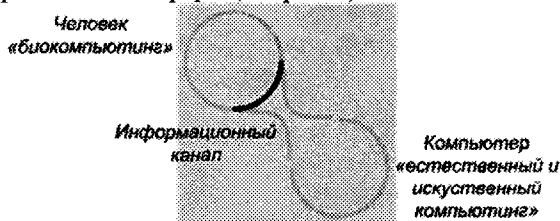


Рис. 1. Симбиоз человека со средой естественного и искусственного компьютеринга.

В этом случае процессы программирования включают концепции стилей, форм, видов и техник как аспекты программирования (более подробно см. в [9]). Эти аспекты формируются при взаимодействии субъектов (Sb) посредством мира знаков и ощущений (WZ, WS), сенсорного мира (SW) в естественном и искусственном мирах (NR, AR).

Таблица 1
Аспекты программирования

Атрибуты программирования	<i>Sb</i>	<i>NR</i>	<i>AR</i>	<i>WS</i>	<i>SW</i>	<i>WZ</i>
Стиль	*	*		*		*
Форма	*		*		*	*
Вид		*	*		*	*
Техника	*		*	*		*

Реальный компьютеринг может осуществляться в синхронных (S) и асинхронных (As) режимах, с учетом некоторых условий и без них, сочетание которых порождает ряд стратегий (табл.2.).

Таблица 2
Матрица стратегий компьютеринга

Исполнение	<i>Безул.</i>		<i>Усл.</i>		<i>Смеш.</i>	
Последовательное	<i>S</i>	<i>As</i>	<i>S</i>	<i>As</i>	<i>S</i>	<i>As</i>
Параллельное						
Конкурентное						
Коллаборативное						
Антагонистическое						

Стратегии компьютеринга должны быть дополнены методами дедуктивного и индуктивного синтеза. Кроме того, часто в реальных задачах можно сначала достаточно точно определить условия результата, а затем восстановить процесс его получения. Так, при вычислении с разреженными матрицами удастся найти структуры для ненулевых элементов и в операциях над этими матрицами произвести вычисления над этими структурами, сразу получив возможные условия ненулевых результатов. Затем именно по этим условиям вычисляются сами ненулевые значения [11].

3.5. Ошибки, препятствия, противоречия и их преодоление

Редко удастся сразу получить качественное решение, если вы решаете достаточно сложную задачу. Ошибки возникают на *стратегическом* уровне, когда плохо осознаются явные и неявные ценности и потребности, на *оперативном и тактическом* уровнях, когда человек тонет в деталях, нарушает границы и интервалы применимости ранее выработанных знаний, преодолевает несогласованности различных уровней описания, часто в условиях дефицита времени. В рамках коллективной и междисциплинарной творческой деятельности человек вынужден преодолевать психологические, культурные, языковые, финансовые, технические, технологические *барьеры*, а также уровни знаний и

их противоречивость. Необходимо единое основание для описания и построения механизмов преодоления перечисленных препятствий.

Таким основанием может служить объединение результатов анализа целей, степеней свобод и уровней управления, которые в простых случаях дают механизмы изменения степеней свобод, т.е. их сокращения или увеличения, их ограничения или генерации и изменения состава. В более сложных случаях осуществляется трансмерный переход из состояния меньшей размерности в состояние большей размерности или переход в новое семантическое пространство на основе эквивалентности решений по ресурсам, а также на основе устойчивых универсальных мер (LT-система [2, 7]).

Главное в разрешении противоречий понять, каких степеней свобод и на каком уровне не хватает или их необходимо устранить, т.е. определить тип противоречия.

Разрешение противоречия можно кратко сформулировать в виде гипотезы конструктивного процесса [8]:

$$F: (m : (g : G = \{L, I\} \rightarrow C(g)) : e = \{L', I'\} \rightarrow S(m; e)) : R,$$

где *I* – исполнитель – Homo Sapiens,

L – профессиональный язык и «язык природы»,

m – метод решения или выявления противоречия в задаче,

e – среда, в рамках которой ищется решение задачи,

L' – языки либернетики, ресурсов и правил их интерпретации,

I' – человеко-машинная система, осуществляющая либернетический переход,

F – запись на языке либернетики,

G – цель (инвариант),

C – противоречие,

R – ресурсы,

S – решение проблемы (баланс).

Попытка достичь цель *G* в среде *e* методом *m* часто приводит не к решению, а к противоречию *C*. В этом случае необходимо исследовать пространство степеней свобод *F* в рамках заданных ресурсов *R*, чтобы получить решение *S*. Решение и в этом случае может быть не найдено, но эффективность этого конечного конструктивного процесса строится как на едином основании анализа и синтеза степеней свобод, так и на возможности формирования и управления стратегиями поиска решения на различных уровнях и слоях. В случаях «слабых» противоречий решения часто строятся методами поиска компромисса (в частности, методами оптимизации) или методами простых рекомбинаций. В случаях «сильных» противоречий необходимы творческие методы разрешения противоречий, разнообразные стратегии поиска решений. Эффективность при этом следует из изучения и установления зако-

номерностей, которые потом закладываются в механизмы организаторов либернетического построения систем. Очень важно как можно раньше понять, что выделенных ресурсов недостаточно или что план нужно перестроить.

3.6. Ресурсы и измерители

Ресурсы – это то, что расходуется и может возобновляться при реализации (или исполнении) планов (или описаний), и то, из чего строятся сами системы [8].

Различаются возобновляемые и невозобновляемые, субъективные и объективные ресурсы. В качестве основных ресурсов можно выделить:

- *временной* – наиболее ценный невозобновляемый ресурс (особенно для субъекта);
- *пространственный* – возобновляемый существенный ресурс;
- *вещественный* – объективный ресурс, может быть аккумулятором и расточителем энергии, носителем информации;
- *энергетический* – наиболее всеобъемлющий, жизненно важный, созидающий и разрушающий, объективный ресурс;
- *информационный* – объективно-субъективный ресурс, мысленных планов и процессов, «следов» взаимодействия, управляющий энергетическими потоками, состоянием и развитием материальной среды, ресурс с минимальным энергопотреблением;
- *финансовый* – наиболее противоречивый, субъективный (относительно-конвенциональный) ресурс, часто «иллюзорный», приводящий к искаженному представлению о действительности, легко перенаправляемый как на созидание (развитие, организацию), так и на разрушение (деградацию, дезорганизацию);
- *интеллектуальный* – организующий (управляющий), анализирующий, синтезирующий, генерирующий формы и связи взаимодействия других ресурсов, наиболее нестабильный ресурс, уровень развития которого сильно зависит от критической массы его в заданной области пространства и времени.

По каждому ресурсу образуется точка эквивалентности, т.е. точка возникновения аналогий или альтернатив в организации живой и неживой материи, а в частном случае, для организации системы знаний как на абстрактном, так и на конкретном уровне. Действуя совместно, они создают узлы перестройки пространства и потоков энергии, вещества и информации, т.е. создают почву для оценки реальности и реализации замыслов в построении систем, с осознанием субъектом сложности, цены и ценности принимаемых решений.

Существенным фактором в системном анализе является возможность сравнения качественных и количественных характеристик проектируемых систем, наличие эталонов для измерения. Неудачи проектов в ИТ-сфере во многом определяются тем,

что показатели не отражают суть системы, например «количество строк программного текста» или «возврата инвестиций» (ROI). Во многом более эффективный и объективный путь уже найден – это система пространственно-временных LT-размерностей Бартини-Кузнецова [7], которая требует дальнейшей конкретизации. Самое важное, что она позволяет устанавливать границы действия различных систем, соразмерять и соизмерять возможности и потребности систем, а также разрабатывать механизмы управления знаниями для устойчивого развития систем.

4. ОБСУЖДЕНИЕ И ПРИЛОЖЕНИЕ РЕЗУЛЬТАТОВ

Рассмотренный выше метод системного описания позволяет строить некоторую систему, прежде всего, в виде многослойной терминологической системы. Для программирования и с когнитивной точки зрения также важна форма и информационные каналы, механизмы компьютеринга, в рамках которых формируются образы, знаки и поведение для концентрации наших знаний. Все перечисленные аспекты также проходят путь предложенного системного описания, в результате чего формируется мощная система аналогий, разнообразных, управляемых и творческих действий субъекта, сокращающие ему время для концентрации необходимых знаний, а также целенаправленного процесса поиска препятствий и их устранения.

Практическая апробация метода осуществлена на следующих задачах:

- управление интеллектуальными ресурсами в ИТ-сфере;
- концептуальная классификация программного обеспечения,
- разработка конструктивных электронных изданий (или публикаций) и фонда электронных ресурсов,
- разработка технологии когнитивного, конструктивного и корректирующего чтения,
- анализ терминологии целенаправленной механики;
- частично, в разработке многоуровневой и мультиязыковой модульной системе программирования: $mx - mi - module-x$.

Характерной особенностью стало то, что решения всех этих задач легко увязываются в непрерывную технологию или процесс формирования индивидуальной системы знаний субъекта (или ИИТ-сферу).

5. БЛАГОДАРНОСТИ

Выражаю искреннюю признательность проф. В.Э. Вольфенгагену, проф. А.П. Бельтюкову, проф. В.В. Смольянову за многочисленные обсуждения различных аспектов рассмотренного подхода.

Работа выполнена при финансовой поддержке РФФИ, грант 08-07-00460-а.

ВЫВОДЫ

Таким образом, предложенный метод системного описания позволяет:

1. Преодолеть ограничения «усеченного системного анализа», используемого при построении онтологий.

2. Строить описание систем в виде многослойной терминологической системы, которая «оживает» при реализации различных конструктивных переходов, погруженных в стратегии компьютеринга.

3. Констатировать, что возможности синтеза информационных, когнитивных и нано-технологий во многом зависят от правильно организованного потока информации между человеком, компьютером и создаваемой системой на абстрактном и физическом уровнях, т.е. контекста ситуаций S^I и S^{II} .

4. Утверждать, что фундаментальный базис системного построения должен включать в себя ресурсный, либернетический и LT-анализ, потому что они целенаправленно расширяют базу эквивалентных преобразований, дополняя психологические методы преодоления инерции мышления человека в процессе творчества.

5. Классифицировать и перейти к более четкому осознанию причин неудач и противоречий, а также перейти к их предупреждению и разрешению.

СПИСОК ЛИТЕРАТУРЫ

1. Макконнелл С. Профессиональная разработка программного обеспечения. - Пер. с англ. - СПб.: Символ-Плюс, 2006. 240 с.

2. Кузнецов О.Л. Итоговый научный отчет за 2008 - 2009 годы по гранту Президента Российской Федерации для государственной поддержки ведущих научных школ РФ НШ-1269.2008.9 / О.Л. Кузнецов, Б.Е. Большаков, А.Е. Петров, // Дубна, 2009. (http://www.uniduba.ru/departments/sustainable_development/Portal/science_returns/?id=1554)

3. Вольфенгаген В. Э. Аппликативные вычислительные технологии. Готовые решения для инженера, преподавателя, аспиранта, студента / Под ред. к.т.н. Л.Ю. Исмаиловой / М.: ЗАО «ЮрИнфоР», 2009. 64 с.

4. Клещев А.С. Использование онтологий в разработке программного обеспечения // Всерос. конф. «Знания – Онтологии – Теории» с междунар. участием 14-16 сентября 2007 г. Новосибирск: Институт математики им.С.Л. Соболева. Т.1. С.122-130.

5. Ильин В.Д. Символьное моделирование в человеко-машинной среде: основы концепции / В.Д. Ильин, И.А. Соколов // Информационные технологии и вычислительные системы. 2008. №1. С. 51-60.

6. Нечипоренко А.В. На рубеже знаниевых технологий // Кентавр, №32. С.8-19. (<http://www.circleplus.ru/archive/n/32/2>)

7. Большаков Б.Е. Системный анализ методов управления знаниями в области устойчивого развития / Б.Е. Большаков, Е.Ф.Шамаева // Устойчивое инновационное развитие: проектирование и управление 2009, №4. С. 39-54. (www.typravlenie.ru)

8. Бельтюков А.П. Либернетическая парадигма в IT-сфере / А.П. Бельтюков, С.Г. Маслов, О.А. Морозов // Технологии информатизации профессиональной деятельности (в науке, образовании и промышленности): Сб. тр. II Всероссийской науч. конференции с междунар. участием. Часть I. Ижевск.: ООО Информационно-издательский центр «Бон Анца», 2008. С.37-52.

9. Бельтюков А.П. Конструктивные механизмы в IT-сфере / А.П. Бельтюков, С.Г. Маслов, О.А. Морозов // Вестник Удмуртского университета. Сер. Математика. Механика. Компьютерные науки. - 2009. - Вып. 2. - С. 102-109.

10. Смолянинов В.В. Либернетика – наука о свободах системной организации // Мост. 2000. №1. - С. 28-30.

11. Маслов С.Г. Структурные аспекты моделирования сложных систем // НТИ. Сер.2., Информационные процессы и системы. М.: ВИНТИ, №3, 1993. С. 7-9.