

НОВОСТИ ИИ. – 2000. – №3.

МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ «ИНТЕЛЛЕКТУАЛЬНЫЕ САПР-98»

КРУГЛЫЙ СТОЛ «НУЖНЫ ЛИ НЕТРАДИЦИОННЫЕ ЛОГИКИ В ИНТЕЛЛЕКТУАЛЬНЫХ САПР?»

Основные участники: Дмитрий Владимирович Бутенко, Вадим Николаевич Вагин, Елена Юрьевна Головина, Александр Павлович Еремеев, Виктор Михайлович Курейчик, Олег Николаевич Пьявченко, Валерий Борисович Тарасов

В.Н.Вагин (Московский энергетический институт)

Я предлагаю затронуть сегодня на нашем Круглом столе следующие основные проблемы, относящиеся к интеллектуализации систем автоматизированного проектирования.

1. Что такое интеллектуальные САПР (ИСАПР)? Каковы принципы их построения и основные компоненты?
2. Что такое знания? В чем состоит специфика проектных знаний?
3. Какие языки представления лучше подходят для ИСАПР?
4. Каковы механизмы обработки знаний в ИСАПР?
5. Что такое НЕ-факторы знаний и как их моделировать?

На мой взгляд, интеллектуализация САПР представляет собой усиление традиционных систем автоматизированного проектирования новыми информационными технологиями, основанными на знаниях. Для повышения эффективности САПР, особенно на ранних стадиях разработки, требуется внедрить в состав их средств системы инженерии проектных знаний. Системы инженерии знаний – это программные системы, снабженные специальным инструментарием переработки трудноформализуемых сведений, включающим стратегии поиска, управление базами знаний, механизмы вывода и пр. Еще раз отмечу, что эти системы на основе знаний никоим образом не подменяют обычные программные модули САПР, такие как подсистемы геометрического моделирования, пакеты программ расчетов или базы данных по конструкциям и материалам. Речь идет именно о совместном использовании интеллектуальных компонентов и традиционных технологий (см.[1-3]).

Не пытаюсь сейчас построить какую-либо общую архитектуру подобной гибридной системы, я ограничусь определением интеллектуальной подсистемы САПР как семиотической системы. Такое представление об интеллектуальных САПР восходит к известной гипотезе А.Ньюэлла, согласно которой «необходимое и достаточное условие для того, чтобы некоторая физическая система могла проявлять интеллектуальность, состоит в том, чтобы она являлась универсальной системой формальных манипуляций над конкретными символами» (фактически это условие эквивалентно определению машины А.Тьюринга).

Логический подход к ИСАПР охватывает теоретические и практические проблемы представления проектных знаний – понятий, суждений, умозаключений. Речь идет об автоматизированном формировании проектных понятий и связей между ними, а также о компьютерной обработке суждений и рассуждений проектировщиков. Чем же отличаются семиотические системы от обычных логических систем? Прежде всего, это открытость и динамичность семиотических систем.

Исходным элементом семиотической системы является знак. Знак включает в себя три аспекта: имя знака (синтаксический аспект), содержание знака (семантический аспект), назначение знака (прагматический аспект). В проектной деятельности для представления инженерных знаний широко используются разнообразные знаки, причем синтаксис соответствует структуре проектных знаний, семантика – их смыслу, содержанию, а прагматика – их использованию.

Семиотические системы по Д.А.Поспелову являются расширением классических формальных систем. По сути, семиотические системы представляют собой такие динамические системы, текущие состояния которых можно описать с помощью формальных систем. Особенность любой семиотической системы проявляется, прежде всего, в ее открытости, так как в ней по определению имеются: 1) правила, изменяющие синтаксис формальной системы; 2) правила, изменяющие саму систему исходных постулатов, аксиом, общих закономерностей; 3) правила

изменения набора правил вывода. Данные системы принимают во внимание не только синтаксические, но и семантические и прагматические аспекты обработки знаний, и будучи по своей сути динамическими системами, поддерживают рассуждения здравого смысла и методы реализации немонотонных, модальных, абдуктивных и нечетких рассуждений.

В.М.Курейчик (Таганрогский государственный радиотехнический университет)

В свое время одно из первых определений ИСАПР как гибридной экспертной системы предложил Д.А.Поспелов [4, 5]). Он связывал архитектуру ИСАПР с гибридной экспертной системой, включающей интеллектуальный интерфейс, базу знаний, решатель, вычислитель, блок имитационного моделирования. Мы много спорили на эту тему.

В.Б.Тарасов (Московский государственный технический университет им. Н.Э.Баумана)

Но то, что Вы упомянули, есть, вообще, определение гибридной экспертной системы. Почему оно отождествляется с ИСАПР? Все-таки методология и архитектура гибридных экспертных систем – лишь одна из возможных. Мне кажется, что структура ИСАПР может быть переменной и различной. Ее конкретная реализация должна зависеть по крайней мере от таких факторов как стадия проектирования и уровень интеллектуализации (индивидуальный, групповой или организационный) [3]: не следует считать гибридную экспертную систему панацеей, поскольку на групповом уровне речь может пойти о многоагентной системе, а на начальных стадиях проектирования – о семиотической базе знаний.

В.Н.Вагин (Московский энергетический институт)

Из всех метафор, используемых для определения понятия «знание» (вспомним, например, евангельскую метафору «выращивания зерна» или метафору «зажженного факела»), мы будем придерживаться следующей: «знание – это обоснованное истинное убеждение». Эта метафора удивительно перекликается с определением знания, данным в Большой энциклопедии под редакцией С.Н. Южакова, СПб, т. 9, 1902, с. 688, где сказано, что «знание... – убеждение в его истине по реальным причинам».

Любое убеждение, которое находится в базе знаний семиотической системы, становится знанием, если его истинность подтверждается (обосновывается) новыми фактами, объяснениями или наблюдениями [6].

Конечно, в идеале хорошо было бы иметь дело с полным и непротиворечивым знанием, описывающим предметную область. В классических логических системах свойство полноты обычно формулируется следующим образом: для множества формул с заданными свойствами исходная система постулатов и правил вывода должна обеспечить вывод всех формул, входящих в это множество. Важно также свойство непротиворечивости, суть которого сводится к тому, что исходная система постулатов и правил вывода не должна давать возможность выводить формулы, не принадлежащие заданному множеству формул с выбранными свойствами. Например, полные системы аксиом и правил вывода в классическом исчислении предикатов первого порядка позволяют получить любую формулу из множества общезначимых формул и не дают возможности получить какие-либо формулы, не обладающие этим свойством. От полноты информации, содержащейся в базе знаний, зависит и полнота ответов на запросы, предъявляемые в этой базе

Если в базу знаний, содержащую сведения о студентах некоторого вуза, поступил запрос, является ли Иванов студентом этого вуза, то ответ будет положительным, если это на самом деле так. В случае неудачи с этим запросом возможны два варианта: а) Иванов – не студент и б) статус Иванова не известен. Первый вариант имеет место, если мы придерживаемся предположения о замкнутости мира (Рейтер), когда рассматриваются факты, которые не могут быть найдены (выведены) в базе знаний, как их отрицания. О втором варианте трудно что-либо сказать. Таким образом, если в базе знаний содержатся сведения не о всех студентах, то, естественно, такая неполная база и будет давать ответы только о тех студентах, сведения о которых в ней содержатся.

В проектировании, как и во многих других практических областях, получить полную и непротиворечивую базу знаний, как правило, невозможно. Это определяется спецификой проектных знаний. В практике НИОКР достаточно высок удельный вес плохо формализуемых или совсем не формализуемых знаний. Поэтому трудно (а чаще всего, невозможно) сформулировать исходные (априорные) свойства, которым должны удовлетворять формулы вывода в ИСАПР как семиотической системе.

В.Б.Тарасов (Московский государственный технический университет им. Н.Э.Баумана)

Вообще говоря, семиотизация знания связана с построением логико-алгебраических моделей. Мне кажется, что построение логико-алгебраических и топологических теорий проектирования (а первые варианты подобных теорий уже созданы, я могу упомянуть здесь общую теорию

проектирования Йошикавы [7] и работы известного логика Треура [8] – см. также [9]) является неотъемлемым аспектом разработки интеллектуальных САПР.

Но вначале, несколько слов о проектных знаниях. Специфика баз проектных знаний состоит в том, что они включают знания о предметной области, знания об ограничениях и знания о решениях проблемы. Кроме того, можно выделить различные типы проектных знаний по признакам их глубины и жесткости. По первому признаку, проектные знания подразделяются на глубинные и поверхностные. Поверхностные знания основаны на интерпретации отдельных фрагментов индивидуального опыта проектантов и конструкторов, для них не существует никакой единой теории. Наоборот, глубинные знания основаны на некоторой фундаментальной теории, например, законах механики.

Глубинные знания всегда более устойчивы, абстрактны и общепризнанны, тогда как поверхностные знания, будучи скорее локальными, объясняют отдельные факты, подтверждают отдельные идеи и конкретные гипотезы. Разумеется, дихотомию «поверхностные – глубинные» знания можно перевести и в плоскость «полных – неполных» знаний и в плоскость «противоречивых-непротиворечивых знаний».

По второму признаку, проектные знания подразделяются на жесткие и нежесткие (мягкие). Показателем жесткости служат точность, однозначность, возможность количественного описания с помощью уравнений, алгоритмов и реализации в виде традиционных пакетов программ. Мягким знаниям свойственны неточность, многозначность, нечеткость. Для их представления и обработки в компьютере нужны специальные методы и модели, например, лингвистические переменные, нечеткие множества.

Представляется очевидным, что интеллектуализация САПР должна проводиться, исходя из идеи дополнительности указанных типов знаний.

В.Н.Вагин (Московский энергетический институт)

С этим трудно не согласиться.

Итак, представить проектные знания – это значит выразить в некотором формализме имеющийся у нас образ изделия (создаваемого продукта). Соответствие между продуктом и его представлением можно установить путем семантического анализа. Следовательно, представление должно позволить осуществлять анализ истинности высказываний о мире, т.е. позволить определить, что в этом мире влечет истину, а что – ложь. Здесь и кроется основная трудность семантического анализа знаний: истинностное значение высказывания может быть как известным, так и неизвестным, независимо от того, какое оно на самом деле.

Формализуя строгие, корректные рассуждения, классическая логика, к сожалению, не принимает во внимание некоторые аспекты реальных человеческих рассуждений (здравый смысл, противоречивость, неопределенность, неточность, нечеткость информации). Имея дело с неполной, неточной, противоречивой, неопределенной или нечеткой информацией (все эти слова, начинающиеся с приставки «не» с легкой руки А.С.Нариньяни [10] стали называть НЕ-факторами знаний), человеческие рассуждения чаще всего предположительны и лишь правдоподобны, динамичны и должны подвергаться пересмотру (ревизии).

Жесткие формальные системы классической логики не подходят для моделирования нестрогих, подвергающихся пересмотру проектных рассуждений. Кроме того, в классических формальных системах между формулами определяется отношение выводимости \tilde{A} , которое:

- рефлексивно: $A_1, A_2, \dots, A_n \tilde{A} A_i, i = 1, \dots, n;$
- транзитивно: если $A_1, A_2, \dots, A_n \tilde{A} B$ и $A_1, A_2, \dots, A_n, B \tilde{A} C$, то $A_1, A_2, \dots, A_n \tilde{A} C;$
- монотонно: если $A_1, A_2, \dots, A_n \tilde{A} B$, то $A_1, A_2, \dots, A_n, C \tilde{A} B.$

Эти фундаментальные свойства вывода формализуют требования к общезначимым рассуждениям

- вывод заключения, являющегося одной из посылок, есть общезначимая операция;
- промежуточные результаты можно использовать для установления общезначимости заключения;
- полученный результат не опровергается добавлением новых посылок.

Таким образом, свойство монотонности препятствует прямой формализации пересматриваемых рассуждений. Следовательно, с чисто синтаксической точки зрения построение немонотонной системы вывода делает необходимым ослабление свойств дедуктивных систем классической логики [6, 11].

Что касается противоречивости проектных знаний, то ее последствия в силу потери разницы между истиной и ложью в логических конструкциях могут быть поистине разрушительными. Отметим, что при представлении и обработке противоречивых знаний возникают две проблемы. Первая касается ассимиляции (усвоения) противоречивой информации, т.е. способности включать противоречия в базу знаний и возможности работать с противоречивой базой знаний.

Вторая проблема заключается в аккомодации (приспособлении) противоречивого знания, т.е. такой модификации базы знаний, при которой она становится непротиворечивой. Эти две задачи тесно связаны между собой, и от их совместного решения зависит общая проблема вывода утверждений в противоречивой базе знаний.

На первый взгляд, исходя из здравого смысла, проблема ассимиляции противоречий может быть разрешена путем введения порядка реализации вывода, когда сначала пытаются вывести рассуждения, связанные с частными фактами, а затем уже рассуждения, относящиеся к общим законам. Трудность нахождения порядка реализации вывода заключается, конечно, в семантике утверждений, которая в общем случае неочевидна. Можно говорить, что утверждение B выводимо из посылок A_1, A_2, \dots, A_n , если каждая посылка релевантна этому утверждению. Но построение такой релевантной логики применительно к проблемам проектирования связано с большими трудностями. В данном случае можно вспомнить об идеях ситуационной семантики, где выводы делаются только в пределах той ситуации, которая имеет место в данный момент. Естественно, при переходе к другой ситуации должна производиться ревизия базы знаний и выведенные ранее утверждения могут оказаться неприменимыми для вывода новых.

Что касается аккомодации противоречия, то здесь проблема заключается в обработке «подозрительных» формул (т.е. формул, которые могли бы быть противоречивыми) как гипотез. Здесь возникают три задачи: во-первых, каким образом установить непротиворечивость «подозрительных формул» среди множества других формул; во-вторых, как формально установить и описать множество общих утверждений, выдвигаемое как гипотезы, и в-третьих, каким образом эти гипотезы могут взаимодействовать с другими утверждениями для вывода новых.

Вернемся к такому важному НЕ-фактору знаний как немонотонность, часто проявляющемуся в семиотических системах. Немонотонность означает, что пересматриваемые рассуждения не являются в классическом смысле общезначимыми и если утверждение B выводимо из посылок A_1, A_2, \dots, A_n , то существует модель для $\{A_1, A_2, \dots, A_n\} \cup \{F\}$, не подтверждающая B , где $\{F\}$ – множество новых посылок. Отметим, что проблема немонотонности тесно связана с такими НЕ-факторами знания, как неполнота и противоречивость.

Теперь обратимся к таким НЕ-факторам знания как неточность и неопределенность. Если неточность относится к содержанию информации, то неопределенность – к ее истинности, понимаемой в смысле соответствия реальной действительности (степени уверенности в том или ином знании). Каждый факт реального мира связан с мерой определенностью информации, которая указывает на степень этой уверенности. Понятия «определенности» и «уверенности» довольно трудно формализуемы и для их определения чаще всего используются количественные меры. Основная идея такой меры заключается в введении функции неопределенности $unc(p)$, понимаемой как определенность того, что высказывание p , содержащееся в базе знаний, истинно. Для этих целей разработаны вероятностная и возможностная логика. Нечеткая же логика была адаптирована для обработки недоопределенных, нечетких и неоднозначных знаний.

Например, заключение «Петр юн» не является общезначимым следствием из двух посылок: «Большинство студентов юны» и «Петр – студент». Это заключение просто выполнимо с этими посылками. Конечно, если мы не имеем никакой дополнительной информации, то *по умолчанию* полагаем: «Если вы не знаете ничего другого, то предположите, что все студенты действительно юны». Тогда и Петр, будучи студентом, также юн.

При поступлении в базу знаний новой информации предположения могут стать невыполнимыми с новым множеством посылок и будут отвергнуты. Узнав, например, что Петр не юн и имея, таким образом, противоречивую базу знаний, мы должны или отвергнуть ранее выведенное заключение, или потребовать дополнительной информации.

Таким образом, одна из версий немонотонных рассуждений реализована в логике умолчания Рейтера [12], в которой немонотонность обусловлена необщезначимостью правил вывода, присущих той или иной прикладной области. Одно из таких правил умолчания следующее:

$$D: \frac{\alpha(x_1, \dots, x_n): M\beta(x_1, \dots, x_n)}{\gamma(x_1, \dots, x_n)},$$

где $\alpha(x_1, \dots, x_n)$, $\beta(x_1, \dots, x_n)$, $\gamma(x_1, \dots, x_n)$ – формулы языка логики предикатов первого порядка, свободные переменные у которых выбраны среди x_1, \dots, x_n ;

$\alpha(x_1, \dots, x_n)$ – предпосылка умолчания D ;

$\beta(x_1, \dots, x_n)$ – обоснование умолчания D ;

$\gamma(x_1, \dots, x_n)$ – следствие умолчания D ;

M – некоторый оператор, интерпретируемый как «непротиворечивый».

Это правило гласит: « Если для конкретных значений x_1, \dots, x_n утверждение α истинно и предположим, что утверждение β непротиворечиво с α , тогда предполагаем, что и утверждение γ также истинно».

Умолчание называется *нормальным*, если $\beta(x_1, \dots, x_n) = \gamma(x_1, \dots, x_n)$. Например, правило умолчания вида

$$\frac{\text{Личность}(x): \text{МСтудент}(x)}{\text{Студент}(x)}$$

устанавливает, что если кто-то является личностью и непротиворечиво предположить, что этот кто-то является студентом, то действительно предполагаем, что кто-то – студент. В этой логике в зависимости от правил умолчаний имеются только правдоподобные предположительные заключения. Иногда их можно считать общими правилами, которые истинны в большинстве случаев, но допускают исключения. Для такой логики существует несколько множеств выводимых утверждений и каждое из этих множеств может рассматриваться как допустимое множество убеждений в этой логике.

В другой немонотонной логике, предложенной Мак-Дермоттом и Дойлом [13], также вводится оператор М, интерпретируемый как «непротиворечивый». Тогда утверждение типа: «Обычно студенты юны» будет записано следующим образом:

$$\forall x (\text{Студент}(x) \ \& \ \text{МЮн}(x) \rightarrow \text{Юн}(x)),$$

т.е. если кто-то является студентом и это не противоречит тому, что он юн, то этот кто-то действительно юн. Если в базу знаний вводится информация типа «Петр – студент» и невозможно вывести $\neg\text{Юн}(\text{Петр})$, т.е. $\text{МЮн}(\text{Петр})$ истинно, то можно сделать вывод о том, что Петр юн.

Однако система вывода здесь немонотонна, так как введение добавочной информации может блокировать предыдущий вывод. Например, добавив новый факт о том, что Петр не юн, мы отвергнем первоначально заключение, так как в данном случае $\text{МЮн}(\text{Петр})$ не будет истинным.

Трудность вывода в немонотонной системе Мак-Дермотта заключается в том, что понятие «непротиворечивости» здесь довольно «слабое» в том смысле, что истинностные значения утверждений Р и МР не связаны между собой никаким отношением, т.е. пара $\{\text{МР}, \neg\text{Р}\}$ необязательно может быть противоречивой [6]. Мур развивает идею немонотонного вывода дальше, вводя два типа рассуждений по умолчанию [14]. Первый тип рассуждений характеризуется тем, что он имеет дело с фактами, касающимися внешнего мира: вообще объекты типа Х имеют свойство Р. Если А – объект типа Х, то можно сделать вывод, что А (по-видимому) обладает свойством Р. Например, если Петр – студент, то можно сделать вывод, что (по-видимому) Петр юн.

Второй тип рассуждений, названный *автоэпистемическими* рассуждениями, касается рассуждений, связанных с чьими-то убеждениями (мнениями). Этот тип рассуждений пересматривается, исходя из текущего состояния знаний агента, из его интроспективной природы. Например, если мне ничего неизвестно о том, что Петр не юн, то я делаю вывод, что Петр, будучи студентом, юн.

Механизм рассуждений интроспективен и основан на предположении о том, что все знания, касающиеся этого вопроса, таковы: «Если бы Петр был не юн, то я бы об этом знал». Иначе говоря, можно потребовать от рассуждений «общезначимости» относительно этого состояния знаний.

Для представления такой информации, семантического ее анализа и обоснования выводов и разработаны «нестандартные» логики, так сказать логики с нетрадиционной ориентацией, являющиеся расширением классических логик. Эти расширения касаются языка логики и понятия вывода.

О.Н. Пьявченко (Таганрогский государственный радиотехнический университет)

Мне непонятно, как все эти неклассические логики могут применяться в реальной жизни проектантами. У них имеются конкретные задачи, которые решаются с помощью хорошо известных методов и инструментальных средств. Все они четко определены и структурированы (см., например, [15, 16]). Конечно, иногда встречаются и инновационные, творческие задачи, решение которых опирается на опыт, интуицию, талант проектантов. Но причем тут логики?

В.Б. Тарасов (Московский государственный технический университет им. Н.Э.Баумана)

В поиске ответа на этот вопрос я хотел бы дать свои комментарии по двум проблемам, вынесенным на обсуждение Вадимом Николаевичем, – а именно, по первой и последней. Мне представляется очень важным делом развитие концепции современных САПР как семиотических систем. Это особенно актуально для начальных стадий проектирования, при формулировании технического задания и разработке технического предложения, когда центральное место занимают проблемы поддержки диалога между заказчиком и производителем (поставщиком) и

проблемы интерпретации текстов спецификаций, написанных на ограниченном естественном языке. Как организовать компьютерную поддержку ранних стадий разработки? Здесь возможный аппарат формализации может опираться на давно известные (но в настоящее время переживающие в рамках прикладной семиотики второе рождение) идеи ситуационного анализа и семиотического моделирования [17]).

Остановимся на самом главном, что отличает интеллектуальные САПР от обычных, неинтеллектуальных. На мой взгляд, специфика ИСАПР состоит в моделировании процесса проектирования, а в более широком плане, в управлении жизненными циклами как продукта проектирования (товара), так и процесса, так и средств проектирования. Грубо говоря, ядро ИСАПР = (система моделей продукта + система моделей процесса проектирования). И как раз моделирование процесса проектирования требует привлечения нетрадиционных логик. Именно здесь, Олег Николаевич, они и нужны. Почему?

Процесс проектирования представляет собой важнейшую стадию жизненного цикла продукции, который также включает стадии ее производства и эксплуатации. Жизненный цикл (ЖЦ) охватывает период от формирования потребности в продукте (изделии) и формулирования общих требований к нему до разрушения (утилизации) продукта. В последние годы, в контексте идей «контролируемого производства – поддерживаемого в рамках допустимой нагрузки на окружающую среду» (sustainable manufacturing), активно развивается представление об «обратном жизненном цикле» (inverse life cycle), включающем процессы, связанные с рекуперацией и повторным использованием отработанной продукции. Мы остро нуждаемся в специальных методиках управления жизненными циклами с использованием компьютерных моделей ЖЦ. Разработка таких методик может включать следующие шаги: а) нелинейное (например, спиральное) представление реального ЖЦ, его детальное структурирование и определение основных параметров (см., например, [18, 19]); б) создание математических методов и моделей имитации ЖЦ и управления его основными параметрами; в) построение архитектуры соответствующих программных средств и систем; г) реализация, отладка и практическое использование инструментальных средств поддержки управления ЖЦ.

Кроме того, не так давно были предложены новые, интересные стратегии реорганизации (реинжиниринга) собственно процессов проектирования. Прежде всего, к их числу относятся «совмещенное или параллельное проектирование» (concurrent design или simultaneous design) [20] и «кооперативное проектирование» (cooperative design или collaborative design).

В узком смысле под совмещенным проектированием понимается одновременная разработка продукта (изделия) и технологического процесса его изготовления. В более широком аспекте совмещенное проектирование требует кооперации разнородных агентов, поскольку оно предполагает объединение усилий специалистов и подразделений внутри промышленной фирмы, а также организацию ее тесного взаимодействия с заказчиками, партнерами и подрядчиками на всех стадиях разработки создаваемого изделия (и, прежде всего, на ранних этапах его проектирования).

Подходы, охватываемые термином «совмещенное проектирование» роднит то, что они стремятся учесть влияние практически всех аспектов жизненного цикла системы на исходный проект. Идеология совмещенного проектирования предполагает рассмотрение на стадии проектирования целой гаммы факторов, относящихся к различным этапам жизненного цикла разрабатываемого класса изделий и влияющих на его стоимость и эффективность, в интересах улучшения функциональных характеристик изделия, сокращения сроков проектирования и производства, уменьшения затрат на создание системы.

В свою очередь, понятие «кооперативного проектирования» ставит акцент на улучшении процессов коммуникации между различными отделами и службами предприятия (агентами проектирования) в ходе проектирования некоторого продукта и, вообще, предполагает более интенсивное общение и сотрудничество между заказчиком и исполнителем. В современных условиях такое сотрудничество может быть организовано в рамках виртуального предприятия – сетевой, компьютерно интегрированной организационной структуры, которая состоит из неоднородных взаимодействующих агентов, расположенных на удалении друг от друга, в различных местах [21]. Эти агенты разрабатывают совместный проект (или ряд взаимосвязанных проектов), находясь между собой в отношениях партнерства, кооперации, сотрудничества, координации и т.п. Поэтому здесь главную роль играют лингвистические и логико-лингвистические модели, т.е. требуется рассмотрение семиотических аспектов проектирования. Подходы семиотического моделирования лежат в основе построения распределенных баз проектных данных/ знаний для кооперативного концептуального проектирования.

Когда говорят о виртуальном предприятии, речь идет об интенсивном взаимодействии реально имеющихся специалистов и подразделений различных предприятий в виртуальном пространстве с

помощью новейших информационных и коммуникационных технологий. Это взаимодействие призвано повысить уровень кооперации и координации партнеров, а в конечном итоге, конкурентоспособность производимой ими продукции и, соответственно, прибыль.

В частности, виртуальное проектное бюро как базовая информационно-коммуникационная структура кооперативного проектирования может быть образовано путем отбора требующихся человеческих, организационно-методических и технологических ресурсов из разных проектных бюро и их компьютерной интеграции, что приводит к формированию гибкой, динамичной виртуальной организационной метасистемы, наиболее приспособленной для скорейшего выпуска новой продукции и ее оперативной поставки на рынок. Очевидно, что создание виртуального проектного бюро связано с распределением проектных функций и полномочий между организациями-партнерами и реализацией дистанционного проектирования.

Типичная инфраструктура виртуального проектного бюро для осуществления кооперативного и совмещенного проектирования может включать следующие основные составляющие: сеть Internet/Intranet, международный стандарт для обмена данными по моделям продукции STEP (Standard for the Exchange of Product model data), «надстроенный над ним» стандарт по управлению знаниями KIF и стандарт на взаимодействие прикладных программ CORBA (Common Object Request Broker Architecture).

Итак, эффективное воплощение идей и принципов совмещенного и кооперативного проектирования связано с применением методологии ИИ, позволяющей интегрировать различные данные и знания, относящиеся ко всем стадиям жизненного цикла изделия, объединять неоднородные проектные и производственные модели и процедуры рассуждений, строить многоуровневый интеллектуальный интерфейс для специалистов разного профиля. Так, например, интеллектуализация совмещенного проектирования предполагает использование систем, основанных на знаниях, для преобразования последовательного жизненного цикла разрабатываемого объекта (системы) в параллельно-последовательный с целью уменьшения числа итераций в процессе проектирования, что приводит к снижению сроков и стоимости разработки. В целом, анализ инженерного метазнания в системах автоматизированного проектирования/ производства CAD/CAM и моделирование жизненного цикла продукции позволяют обосновать привлечение комплексного показателя «совмещения» (conspicency), включающего некоторый конечный набор НЕ-факторов (нечеткость, неопределенность, непоследовательность, немонотонность), что оказывается хорошим подспорьем в определении базовой системы нетрадиционных логик в контексте ИСАПР.

Подобный минимальный базис нетрадиционных логик совмещенного и кооперативного проектирования, определяющих соответствующий вариант технологии разработки ИСАПР, может быть представлен в виде пятерки <логика интеграции, модальная логика, немонотонная логика, нечеткая логика, логика неопределенности> [22]. Здесь первый компонент отражает несовершенство гибридной интеллектуальной системы, связанное с проблемой интеграции разнородных моделей при проектировании, включая символичные (логические), образные (коннекционистские), эволюционные, численные и графические модели.

Следующие два компонента базиса нетрадиционных логик проектирования связаны с двумя неотъемлемыми характеристиками любого процесса проектирования – а) частичным совмещением, запараллеливанием, кооперативным выполнением ряда проектных процедур; б) итерационным характером, наличием этапов перепроектирования и корректировки проектной документации, связанных с отказом от ранее принятых решений и пересмотром имеющихся знаний. Так второй компонент, определяемый требованиями кооперативности и параллельности разработки (simultaneous engineering), предполагает учет различных характеристик и параметров жизненного цикла разрабатываемого объекта на начальных стадиях проектирования. Это может, в частности, пониматься как оперирование в различных мирах, что связано с применением модальных логик, интерпретируемых с помощью моделей Крипке, которые позволяют устанавливать необходимые соотношения между мирами. Третий компонент – немонотонные логики – позволяет моделировать возврат назад, на более ранние этапы проектирования, с целью ревизии ранее принятых решений.

Последние два компонента – нечеткая логика и логика неопределенности – ориентированы на решение проблем обработки качественной, нечисловой информации и управления неопределенностью при проектировании.

Мне кажется, что в отличие от специалистов по логике (занимающихся доказательством полноты и непротиворечивости) и классическому ИИ, проектанты изделий и систем, а также разработчики САПР давно привыкли на практике иметь дело с НЕ-факторами, с неполной, противоречивой и неточной информацией, и не видят в этом ничего оригинального и сверхъестественного. Их не стоит «агитировать за державу», т.е. убеждать, что НЕ-факторы в

проектировании действительно существуют и с ними надо работать. Они это и так прекрасно понимают. Следует сосредоточиться на том, чтобы достаточно простым, понятным для инженеров, языком объяснить, как эти факторы учитываются в нетрадиционных логиках, разработанных для ИИ.

В целом, построение интегрированной интеллектуальной системы, основанной на принципах совмещенного проектирования, предполагает использование семейства неклассических логик, отражающего учитываемые НЕ-факторы; синтетической подсистемы рассуждений с модулем правдоподобной аргументации; средств когнитивно- и коммуникативно-графической поддержки.

В.М.Курейчик (Таганрогский государственный радиотехнический университет)

У меня возникает вопрос, как соотнести предлагаемые взгляды на ИСАПР как семиотическую систему, концепцию совмещенного проектирования и методы работы с НЕ-факторами с традиционно выделяемыми по ГОСТ уровнями проектирования и видами обеспечения САПР. При проектировании технических объектов обычно рассматривается несколько вертикальных уровней – функциональный, алгоритмический, конструкторский, технологический [15, 23]. И четко оговаривается наличие различных видов обеспечения – технического, информационного, математического, программного, лингвистического, методического, организационного. Можно ли считать, Валерий Борисович, что предлагаемые подходы затрагивают все уровни или этапы проектирования и все виды обеспечения? Или речь идет о каких-то отдельных уровнях, например, только уровне функционального проектирования? Сохраняется ли в целом традиционная структура компонентов и подсистем САПР и соответствующие виды обеспечения? Или надо все ломать, пересматривать требования старых ГОСТ и вводить принципиально новые стандарты?

В.Б. Тарасов (Московский государственный технический университет им. Н.Э.Баумана)

Это очень интересный вопрос, Виктор Михайлович, достаточно исчерпывающий ответ на который даст только время, практика создания и реального использования ИСАПР в различных отраслях. На мой взгляд, тут надо говорить не о ломке, а о постоянном развитии, пополнении стандартов, подобно тому, как это происходит со стандартом STEP. С одной стороны, конечно уровни проектирования и виды обеспечения тесно взаимосвязаны, и включение, скажем, семиотических моделей в математическое обеспечение САПР неминуемо приведет к изменениям лингвистического, программного и методического обеспечения. В этом смысле, речь идет о сквозной интеллектуализации проектирования. С другой стороны, мне кажется, что потребность в интеллектуализации на различных уровнях, для различных задач и стадий проектирования, конечно, различна. Сейчас я говорю как о проектировании в машиностроении, так и о создании самих САПР, опираясь на традиционные представления об основных стадиях разработки: техническое задание, техническое предложение, эскизное проектирование, техническое проектирование, рабочее проектирование (см. например, [24-26]). Абстрагируемся пока от идей совмещенной разработки. Так вот в случае рутинных, хорошо известных задач, или на стадии технического проектирования, когда уровень проектной неопределенности небольшой, легко обойтись безо всякой интеллектуализации. Здесь потребность в ней сравнительно невелика. Ожидаемый эффект также ограничен. В принципе можно обойтись традиционными средствами САПР. Напротив, на этапе анализа технического задания или разработки технического предложения, при решении трудноформализуемых, творческих задач исследовательского проектирования [27]), выборе целей и стратегий проектирования, определении основных признаков и общего облика проектируемого объекта потребность в интеллектуализации резко возрастает. Особенно это касается организации и деятельности метасистем, например, МетаСАПР [28] или МетаКИП [29]. Для наглядности данный тезис о зависимости потребности в интеллектуализации от характеристик продукта и процесса проектирования можно изобразить в виде следующего графика в трехмерном пространстве «Этапы проектирования – степень инновации проектирования – потребный уровень интеллектуализации» (см. рис.1).

А.П. Еремеев (Московский энергетический институт)

Я бы хотел ответить на замечание О.Н.Пьявченко, о том, что при проектировании и управлении сложными техническими системами нечеткости и неопределенности быть не должно, что этот процесс достаточно хорошо детерминирован и что поэтому неясна здесь роль нетрадиционных логик и, в частности, нечетких логик и моделей вычислений.

Сегодня проблема улучшения управления автоматизированным проектированием стоит очень остро, хотя на эту тему уже появилось немало работ, и что мне особенно приятно отметить, одна из первых отечественных монографий по управлению процессами проектирования написана специалистами из ТРТУ [30].

Действительно, когда процесс жестко детерминирован, можно обойтись классическими моделями поиска решения, позволяющими находить четкие и оптимальные решения. Процесс функционирования системы может быть описан при этом или марковской моделью типа

$$s(t+t') = f(s(t), u(t), v(t)),$$

где $s(t)$ - состояние системы в момент времени t ,

$s(t+t')$ - состояние через заданный временной интервал t' ;

$u(t)$ - контролируемые и управляемые параметры;

$v(t)$ - неуправляемые контролируемые или неконтролируемые параметры, образующие фон или внешнее окружение объекта;

f - функция управления,

или фиксированным набором продукционных правил типа

$$\{S_i(t) \rightarrow R_i(t)\}, i=1, \dots, n,$$

где $S_i(t)$ – текущее состояние,

$R_i(t)$ - принимаемое решение.

В последнем случае решение находится путем использования классического правила вывода *modus ponens* ($S_i(t), S_i(t) \rightarrow R_i(t) \Rightarrow R_i(t)$).

Однако в режиме реального управления сложной системой, детерминированная, четкая модель поиска решений применима далеко не всегда. Рассмотрим, например, управление энергоблоком, в особенности, случай, когда система перешла в нештатный режим функционирования. Это возможно по разным причинам, в частности, из-за неверных показателей или вообще отказа некоторых датчиков, данные которых $u(t)$ и $v(t)$ используются при поиске управляющих решений.

В этой ситуации для поиска корректирующего решения по управлению можно применить приближенную немарковскую модель типа

$$s(t+t') = f(s(t), s(t-t'), \dots, s(t-nt'), u(t), u(t-t'), \dots, u(t-nt'), v(t), v(t-t'), \dots, v(t-kt')),$$

базирующуюся на анализе предыстории поведения управляемой системы. Конечно, нахождение оптимального решения здесь не гарантируется.

Другой пример. При использовании для поиска решения при управлении или проектировании (особенно, поисковом проектировании) продукционной модели, приведенной выше, возможно возникновение некоторого состояния $S'_i(t)$, в определенной степени аналогичного (похожего) на состояние $S_i(t)$ в соответствующем продукционном правиле, но не тождественного (эквивалентного) ему. Классическое правило вывода *modus ponens* в этом случае просто не применимо. Но решение (некоторое приближение к $R_i(t)$) можно найти, используя средства нечеткой логики, в частности, «нечеткий *modus ponens*» вида

$$(S_i(t) * (S_i(t) \rightarrow R_i(t))),$$

где $*$ некоторая операция композиции нечетких отношений, задаваемая экспертом.

И такие примеры можно продолжить.

Подводя итог сказанному, отметим, что нетрадиционные логики и модели поиска решения необходимы там и тогда, где и когда классические логики и точные модели вычислений неприменимы или в силу отсутствия таких моделей, или в силу невозможности их применения из-за ресурсных (как правило, временных) ограничений (например, при управлении в режиме реального времени) [31, 32]. Типичными примерами таких ситуаций являются управление в открытых, динамических проблемных областях, а также поиск решения в нештатных ситуациях. А в области разработки САПР – это поисковое проектирование и управление проектированием.

Е.Ю.Головина (Московский энергетический институт)

Мне кажется, здесь следует также упомянуть о роли нетрадиционных логик на этапе стратегического планирования в CASE-технологии. В настоящее время в технологии разработки программных средств происходит смена парадигмы с «программной инженерии» (Software Engineering) на «информационную инженерию» (Information Engineering). Информационная инженерия представляет собой совокупность методологий и программных средств, поддерживающих создание информационных систем, автоматизирующих деятельность человека [33]. Главной отличительной особенностью информационной инженерии по сравнению с программной инженерией является наличие методов и программных средств, поддерживающих этап стратегического планирования жизненного цикла программного обеспечения. На этапе стратегического планирования осуществляется обследование деятельности организации с целью повышения эффективности бизнеса. По сути, речь идет о разработке ИСАПР деятельности организаций.

Для автоматизации работ этапа стратегического планирования на кафедре Прикладной математики МЭИ создается программное средство, поддерживающее следующие функции.

1. Определение целей деятельности организации и ограничений, накладываемых на них организацией заказчика (физические операции, структура организации, финансовые ресурсы и т.д.), окружением организации (потребители, конкуренты, законы рынка), разработчиками информационной системы (существующая технология и т.д.).
2. Построение иерархий типа «цель/подцель» с ограничениями. Определение важности целей и ограничений («взвешивание» иерархии).
4. Определение «взаимного влияния» целей, т.е. «кооперирования целей», «взаимного препятствования целей», рекурсивных ситуаций «цель способствует достижению цели», «цель препятствует достижению целей».
5. Выбор наилучшего пути достижения поставленной цели.

В качестве математической модели используется И/ИЛИ-граф, расширенный в соответствии с требованиями проблемной области. Выбор наилучшего пути достижения поставленной цели основан на эвристическом поиске в И/ИЛИ-графе. Эвристическая функция строится на основе аппарата нечеткой логики (треугольных норм и конорм), позволяющего определять различные стратегии достижения поставленных целей в соответствии с заданными ограничениями и ресурсами.

Д.В.Бутенко (Волгоградский государственный технический университет)

Я считаю, что следует коснуться вопросов качества продукции, создаваемой с помощью ИСАПР. Точнее, характера ее развития с позиций концепции инновационного маркетинга. Поскольку развитие технических систем осуществляется по объективным законам эволюционной теории, логично использовать выявленные законы для прогнозирования развития. Можно предсказать последовательность замены существующих на рынке технических продуктов и осуществлять опережающее планирование выдвигания на рынок тех или иных товаров. Иными словами, можно управлять появлением на рынке новых продуктов и, следовательно, обеспечивать конкурентоспособность предприятий. Как это делать?

Здесь за основу можно взять слегка видоизмененную иерархическую схему организации процесса концептуального проектирования, предложенную А.И.Половинкиным [34] (рис.2).

1) ПОТРЕБНОСТЬ
2) ФУНКЦИОНАЛЬНАЯ СХЕМА
3) ПРИНЦИП ДЕЙСТВИЯ
4) ТЕХНИЧЕСКОЕ РЕШЕНИЕ
5) ПАРАМЕТРИЧЕСКАЯ РЕАЛИЗАЦИЯ

Рис.2. Иерархия понятий при описании технических объектов

Необходимо построить модель каждого из базовых понятий, используемых в этой схеме. В частности, потребность можно описать тройкой $P = (D, G, H)$, где D указывает на действие, приводящее к удовлетворению данной потребности, G – предмет этого действия, а H – указание особых условий или ограничений, при которых выполняется действие. Например, D – «перевозка», G – «грузы», H – «большие расстояния». Потребность определяет назначение технического объекта и его функции. Функциональная схема обычно представляет собой ориентированный граф, вершинами которого являются наименования элементов, а ребрами – их функции. В свою очередь, принцип действия опирается на понятие физического эффекта и понимается как исходная структура объединенных и совместимых физических эффектов, обеспечивающих преобразование заданного входного воздействия в требуемый конечный результат. Наконец, техническое решение представляет собой конструктивное оформление

принципа действия, которое часто моделируется (И-ИЛИ)-деревом, а параметрическая реализация переводит абстрактное техническое решение в конкретный, законченный продукт.

К числу общих законов развития техники относятся закон соответствия между функцией и структурой, закон корреляции параметров однородного ряда, закон симметрии и т.п.

Концептуальное проектирование характеризуется неполнотой используемой информации и необходимостью обеспечения максимального числа инноваций. Поэтому проектант применяет такие методы рассуждения как индукция и абдукция, ассоциативный вывод и рассуждения по аналогии. Очень важную роль играет поиск противоречий.

В ВолГТУ школой А.И.Половинкина – В.А.Камаева активно развиваются методы разрешения технических противоречий, многие из которых были предложены еще в работах Г.С.Альтшуллера. Формируются деревья противоречий, отражающие противоречия на отдельных уровнях функционально-структурной организации системы, например, противоречия уровня системного, логического, схемотехнического проектирования и т.д.

Наиболее эффективной стратегией синтеза вариантов нового технического решения мы считаем качественное, логическое моделирование, названное контрадукцией. Контрадукция предполагает использование формы логического вывода в виде выявления, формирования и разрешения противоречий для получения лучших вариантов решений. С помощью фонда типовых эвристических приемов разрешения противоречий формируется массив проекций будущей концептуальной структуры при фиксированных параметрах, из которого посредством индуктивного вывода формируется идея новой структуры.

Особое внимание при исследовании уделяется методикам, позволяющим облегчить синтез новых идей и основывающихся на принципах ассоциативности человеческого мышления. Здесь в качестве теоретической основы были взяты известный метод гирлянд ассоциаций и его давний прототип, так называемый логографический диалог М.В.Ломоносова.

К настоящему времени, в ВолГТУ разработана система ассоциативного поиска решений для различных проблемных областей «ИНСПИРАТОР» [35], использующая эвристические функции языка и визуальных образов. Исходной моделью, представляющей систему связанных понятий (ассоциаций), служит произвольный граф, вершины которого представляют понятия, а ветви – связи между этими понятиями. Также реализована система маркетинга, обеспечивающая функции слежения за состоянием рынка, потоков финансов и продукции. Наконец, построена система наполнения банка потребностей предприятия для реализации ассоциативной цепи «потребность– цели – функции».

Итак, в интересах компьютерной поддержки ассоциативного поиска инновационных решений требуется создание банка знаний с ассоциативными процедурами вывода, а в первую очередь, языка обработки данных для принятия решений на ассоциативной сети.

В.Н.Вагин (Московский энергетический институт)

Мне кажется, что идеи контрадукции, о которых говорит наш коллега, представляют собой вариант абдуктивных рассуждений. Мы много занимаемся этими проблемами на кафедре Прикладной математики МЭИ и будем рады возможному сотрудничеству в области моделирования абдуктивных рассуждений в ИСАПР.

В.М.Курейчик (Таганрогский государственный радиотехнический университет)

Итак, наша сегодняшняя дискуссия достигла своей цели. Мы пришли к общему выводу, что ИСАПР есть обычная САПР плюс надстройка, обеспечивающая работу со знаниями, ассоциативный поиск и т.п. Что инженерия проектных знаний имеет особую специфику и требует синтеза различных видов рассуждений. И что нетрадиционные логики и модели учета НЕ-факторов безусловно нужны и имеют свою сферу применения, особенно на начальных стадиях проектирования, при моделировании процесса проектирования и управлении проектированием. Я благодарю всех участников сегодняшнего Круглого стола и приглашаю продолжить нашу дискуссию через год здесь же, на берегу моря. Надеюсь, что к нам присоединятся и другие заинтересованные специалисты.

Литература (включая рекомендуемые работы участников Круглого стола)

1. Вагин В.Н., Клишин В.В., Филиппова О.В. Опыт реализации экспертных систем в области машиностроения, предназначенных для решения задач проектирования и планирования// Новости искусственного интеллекта. – 1991. – №4. – С.11-22.
2. Курейчик В.М. К вопросу создания интегрированных интеллектуальных САПР СБИС// Автоматизация проектирования электронной аппаратуры. Вып.5. – Таганрог: ТРТИ, 1988. – С.8-13.

3. Тарасов В.Б. Интеллектуальные системы в проектировании// Новости искусственного интеллекта. – 1993. – №4. – С.24-67.
4. Поспелов Д.А. Моделирование рассуждений. – М.: Радио и связь, 1989.
5. Поспелов Д.А. Ситуационное управление – основа прикладных интеллектуальных систем //Вестник МГТУ: Сер. «Приборостроение». – 1995. – №2. – С.3-10.
6. Вагин В.Н. Зачем нужны нетрадиционные логики?//Международный форум информатизации 98. Доклады Международной конференции «Информационные средства и технологии». – М.: Станкин, 1998. – С.6-14.
7. Yoshikawa H. General Design Theory as a Formal Theory of Design// Intelligent CAD I/ Ed. by H. Yoshikawa, D.Gorsard. – Amsterdam: Elsevier Science Publishers, 1989. – P.51-61.
8. Treur J.A. A Logical Framework for Design Processes// Intelligent CAD Systems III: Practical Experience and Evaluation/ Ed. by P.J.W. ten Hagen and P.J.Veerkamp. – Berlin: Springer Verlag, 1991. – P.3-19.
9. Advances in Formal Design Methods for CAD// Ed. by J.Gero. – Melbourne: Chapman and Hall, 1996.
10. Нариньяни А.С. НЕ-факторы и инженерия знаний: от наивной формализации к естественной прагматике// Сборник трудов Национальной конференции по ИИ КИИ-94 (Рыбинск, сентябрь 1994). В 2-х томах. Т.1. – Тверь: АИИ, 1994. – С.9-18.
11. Тейз А., Грибомон П., Луи Ж. и др. Логический подход к искусственному интеллекту: от классической логики к логическому программированию: Пер. с франц. – М.: Мир, 1990.
12. Reiter R. A Logic for Default Reasoning// Artificial Intelligence. – 1980. – Vol.13. – P.81-132.
13. McDermott D., Doyle J. Non-Monotonic Logic I// Artificial Intelligence.– 1980.– Vol.13.– P.41-72.
14. Moore R.C. Semantical Considerations on Nonmonotonic Logic// Proceedings of IJCAI-83, Karlsruhe, 1983, p.272-279.
15. Корячко В.П., Курейчик В.М., Норенков И.П. Теоретические основы САПР. – М.: Энергоатомиздат, 1987.
16. Курейчик В.М., Калашников В.А., Лебедев Б.К. Автоматизация проектирования печатных плат. – Ростов: Изд-во РГУ, 1984.
17. Поспелов Д.А. Ситуационное управление: теория и практика. – М.: Наука, 1986.
18. Тарасов В.Б. О путях и принципах реализации совмещенной разработки машиностроительных конструкций// Интеллектуальные САПР. Межведомственный тематический научный сборник. Вып.5. – Таганрог: ТРТУ, 1995. – С.36-42.
19. Кашуба Л.А. Параллельное проектирование средствами CAD/CAM/CAE в жизненном цикле изделий машиностроения// Программные продукты и системы. – 1998. – №3. – С.24-31.
20. Смирнов А.В., Юсупов Р.М. Совмещенное проектирование: необходимость, проблемы внедрения, перспективы. – СПб: СПИИРАН, 1992.
21. Тарасов В.Б. Новые стратегии реорганизации и автоматизации предприятий: на пути к интеллектуальным предприятиям// Новости искусственного интеллекта.–1996.– №4.–С.40-84.
22. Тарасов В.Б. От искусственного интеллекта к искусственной жизни: новые направления в науках об искусственном// Новости искусственного интеллекта. –1995. – №4. – С.93-117.
23. Курейчик В.М. Математическое обеспечение конструкторского и технологического проектирования с применением САПР. – М.: Радио и связь, 1989.
24. Быков В.П. Методическое обеспечение САПР в машиностроении. – Л.: Машиностроение, 1989.
25. Норенков И.П., Маничев В.Б. Основы теории и проектирования САПР. – М.: Высшая школа, 1990.
26. Володин В.В. Автоматизация проектирования летательных аппаратов.–М.: Машиностроение, 1991.
27. Валькман Ю.Р. Интеллектуальные технологии исследовательского проектирования. – Киев: Port-Royal, 1998.
28. Норенков И.П. Разработка САПР. – М.: Изд-во МГТУ, 1994.
29. Тарасов В.Б. Концепция МетаКИП: от компьютерно-интегрированного производства к Internet/Intranet-сетям предприятий// Программные продукты и системы.–1998.–№3.–С.19-22.
30. Малышев Н.Г., Мицук Н.В. Основы оптимального управления процессами автоматизированного проектирования. – М.: Энергоатомиздат, 1990.
31. Башлыков А.А., Еремеев А.П. Экспертные системы поддержки принятия решений в энергетике. – М.: МЭИ, 1994.
32. Еремеев А.П. Экспертные модели и методы принятия решений. – М.: МЭИ, 1995.

33. Modern Software Engineering. Foundation and Current Perspectives// Ed. by P.A. Ng, R.T. Yeh. – New York: Van Nostrand Reinhold, 1990.
34. Половинкин А.И. Основы инженерного творчества. – М.: Машиностроение, 1988.
35. Бутенко Д.В., Бутенко Л.Н. Концепция маркетинга, основанного на закономерностях развития технических систем (инновационный маркетинг) // Известия ТРТУ. – 1998. – №2. – С.257-258.