

УДК 681.32

ХАРАКТЕРИСТИКА АРХИТЕКТУРЫ КОМПЬЮТЕРНОЙ СИСТЕМЫ, ПРИБЛИЖЕННОЙ К ИДЕАЛЬНОЙ

А. Ф. Кургаев

Институт кибернетики НАНУ, г. Киев, Украина

Для решения выявленных проблем развития архитектуры современных компьютеров впервые предложено принять структуру систем обработки знаний (СОЗ) за образец архитектуры компьютерных систем (КС), а в качестве образца структуры СОЗ — идеал структуры научной теории, поскольку именно в ней явно и наиболее концентрированно выражена форма мышления людей при решении проблем. Сформулированы наиболее существенные признаки идеальной архитектуры КС, непротиворечиво сочетающие универсальность со специализацией КС на решение произвольных проблем.

АРХИТЕКТУРА, МОДУЛЬ СЕМАНТИЧЕСКИЙ, МЕТАЯЗЫК, СИСТЕМА КОМПЬЮТЕРНАЯ.

Постановка задачи

Все выявленные в глубине веков прототипы не имеют отношения к современным компьютерам, их нельзя квалифицировать иначе, чем примитивное воплощение эвристической мечты об искусственной сущности с неограниченными возможностями.

Создание современных компьютерных систем (КС) ведет начало от теоретических моделей-открытий К. Геделя, А. Черча, А. Тьюринга, Э. Поста и др., полученных при решении проблем разрешимости в конструктивной логике, и тезиса Черча-Клини, по сути, предлагающего считать эти модели разными вариантами формального определения интуитивного смысла терминов «разрешимый предикат» и «эффективно вычислимая функция» [1–5]. В последующем было доказано точное совпадение классов функций, вычисляемых с использованием этих моделей. Однако, будучи эквивалентными в этом, они существенно различны в отношении используемых выразительных средств.

Модели А. Тьюринга и Э. Поста (далее Т-машины) оказались весьма близкими *явными* формализациями понятия алгоритма и *неявными* — эффективно вычисляемых функций (или предикатов) через возможность построения для каждой из них соответствующей программы (Т-машины).

Две другие модели представляются разными формальными системами:

- основанная на результатах Д. Гильберта теория рекурсивных функций К. Геделя задает в *явной* форме правила определения *декларативной* семантики вычисляемых функций (и разрешимых предикатов) и в *неявной* форме *процедурную* семантику — правила вывода;

- функциональное λ -исчисление А. Черча задает в *явной* рекурсивной форме правила вывода (*процедурную* семантику) и *неявно* — *декларативную* семантику вычисляемых функций (и разрешимых предикатов).

Для Т-машин фон Нейман обосновал конечность программ каждой из них, возможность коди-

ровать целыми числами, хранить в памяти программы вместе с данными и использовать эту пару (программы и данные всякой Т-машины) в качестве исходных данных для универсальной «исполнительной» программы (U-машины Тьюринга [6]). Это решение определило двухуровневую архитектуру первых компьютеров и дало возможность их реализации даже примитивными средствами электроники.

К особенностям КС с архитектурой Тьюринга-фон Неймана (ТН-архитектурой) обычно относят [7–9]:

- линейную память, в которой программы хранятся вместе с данными;
- последовательное выполнение программы;
- отсутствие различий между командами и данными и в семантике данных;
- чрезмерную универсальность.

Следствиями этих особенностей являются проблемы [7–9]:

- неизбежность «кризиса программирования», связанного с высокой трудоемкостью и низкой эффективностью программирования, непрерывно возрастающей сложностью, ассортиментом и низкой надежностью программного продукта;
- сложность достижения параллелизма в решении задач;
- низкая эффективность использования физических ресурсов (памяти и быстродействия) современных КС и технологии микроэлектроники.

В настоящее время перечисленные особенности и проблемы общепринято [10] объяснять процедурным характером исходной вычислительной модели КС с ТН-архитектурой и ее последующим воплощением в форме императивных языков программирования современных компьютеров.

Совершенствование показателей качества КС в непрерывно меняющихся условиях их создания, производства и применения в конечном итоге до-

стигается за счет эволюции их архитектуры. Используя термин «архитектура» (процессора, компьютера, системы), концентрируют внимание на семантической модели специфичной компоненты соответствующего изделия:

- процессах обработки данных (со многими АЛУ, многопроцессорные, конвейерные, теговые, архитектуры потока данных, RISC-архитектуры, матричные и т. п.);
- функциональной структуре (процессора, памяти, других устройств), определяющей взаимосвязь управления с обработкой данных; общей функциональной структуре КС и др.

В этой области накоплено большое количество методологически значимых и технических решений, достойных внимания архитекторов новых КС. Среди нововведений обычно различают КС с архитектурой, ориентированной на язык высокого или сверхвысокого уровня, группу языков программирования или высокого уровня (КС с гибкой архитектурой).

Целая серия методологически значимых архитектурных нововведений была реализована коллективом под руководством В.М. Глушкова в компьютерах семейства «МИР», объективно получивших непревзойденную в течение многих лет популярность в СССР среди исследователей и других категорий пользователей. Среди нововведений отметим: диалоговый характер взаимодействия с пользователями в процессе постановки и решения проблем, ставший доминирующим в современных КС, и микропрограммную интерпретацию языка, семантически эквивалентного текстовому.

Ряд методологически значимых архитектурных нововведений реализован в микропроцессорах семейства «Электроника С5», созданных Институтом кибернетики АН УССР совместно ЛКТБ ЛОЭП «Светлана» в 70-х годах прошлого века на основе опыта разработки мини-ЭВМ М-180 — одной из первых в СССР с микропрограммным управлением. Среди этих нововведений отметим мультиоперационную обработку данных, реализованную уже в первых моделях микропроцессоров этого семейства, и микропрограммную эмуляцию, реализованную в первом в СССР однокристалльном 16-разрядном микропроцессоре компьютера «Электроника С5-21» [11].

Целью настоящей статьи является определение характеристик архитектуры КС, приближенной к идеальной.

1. Требования к архитектуре КС

С целью создания «более благоприятной среды для программистов и корректных программ» были выдвинуты «шесть основных требований, которым должна удовлетворять новая система» [9, с. 292]:

- обнаружение семантических ошибок в программах;
- ограничение последствий возможных ошибок программирования;
- усовершенствование процесса программирования;
- упрощение процесса программирования;
- рост эффективности тестирования и отладки за счет специальных средств;
- упрощение операционных систем и компиляторов.

В разработках современных КС перечисленные требования постоянно учитываются, обеспечивая непрерывный прогресс КС, но их архитектура еще далека от совершенства:

1. Большой семантический разрыв между языками программирования и языком описания проблем, используемым людьми в науке и практике.
2. Столь же велик семантический разрыв между технологией решения проблем в науке и практике и технологией программирования.
3. Рост коэффициента вторичного использования программного продукта (в свое время по расчетам А.П. Ершова он составлял не более 1,5 [12]) достигается преимущественно за счет тиражирования и комплексирования фирменного системного программного обеспечения.
4. Структура аппаратной компоненты КС статична, не изменяется естественно, в ходе решения задачи. Всякое изменение структуры требует программирования.

2. Основные положения для определения характеристик архитектуры КС, приближенной к идеальной

Основная идея достижения цели состоит в принятии структуры системы обработки знаний (СОЗ) за образец архитектуры КС, а в качестве образца структуры СОЗ — идеал структуры научной теории, поскольку именно в ней явно и наиболее концентрированно выражена форма мышления людей при решении проблем.

Обоснованные современными достижениями логико-гносеологических и лингвистических исследований в [13–18] выполнены анализ и синтез наиболее существенных признаков идеала логической структуры системы научных теорий, непротиворечиво сочетающие формальную строгость с содержательностью знаний научных теорий. Эти результаты обобщаются в форме следующих утверждений.

Утверждение 1. Структура системы научных теорий в развернутой (в пространстве или во времени) форме состоит из четырех блоков идентичной структуры — знаков, синтаксиса языков, семантик и функций научных теорий, взаимосвязанных меж-

ду собой, с источниками-приемниками знаковых образов и с реальным миром.

Утверждение 2. Структура каждого из блоков представляется иерархической сетью модулей идентичной структуры, каждый из которых включает словарь и интерпретаторы концептуальных и эмпирических моделей синтаксиса, семантики и прагматики некоторой подтеории.

Утверждение 3. В свернутой в пространстве (но способной к развертыванию во времени) форме структура научных теорий представима структурой половины модуля (в силу симметрии его узлов), содержащей память словаря, интерпретатор концептуальных моделей с памятьми, содержащими концептуальные модели синтаксиса, семантики и прагматики разных подтеорий, и интерпретатор эмпирических моделей.

Утверждение 4. Функционирование всякого модуля определяется решением задач анализа и синтеза структуры некоторого представления информации. Всякая задача всякой подтеории научной теории представляется итерационно-рекурсивной структурой.

Утверждение 5. Множество функций блока функций включает дескриптивно-прескриптивные и аргументативные функции, под управлением которых осуществляются все деятельностные процессы становления, эволюции и использования всякой научной теории.

3. Наиболее существенные признаки идеальной архитектуры КС

Наиболее существенные признаки идеальной архитектуры КС определим в согласии с требованиями устранения недостатками.

1. Первый недостаток устраним предоставлением пользователям всех категорий метаязыка, выразительные возможности которого адекватны представлению знаний произвольной прикладной теории. Тем самым проблема семантического разрыва сводится к проблеме разработки семантического метаязыка с неограниченными выразительными возможностями.

2. Второй недостаток обуславливает проблему операционной системы, предоставляющей пользователям технологию решения проблем, адекватную

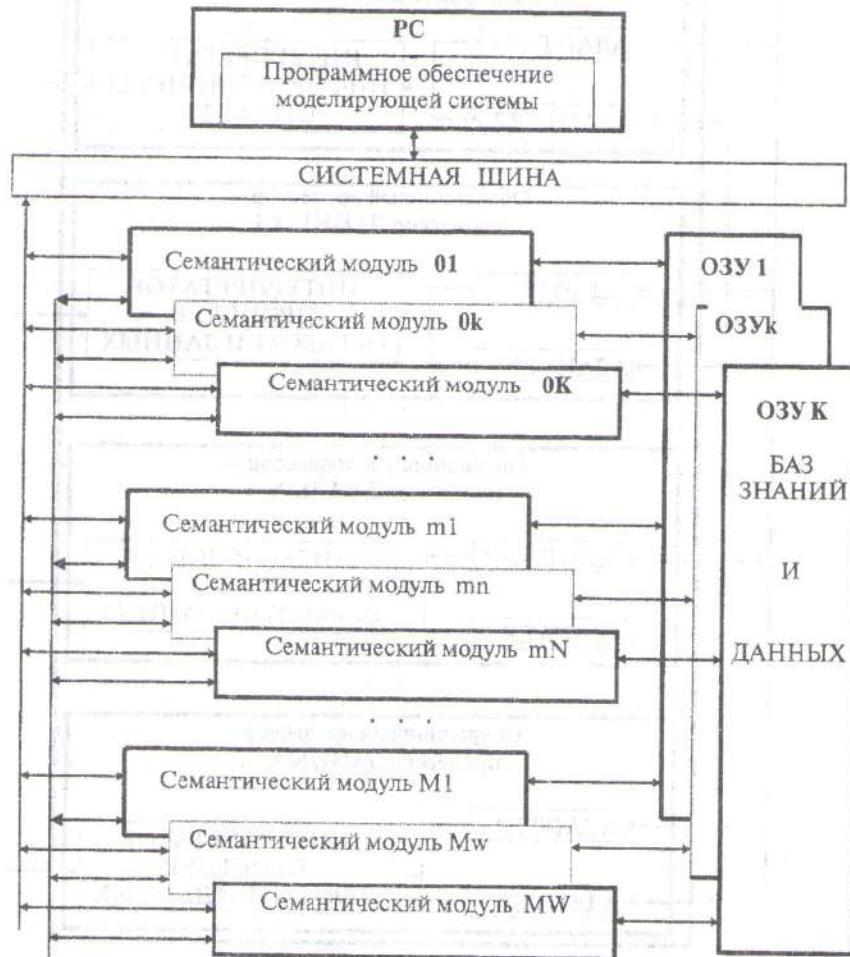


Рис. 1. Моделирующая система с распределенными функциями

используемой в науке и практике, т. е. сводится к проблеме реализации моделей всех функций научных теорий.

3. Третий недостаток устраним реализацией в операционной среде функций накопления знаний для многократного использования (включая разные приложения), модификации и пополнения знаний прикладной теории.

4. Структура аппаратной компоненты КС при решении всякой прикладной проблемы должна быть тождественна структуре соответствующей прикладной теории. Наиболее адекватным решением этой проблемы являются архитектуры КС с распределенными функциями.

4. Архитектура КС, приближенная к идеальной

4.1. Архитектура моделирующей КС содержит (рис. 1): управляющий компьютер РС с программным обеспечением моделирующей системы и сис-

темной шиной, к которой подключены многоярусная ($m = 0 + M$), сеть семантических модулей из $k = 1 + K$ семантических модулей яруса 0, $n = 1 + N$ семантических модулей яруса m и $w = 1 + W$ семантических модулей яруса M , взаимосвязанных между собой и с $k = 1 + K$ блоками памяти.

Все семантические модули сети имеют тождественную структуру (рис. 2): управляющий процессор (процессор баз знаний, ПБЗ) состоит из интерпретатора информационных структур, подключенного к блоку памяти баз знаний, и двух адаптеров, один из которых соединен с системной шиной управляющего компьютера РС, а второй — с шиной управления операционными процессорами семантического модуля и с ПБЗ других семантических модулей; всякий ($j = 1 + J$) операционный процессор (процессор данных, ПД) состоит из интерпретатора процедур обработки данных, подключенного к блоку памяти данных, и двух адаптеров, один из

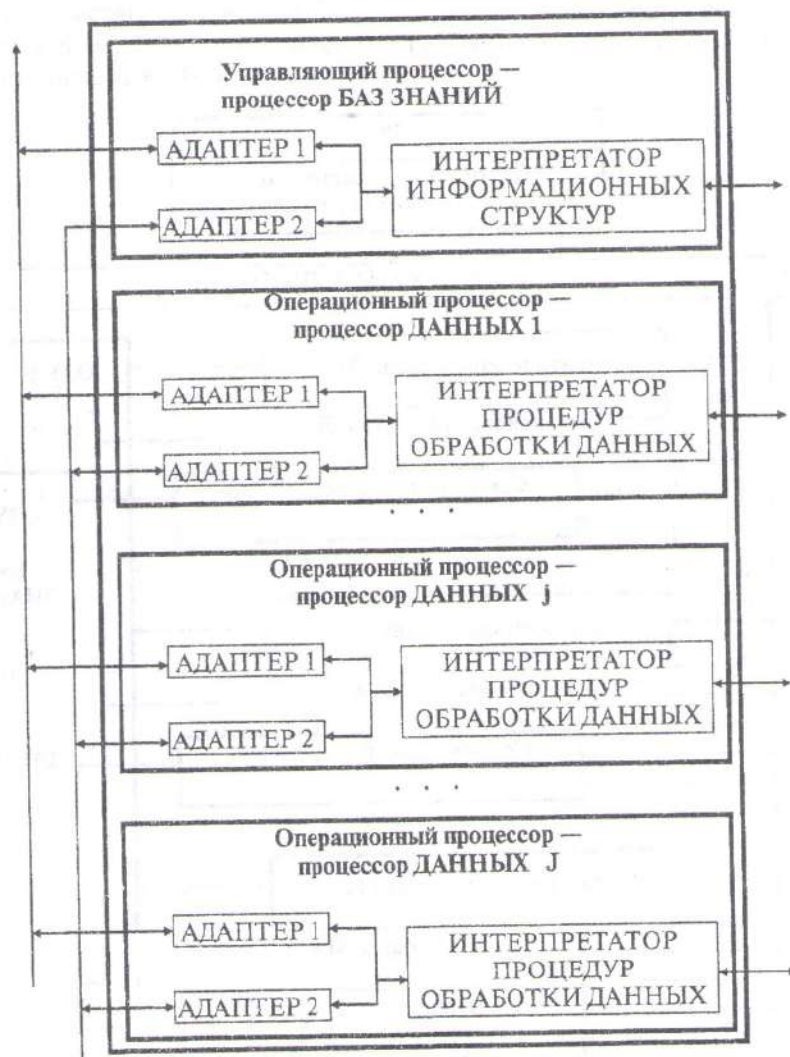


Рис. 2. Структура семантического модуля

которых соединен с системной шиной управляющего компьютера РС, а второй — с шиной управления.

В этой системе управляющий компьютер РС выполняет настройку системы и под управлением сети семантических модулей в процессе постановки и решения всякой прикладной проблемы реализует алгоритмы взаимодействия с пользователем. Функционирование моделирующей системы при решении всякой прикладной проблемы состоит в следующей последовательности работ: под управлением РС установка структуры сети семантических модулей согласно структуре решаемой проблемы; под управлением РС загрузка баз знаний (концептуальных моделей теорий) в ПБЗ, в ПД — операционных программ (операциональных определений эмпирических моделей теорий) и в соответствующие ОЗУ — баз данных (остенсивных определений предметов онтологической картины мира); решение проблемы в процессе взаимодействия ПБЗ с ПД в каждом из семантических модулей и семантических модулей между собой.

В каждом семантическом модуле ПБЗ (во взаимодействии с ПД и с ПБЗ других семантических модулей) интерпретирует концептуальную модель, а каждый ПД выполняет семантические программы эмпирической модели некоторой теории.

Обе структуры (рис. 1, 2) представляют структуру КС в развернутой в пространстве форме. Функционирование этой системы возможно и в КС минимальной конфигурации (рис. 3), содержащей, кроме управляющего РС, сеть в свернутой форме — из одного семантического модуля, содержащего ПБЗ, взаимосвязанный с РС, выполняющим функции ПД [19].

4.2. Системное программное обеспечение моделирующей КС должно выполнять в полном объеме модели всех функций научных теорий, лишь частично реализованных в современных КС, СОЗ, экспертных и интеллектуальных программных системах. В частности, дескриптивная функция научных теорий в современных КС реализована в форме языков программирования (высокого и сверхвысокого уровня — ЛИСП, Пролог, объектно-ориентированных, производственных и др.) с трансляторами полного описания задачи в машинные исполнимые модели, преспиктивная — как интерпретация машинных моделей решения проблем.

Для осуществления в полном объеме дескриптивной функции требуются: универсальный метаязык, выразительные возможности которого необходимы и достаточны для описания произвольных концептуальных моделей, инкрементный транслятор метапрограмм, реализующий функцию накопления машинных исполнимых моделей, и система поддержки проектирования трансляторов (или интерпретаторов) в машинные исполнимые модели с разных языков прикладных теорий, вплоть до естественных языковых описаний проблем. В свою очередь, система поддержки проектирования трансляторов (или интерпретаторов) должна моделировать процесс интерактивного информационного взаимодействия, включая процедуры ввода, накопления, ретрансляции, редактирования описаний.

Для осуществления в полном объеме преспиктивной функции необходима аппаратно-микропрограммная реализация интерпретатора машинных моделей, семантически эквивалентных и адекватных текстовым.



Рис. 3. Минимальная структура моделирующей системы — машина баз знаний

Осуществление *аргументативных* функций в развитой форме требует:

- для обоснования (при накоплении знаний) всякого нового описания (определения), развивающего или модифицирующего концептуальную модель произвольной эмпирической теории, необходимо реализовать модель процедуры синтеза структуры дедуктивной связи терминов нового описания с терминами имеющегося описания и/или терминов имеющегося описания с терминами нового описания;
- для объяснения теоретических терминов концептуальной модели необходимо реализовать модель процедуры ретрансляции структуры дедуктивной связи данного термина с другими;
- для объяснения результата решения задачи необходимо реализовать модель процедуры демонстрации структуры дедуктивного вывода результата;
- для предсказания фактов, допустимых теорией, необходимо придать интерпретатору машинных моделей функции синтеза фактов в процессе дедуктивного вывода результата решения задачи;
- для понимания высказываний необходимо придать интерпретатору машинных моделей функции дедуктивного вывода высказываний из машинных моделей произвольных эмпирических теорий.

Осуществление функций становления и эволюции теорий необходимо требует разработки и реализации моделей всех процедур синтеза компонент теории (см. [15, 18]).

Все эти системные средства и модели должны быть объединены в интерактивную СОЗ разрабатываемого типа с построением моделей. Кроме того, эта СОЗ должна включать две дополнительные компоненты (их прототипы используют наиболее совершенные современные программные системы), существенно облегчающие процессы использования КС, — *трассировщика* процесса решения задач, необходимого для отладки и верификации знаний в процессе критики, попытках фальсификации теории, а также подсистему *обучения* технологии решения задач и проблем.

Следующим этапом развития этой СОЗ является ее дополнение системой автоматизации проектирования структуры и реализации эмпирических моделей теорий, т.е. специализированных ПД.

Процесс создания такой СОЗ — методом «раскрутки», т.е. начиная с реализации аппаратной компоненты моделирующей КС и метаязыковой модели инкрементного транслятора метапрограмм, реализующего функцию накопления метапрограмм. При этом наиболее эффективна реализация каждой из компонент СОЗ по типу структуры научной теории, а всей СОЗ — как структуры системы научных теорий.

5. Эффективность моделирующей КС

Применение моделирующей КС меняет характер взаимодействия человека с техникой. Система, владеющая знаниями в конкретной прикладной области, ее теорией, для решения всякой проблемы должна быть обеспечена только постановкой проблемы, а в случае неполноты исходной информации для получения дополнительной информации система сможет вести разумный диалог с людьми и/или с техническими системами.

Предоставление пользователю метаязыка для описания прикладных проблем и совершенной технологии постановки и решения задач обеспечивает предельно высокий рост надежности «мягкого» продукта и эффективности программирования. При аппаратной реализации ПБЗ пользователю в форме метаязыка предоставляются все возможности аппаратного уровня, т.е. обеспечиваются предельно достижимые возможности проблемной ориентации архитектуры КС и производительности решения задач. В этом случае предельно простыми и, следовательно, эффективными оказываются реализации операционной системы КС, транслятора метаязыка и прикладных систем.

Если оценивать экономический эффект, то прежде всего следует отметить эффект от изменения качества продукции, изготавливаемой с помощью новой техники. В качестве первичной продукции, создаваемой с помощью моделирующей КС, выступают результаты решения задач — новая информация, принимаемые решения. За счет повышения выразительных возможностей языка, большей естественности технологии решения задач обеспечивается более полная и точная постановка задач, что в сочетании с адекватностью архитектуры КС структуре научной теории повышает качество принимаемых решений, их близость к оптимальным. В свою очередь, улучшение качества принимаемых решений при их внедрении обеспечивает экономию общественных ресурсов, вид которых зависит от приложения, от решаемых задач. Хотя численное значение эффекта от изменения качества решений определить в общем виде невозможно, можно утверждать, что для многих приложений он может достигать значительной доли общего эффекта.

Социальный эффект образуется за счет изменения структуры работ в новой (I) информационной технологии в сравнении с традиционной (II) (рис. 4), где 1 — знания, накопленные в предшествующий период; 2 — новые знания; 3 — программирование; 4 — решение задачи; T — время.

Процесс решения задачи в новой технологии, в отличие от традиционной, опирается на накопленные знания и начинается с анализа (в диалоге человека с КС) прикладной области с целью уточнения,

модификации и пополнения имеющихся знаний (концептуальной модели) до нужного уровня и выделения множества эмпирических терминов, соответствующих прикладной проблеме. Следующий шаг состоит в программировании операциональных определений эмпирических терминов и выполняется теми же средствами и включает те же работы, что и традиционное программирование. Отличие лишь в том, что операциональные определения существенно меньше по объему и структурно элементарно просты в сравнении с обычными программами. Существенной особенностью процесса постановки и решения задач в новой технологии является, как правило, использование значительной доли ранее накопленного знания прикладной теории, которая постоянно растет по мере обучения КС с решением всякой новой проблемы.

Социальный эффект состоит из двух компонент, первая из которых обеспечена принципиально новым характером использования труда, вложенного в постановку и решение проблем с помощью новой техники. Современные компьютеры не допускают вторичного использования знаний, заложенных в прикладных программах для решения задач, постановка которых отличается от предусмотренной этими программами, и тем самым по существу сводит программы и труд, затраченный на их создание, к роли малоценных, быстро изнашивающихся средств труда. Новая же КС оснащается специальными средствами накопления знаний (инкрементным транслятором). Знания, введенные и верифицированные, могут использоваться многократно вне зависимости от постановки конкретных задач до того момента, пока не изменится реальный мир или наши представления о нем. Эти изменения потребуют от пользователя только корректировки, уточнения знаний прикладной теории, объем и сложность которых определяются исходными изменениями. Труд, затраченный на создание знаний, накапливается и переходит в категорию капитальных средств, а характер обновления труда совпадает

с характером обновления капитальных средств. Эта компонента социального эффекта может быть рассчитана как суммарная стоимость прошлого труда, вложенного в постановку и решение задач на новой технике, за вычетом стоимости труда, затраченного на корректировку, уточнение, дополнение базы знаний.

Вторая компонента социального эффекта определяется экономией общественного труда на постановку проблем на новой технике по сравнению с традиционной технологией программирования за счет адекватности процессу решения проблем людьми путем создания и использования научных теорий.

Выводы

Для решения выявленных проблем развития архитектуры современных компьютеров впервые предложено принять структуру СОЗ за образец архитектуры КС, а в качестве образца структуры СОЗ — идеал структуры научной теории, поскольку именно в ней явно и наиболее концентрированно выражена форма мышления людей при решении проблем.

Наиболее существенными признаками идеальной архитектуры КС, непротиворечиво сочетающимися универсальность со специализацией КС на решение произвольных проблем, являются: семантический метаязык представления знаний, системное программное обеспечение, выполняющее модели всех функций научных теорий, и структура моделирующей КС по типу КС с распределенными функциями.

Аппаратная реализация метаязыка, метаязыковая реализация инкрементного транслятора метапрограмм, других компонент системного программного обеспечения и прикладных систем обеспечивают предельно высокую эффективность применения моделирующей КС.

Для достижения глобальной цели — максимальной помощи в решении каждой из бесконечного

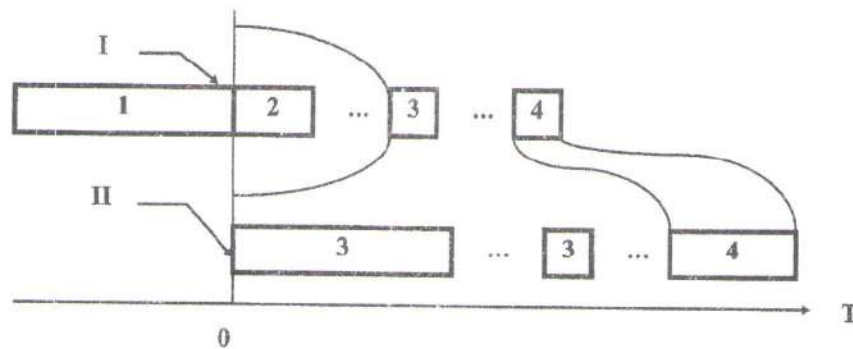


Рис. 4. Сравнение двух технологий работ

множества проблем эволюции общества — идеальная КС должна исполнять модели всех функций научных теорий, объединенные в концептуально единую систему.

Список литературы: 1. *Войшвилло Е.К., Дегтярев М.Г.* Логика как часть теории познания и научной методологии (фундаментальный курс). — М.: Наука, 1994. — Кн. II. — 332 с. 2. *Клини С.К.* Конечная аксиоматизируемость теорий в исчислении предикатов с помощью дополнительных предикатных символов // Математическая теория логического вывода / Сб. пер. под ред. А.В. Идельсона, Г.Е. Минца. — М.: Наука, 1967. — С. 237–284. 3. *Balzer W., Moulines C.U., Sneed J.D.* An architectonic for science: The structuralist program. — Dordrecht: Reidel, 1987. — 431 p. 4. *Мальцев А.И.* Алгоритмы и рекурсивные функции. — М.: Наука, 1986. — 368 с. 5. *Ершов А.П.* Интервью члена редакционной коллегии // Кибернетика. — 1981. — № 4. — С. 9–12. 6. *Turing A.M.* On computable numbers with an application to the Entscheidungsproblem. — Proc. London Math. Soc., Ser. 2, 1936. — 42. — P. 230–265. 7. *Амамия М., Танака Ю.* Архитектура ЭВМ и искусственный интеллект / Пер. с яп. — М.: Мир, 1993. — 398 с. 8. *Симонс Дж.* ЭВМ пятого поколения: компьютеры 90-х годов / Пер. с англ.; Под ред. Б.И. Шитикова. — М.: Финансы и статистика, 1985. — 174 с. 9. *Майерс Г.* Архитектура современных ЭВМ: В 2 кн. / Пер. с англ. — М.: Мир, 1985. — Кн. 1. —

364 с. 10. *Осуга С.* Обработка знаний / Пер. с яп. — М.: Мир, 1989. — 292 с. 11. *Палагин А.В., Кургаев А.Ф.* Проблемная ориентация в развитии компьютерных архитектур // Кибернетика и системный анализ. — 2003. — № 4. — С. 167–180. 12. *Ершов А.П.* Предварительные соображения о лексиконе программирования // Кибернетика и вычислительная техника / Под ред. В.А. Мельникова. — 1985. — Вып. 1. — С. 199–210. 13. *Кургаев А.Ф.* Анализ развития идеала структуры научной теории // Кибернетика и вычислительная техника. — 2003. — Вып. 139. — С. 50–63. 14. *Кургаев А.Ф.* Синтез-обоснование идеала структуры научных теорий: Препр. / НАН Украины. Ин-т кибернетики им. В.М. Глушкова; 2006-1. — К., 2006. — Ч. 1. — 45 с. 15. *Кургаев А.Ф.* Синтез-обоснование идеала структуры научных теорий: Препр. / НАН Украины. Ин-т кибернетики им. В.М. Глушкова; 2006-2. — К., 2006. — Ч. 2. — 24 с. 16. *Кургаев А.Ф.* Синтез-обоснование идеала структуры семантики научных теорий // Кибернетика и вычислительная техника. — 2005. — Вып. 147. — С. 22–32. 17. *Кургаев А.Ф.* Синтез-обоснование идеала структуры языка научной теории // Математичні машини і системи. — 2006. — № 1. — С. 99–112. 18. *Кургаев А.Ф.* Модели функций структуры научной теории // Электронное моделирование. — 2006. — 28, № 3. — С. 19–34. 19. *Кургаев А.Ф.* Исследование архитектуры машины баз знаний // УСИМ. — 2000. — №1. — С. 74–91.

Поступила в редколлегию 10.03.07

