

Символьное моделирование в человеко-машинной среде: основы концепции

В.Д. Ильин, И.А. Соколов

Аннотация. Изложены основы концепции символьного моделирования систем понятий, языков и построенных на их основе систем знаний. Определены классы базовых задач.

Введение

Символьные модели входят в обиход человека, как только он начинает познавать мир. Таблица умножения, записи шахматных партий, музыкальных композиций, алгебраических выражений, программы текстовых и графических редакторов, компьютерные игры и масса других символьных моделей для многих становятся привычными с детских лет. В наши дни профессиональная подготовленность к интеллектуальной деятельности в любой области предполагает некоторые навыки в символьном моделировании. Они необходимы, чтобы записывать уравнения математической физики или формулы химических реакций, разрабатывать специфицированные графические модели конструкций машин или других объектов и т.д.

Инструмент познания. Для производящей знания части человечества символьное моделирование давно стало инструментальной технологией оформления полученных результатов и обмена ими. Чем совершеннее технологии символьного моделирования, тем продуктивнее изобретательская деятельность. Методы построения, сохранения и передачи символьных моделей систем знаний вызывают неуклонно растущий интерес, так как люди ежедневно убеждаются в преимуществах, которые дают способности видеть актуальные задачи, формулировать их и находить методы решения. Именно этот интерес привел к идее применения

машин, помогающих решать задачи символьного моделирования.

Совершенствование инструментов познания неразрывно связано с совершенствованием символьных моделей систем понятий, языков сообщений и систем знаний. На пути от графических моделей языков, рукописных текстов, рисунков и схем к книгопечатанию и графическим моделям типовых решений в проектировании; от звукозаписи, фотографии и радио к кино и телевидению, от первых компьютеров к Интернету, виртуальным лабораториям и дистанционному обучению постоянно растет значение символьных моделей систем понятий, которые человек создает и использует с помощью машин. Трудно найти область деятельности, где бы в наши дни не применялось символьное моделирование в человеко-машинной среде. Среди них: Интернет с его сервисами (электронной почтой, веб-сервисами, поиском и др.); системы автоматизированного проектирования и множество других, включая системы для игры в шахматы, лучшие из которых на равных соперничают с чемпионами мира. Примерами символьных моделей в человеко-машинной среде могут служить компьютерные программы, системы команд микропроцессоров, сетевые протоколы и множество других. Проще сказать: все, с чем имеет дело человек за компьютером – это символьные модели, реализованные в человеко-машинной среде.

На современном этапе в центре внимания – цифровое символьное моделирование. «Цифровое» – потому, что символы любого типа и построенные из них модели представлены в виде цифровых кодов, рассчитанных на манипулирование посредством электронных машин. В таких машинах основу аппаратной составляющей представляют микросхемы, а цифровые коды реализуются такими схемами в виде двоичных кодов (отсюда и слова «электронный» и «цифровой» в названиях понятий электронный документ, электронная почта, электронные деньги, цифровая технология и др.).

Конкурентоспособная часть человечества, производя и применяя электронные документы, стремится делать это с помощью машин для символьного моделирования (серверов, рабочих станций, коммутаторов и др.) При этом для изготовления программ, книг, аудио- и видеоклипов, веб-сайтов, проектной документации и других символьных моделей используются текстовые, графические и другие редакторы. Совершенствующееся множество редакторов различного назначения – современный инструментальный для конструирования электронных символьных моделей. Интерпретация таких моделей, их распознавание, преобразование, сохранение, накопление, поиск, обмен, конструирование и защита – все эти задачи сегодня решаются с помощью программ, выполняемых программируемыми машинами. А сами программы представляют собой символьные модели технологий решения задач [3] в человеко-машинной среде.

Как инструмент познания символьное моделирование с помощью программируемых машин привлекло внимание исследователей, прежде всего, своей высокой продуктивностью, не зависящей от природы моделируемых объектов. На определенном этапе стало ясно, что новая реальность, созданная людьми, заслуживает внимания организационно оформленных коллективов исследователей. Для краткости такое моделирование назвали s-моделированием, среду его реализации – s-средой, получаемые модели – s-моделями, а машины для построения и применения символьных моделей (компьютеры, цифровые фото- и видеокамеры и др. устройства s-среды) – s-машинами [1].

Главной особенностью s-моделирования является то, что оно осуществляется с помощью s-машин, помогающих людям в построении и применении символьных моделей. Заметим, что и до того машины (печатные станки, типографские машины и др.) использовались людьми в символьном моделировании. Но это не была помощь в изобретении и конструировании символьных моделей. Такая помощь стала реальностью, когда появились программируемые машины со сменяемыми программами, хранимыми во внешней памяти таких машин (компьютеры).

Предмет исследований. Какими свойствами обладает s-моделирование? Какие закономерности ему присущи? Как его типизировать? Что представляют собой s-модели систем понятий, языков и систем знаний, каковы их свойства, как строить такие модели и манипулировать ими, используя s-машины? От ответов на такие вопросы зависят ответы на более конкретные вопросы. Какими должны быть правила конструирования s-моделей с помощью s-машин? Какой должна быть s-среда? Из поиска ответов на эти и связанные с ними вопросы состоит процесс исследования s-моделирования. Какие задачи s-моделирования актуальны на современном этапе и как их решить? Какой должна стать s-среда, чтобы соответствовать требованиям к реализации задач s-моделирования ближайшего будущего? На разных этапах развития информатики ответы на поставленные вопросы будут отличаться.

Одной из составляющих ядра концепции s-моделирования является утверждение о том, что основанием для решения задач, изучаемых информатикой, служат классы базовых задач s-моделирования [2]. Неубывающая актуальность выявления и уточнения классов базовых задач, создания методологии их решения вряд ли нуждается в пространном объяснении. В любой области деятельности исследователи и изобретатели постоянно совершенствуют инструментальный, пользуясь которым можно было бы изучаемые задачи представлять в виде некоторых конструкций из базовых задач. Применительно к автоматизированному конструированию программ такой подход был предложен и разработан в [3].

В данной статье, тематически продолжающей [1-2], изложены основы концепции s-моделирования. Приведены определения основных понятий, характеристика комплексного процесса s-моделирования как объекта исследований и описание классов базовых задач, рассматриваемых в качестве основания для решения прикладных задач информатики. Уточнен ряд определений, предложенных в [1-2].

1. Средства представления материала: о комплексе Tsm

В статье используется комплекс *Tsm* средств, разработанный на основе языка спецификации задачных конструктивных объектов, предложенного в [3] и позднее усовершенствованного. Этот комплекс применяется также для оформления моделей, входящих в состав создаваемой в ИПИ РАН системы *Sinf* знаний информатики [2].

Выделения. Для выделения определений, замечаний, примеров, имен понятий и отдельных частей текста используются следующие средства:

◀ <текст> ▶ ≈ часть текста с фиксированными в ее пределах обозначениями (здесь и далее символ ≈ заменяет слово «означает»);

□ <текст> □ ≈ утверждение (определение, аксиома, теорема, следствие и др.);

◇ <текст> ◇ ≈ замечание;

☼ <текст> ☼ ≈ пример;

{S<текст><список>S} ≈ здесь <текст> ≈ набранный курсивом текст (может быть пустым), который следует интерпретировать как расширенный префикс s-<текст> для выделенных курсивом элементов списка;

☼ {S модель<список>S} – здесь расширенным префиксом служит s-модель; {S<список>S} – здесь префикс s-☼.

Сокращения. Введены сокращения для часто повторяющихся названий понятий:

СМ ≈ символьное моделирование;

s-моделирование ≈ СМ произвольных объектов в человеко-машинной среде;

s-машина ≈ машина, помогающая создавать и применять s-модели;

s-среда ≈ совокупность взаимодействующих людей и управляемых ими s-машин, предназначенная для решения задач s-моделирования.

Запись формул. Для теоретико-множественных и других формул применяется «одноэтажная» форма записи: $a: elem A \approx a$ является элементом множества A ; $A: set a \approx A$ – множество, содержащее элемент a ; $A < B$ (когда оговорено, что A и B рассматриваются как множества) $\approx A$ – подмножество B ; $B = D \approx$ множества D и B совпадают; $C \leq B \approx C$ является подмножеством B или совпадает с ним; $B > A \approx B$ содержит A ; $A \geq E \approx A$ содержит E или совпадает с E ; $A \wedge B \approx$ пересечение множеств A и B ; $A \setminus B \approx$ разность множеств A и B ; $A * B \approx$ декартово произведение множеств A и B ; $R \leq A * B \approx$ бинарное отношение, заданное на множествах A и B . Символ \emptyset обозначает пустое множество или нуль (в зависимости от контекста); символ $\#$ обозначает «не равно». ☼ Если $x: elem X$, $y: elem Y$ и $x = y$, то $x: elem(X \wedge Y)$; если $X: set x$, $Y: set y$ и пара $(x, y): elem R$, где $R < A * B$, то $(X * Y) \wedge (A * B) \# \emptyset$ ☼.

При записи операций символы «+», «-», «*», «/» обозначают соответственно сложение, вычитание, умножение, деление, а символ «**» – возведение в степень. Аргументы функции размещаются в круглых скобках, стоящих сразу за идентификатором, обозначающим функцию: $f(x) \approx f$ от x ; $f[max;] (x[i=1...n]) \approx f$ с верхней пометой max от $x[i=1...n]$. Для записи суммы вместо заглавной «сигмы» используется sum ; при этом индекс суммирования, его начальное и конечное значения записываются в квадратных скобках справа от sum : $sum[i=1...n] \approx$ сумма по i от 1 до n .

В *Tsm* не накладывается никаких ограничений на максимальное число индексов для переменных и помечающих символов (помет). Все индексы и пометы записываются в строчку внутри квадратных скобок, следующих сразу за индексируемой (или / и помеченной) переменной. В *Tsm* квадратные скобки зарезервированы только для индексов и помет. Индексы, определяющие элемент массива, отделяются запятыми, индексированные индексы – косой чертой «/». Верхний индекс от нижнего отделяется точкой с запятой «;». Если в описании индекса точка с запятой не встречается, то индекс считается нижним. ☼ $x[out; j = 1...n] \approx$ вектор из n компонент, имеющий помету out ; $a[inp; i = 1...m, j = 1...n] \approx$

матрица размера $m * n$, имеющая помету $inp; c[';1] \approx$ с-один со штрихом (штрих «'» – верхняя помета, 1 – нижний индекс); $d[j/i;] \approx d$ с верхним индексированным индексом j i -тое (чтобы показать отсутствие нижних индексов, поставлена точка с запятой, за которой сразу следует закрывающая квадратная скобка; $d[j/i]$ $\approx d$ с нижним индексированным индексом j i -тое (отсутствие точки с запятой указывает на отсутствие верхних индексов) ☼.

2. S-моделирование: основные понятия

Изучить (понять) – значит описать характеристики исследуемого объекта и связи с другими, ранее изученными объектами. Результат изучения любого объекта представлен в виде некоторых изменений систем понятий, существующих в памяти исследователя. Чтобы измененные системы понятий сделать доступными другим людям, необходимо представить их в символической форме, удовлетворяющей ряду необходимых требований [1].

Итак, существует цепочка: *изучение производных объектов* → *формирование систем понятий в памяти исследователя* → *построение символических моделей систем понятий и сохранение их во внешней среде*.

Из этого следует, что ☐ *Объектами символического моделирования всегда являются системы понятий, отражающие результаты изучения различных объектов* ☐. Символические модели систем понятий не только пополняют арсенал средств, используемых для изучения различных объектов. Они сами и процесс их символического моделирования также служат объектами исследований.

Чтобы обмениваться символическими моделями систем понятий, необходимы средства формирования и передачи сообщений. Исследование языков сообщений предполагает изучение типов и правил построения сообщений, систем метапонятий, на которых интерпретируются сообщения, и механизмов интерпретации [2]. Стихийно складывающиеся языки (например, разговорные) получили название естественных. Неопределенность и слабая управляемость развитием таких языков служат препятствием к их неадаптированному применению в s-среде.

Языки сообщений в s-среде целесообразно строить в соответствии с четко определенными спецификациями [2].

☐ *Язык сообщений в s-среде* определен, если специфицированы:

- система символов;
- правила построения сообщений;
- система метапонятий для интерпретации сообщений;
- механизм интерпретации ☐.

2.1. Символьное моделирование с помощью машин

С развитием техники не могла не родиться идея конструирования символических моделей с помощью машин. С ростом значения символических моделей в исследованиях и проектировании эта идея становилась все более мощным стимулом для изобретателей машин, помогающих решать задачи. Было предложено и построено немало машин такого назначения. Особое место среди них принадлежит программируемым машинам со сменяемыми программами. По мере увеличения производительности s-машин этого типа возрастал интерес к их применению. Настало время, когда в математике сформировался и стал интенсивно развиваться раздел вычислительной (машинной) математики. Изобретатели s-машин различных архитектур и программисты начального этапа s-моделирования, физики и инженеры, создававшие элементную базу, проектировавшие комплектующие изделия и периферийные устройства для первых s-машин – все они результатами доказали целесообразность серийного производства s-машин со сменяемыми программами, хранимыми в памяти этих машин.

Особое место в развитии символического моделирования принадлежит идее его формализации, заключающейся в том, чтобы строить символические модели по определенным правилам из заранее определенных элементов. Эта идея реализована в математических методах символического моделирования. Однако метод формализации, применяемый в математике для получения формальных систем, нельзя перенести на s-моделирование, так как s-модели не являются формальными системами. Объясним подробнее это важное замечание.

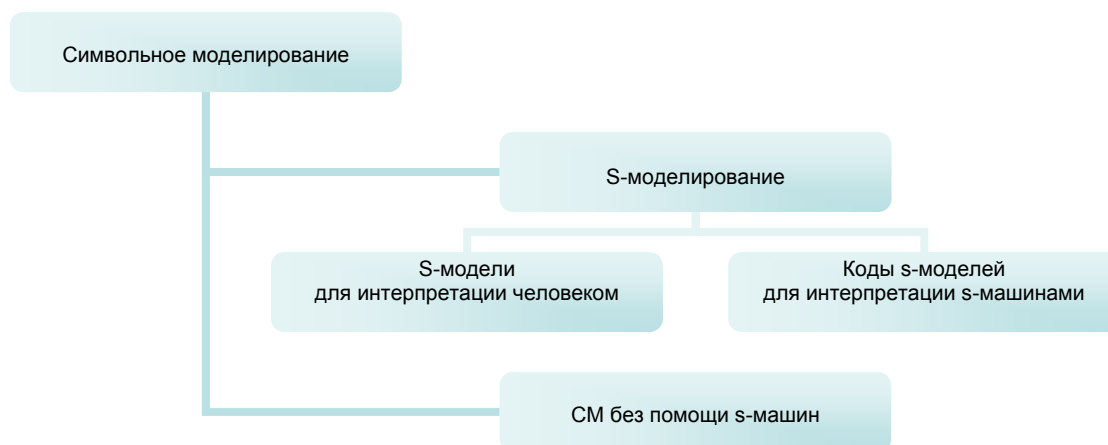


Рис. 1. S-моделирование: специализация символьного моделирования

В s-моделировании *задача* [3] имеет более широкий смысл, чем в математике. Рассматриваемые далее базовые задачи s-(представления, распознавания, преобразования, конструирования, интерпретации, обмена, сохранения, накопления, поиска, защиты) не являются математическими, хотя для их решения применяются и математические методы. Но математический арсенал недостаточен для того, чтобы каждую из указанных задач можно было сформулировать и решить как математическую задачу. Дело здесь не только в том, что в классической математике царствует формальное доказательство (существования, единственности решения), а в s-моделировании – конструктивное доказательство (существования s-модели; а о единственности вообще речь не идет) [2]. Важно другое: неформальность s-моделей – это их полезное отличие от формальных математических моделей. Именно оно позволяет привлечь знания человека-эксперта для управления ходом решения. Примером может служить разработанная в ИПИ РАН методология интерактивного преобразования ресурсов по правилам, изменяемым экспертом в ходе решения [4].

Символьное моделирование без помощи s-машин предполагает представление символов и символьных моделей в формах, рассчитанных только на восприятие людьми, а s-моделирование – в формах двух видов (Рис. 1). Один из них рассчитан на интерпретацию человеком, а другой – программами s-машин. В s-моделировании множество символов, применимых для

построения s-моделей, изучается как множество элементарных конструктивных объектов, каждый из которых наделен набором атрибутов и совокупностью допустимых операций. Построение конструкций из элементов этого множества определено системой правил конструирования s-моделей.

2.2. Символы

□ *Символ* в s-моделировании – это заменитель некоторого объекта, представленный в двух формах, одна из которых рассчитана на интерпретацию человеком, другая (в форме кода символа) – программой s-машины. Изучаемые в информатике символы и символьные конструкции – это воспроизводимые в s-среде средства представления систем понятий, отражающих произвольные объекты изучения. Они должны обладать набором свойств, обеспечивающих возможности:

- применения без участия изобретателя;
- неограниченного числа воспроизведений по заданным правилам;
- распознавания и преобразования;
- построения на их основе символьных конструкций;
- сохранения, накопления, копирования и удаления;
- поиска и перемещения в s-среде □.

◇ Оригинальный символ, хранящийся только в памяти человека, который его изобрел, приобретает шанс стать воспроизводимым, когда он представлен в s-среде. Это означает, что новый символ описан его изобретателем с исполь-

зованием других символов, считающихся известными. Другими словами, воспроизводимость предполагает обязательную опору на уже известные символы и средства их изготовления и применения \diamond .

По своей природе символы, осознанно используемые человеком, дискретны. Дискретность символов – синоним их различимости. Она так же важна для человека, как для s-машин важна различимость кодов символов. Вывод о дискретной природе символов вполне естественно привел к идее их представления с помощью устройств, в основе которых лежит принцип переключения из одного устойчивого состояния в другое (на множестве известных состояний).

Типы символов. \square *Типу* $ts[A]$ *символов* соответствует заданное множество A символов, для которых определены набор атрибутов $at[A]$ и семейство $op[A]$ допустимых операций \square .

\square *Специализации типа* $ts[B]$ *символов* соответствует некоторое подмножество $B \subset A$ символов с набором $at[B]$ ($at[B] > at[A]$) атрибутов и семейством $op[B]$ ($op[B] > op[A]$) допустимых операций. *Тип* $ts[A]$ будем называть *обобщением типа* $ts[B]$ \square .

К базовым типам символов относятся (Рис. 2):

- аудио,
- графический,
- механический.

Тип *речевой* – является специализацией типа *аудио*. Типы *текстовый*, *видео*, *анимационный* и *гипертекстовый* – это специализации типа *графический*, а тип *жестовый* – специализация типа *механический*. Тип *числовой*¹ является специализацией типа *текстовый*.

Поясним подробнее введенный нами тип *механический*, который используется для построения сообщений в виде механических воздействий. Для ввода сообщений, адресованных программам s-машин, человек использует клавиатуры, мыши, джойстики, стилусы и другие устройства, реагирующие на механические воздействия (нажатия, перемещения и др.). Вывод

механических сообщений из s-машин реализуется в форме вибровывода мобильных телефонов, силовой обратной связи (force feedback) игровых панелей и др. \odot Нажатие на клавишу с изображением буквы – это преобразование типа *механический* в тип – *текстовый*. При вводе, не глядя на клавиатуру, человек использует систему заученных жестов, привязанных к конкретному устройству \odot .

Композиции типов символов. Композицией типов символов будем называть любое сочетание из числа возможных сочетаний различных типов. Вряд ли нуждается в пояснениях стремление использовать в сообщениях все реализуемые в s-среде типы символов. Достаточно вспомнить, какую роль сыграла мультимедийная форма представления документов, которая с добавлением видео затем стала гипермедийной.

2.3. Коды

\square *Код в s-моделировании* – рассчитанный на интерпретацию программами s-машин заместитель символа или s-модели. Удовлетворяет требованиям применения при решении базовых задач *преобразования, распознавания, конструирования, интерпретации, обмена, сохранения, накопления, поиска и защиты* в s-среде \square .

Постановку числа в соответствие символу называют цифровым кодированием, а его результат – цифровым представлением символа. Для чисел, заменяющих символы, при проектировании s-машин выбирают систему счисления и число разрядов для представления чисел в памяти s-машин. Этот выбор направляется стремлением обеспечить наиболее эффективное манипулирование кодами символов и s-моделей, которое выполняют s-машины при решении различных задач. Выбор ограничен рядом условий, среди которых физико-техническая реализуемость используемых для построения s-машин элементов с количеством устойчивых состояний, равным основанию выбранной системы счисления. Программы и данные в современных s-машинах представлены в виде двоичных кодов. Преобразования из аналоговой формы представления символов в цифровую (АЦП) при вводе и обратное преобразование (ЦАП) при выводе необходимы для

¹ В семейство операций типа *числовой* входят арифметические операции, которых нет в семействе операций типа *текстовый*.

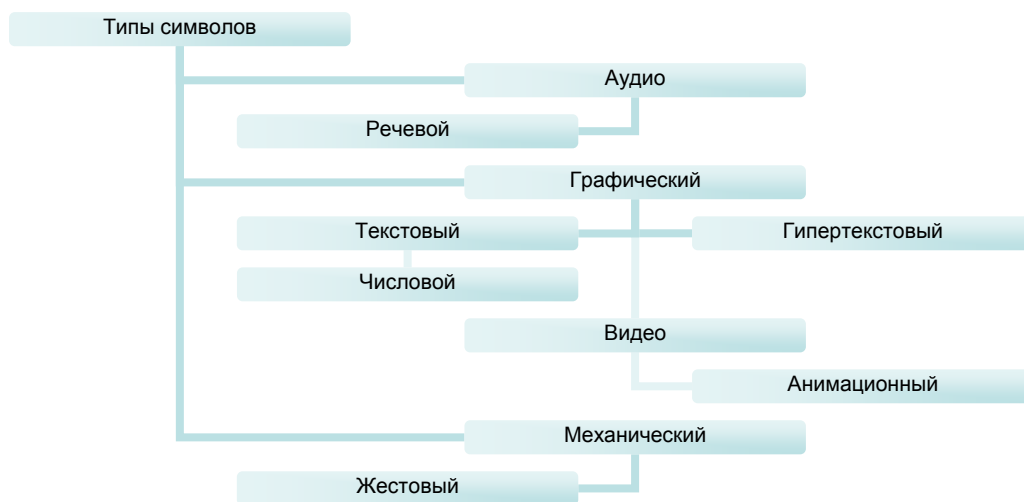


Рис. 2. Типы символов в s-моделировании

связи s-машин с их аналоговым окружением в s-среде, к которому относятся и люди, управляющие s-машинами.

3. Классы базовых задач

Изучение свойств и закономерностей s-моделирования необходимо, чтобы определить, из каких типовых задач оно складывается. Говоря здесь о задачах, имеем в виду задачи исследований. Будем называть их *базовыми задачами s-моделирования*. Деление базовых задач на классы – результат изучения s-моделирования как комплексной технологии представления, преобразования, распознавания, конструирования, интерпретации, обмена, сохранения, накопления, поиска и защиты символьных моделей в человеко-машинной среде.

1. *Представление s-моделей на языках сообщений*, рассчитанных на восприятие человеком и машинами, предполагает существование средств формирования сообщений, включая языки. В этом классе задач изучаются базовые типы символов и кодов, на основе которых строятся системы символов и кодов, используемые соответственно в человеко- и машинно-ориентированных языках. К первым относим языки спецификации, программирования, запросов, ко вторым – системы команд. Этот класс включает задачи представления данных и документов. В него входят задачи представления моделей систем понятий, на которых

интерпретируются сообщения, составленные на языках. На верхнем уровне задачной иерархии этого класса находится представление моделей систем знаний.

2. *Преобразование типов и форм представления s-моделей* позволяет устанавливать соответствия между моделями. Задачи преобразования типов (аудио в текст и обратно и др.) и форм (аналоговой в цифровую и обратно; несжатой в сжатую и обратно; незашифрованной в зашифрованную и обратно; несжатой и незашифрованной в сжатую и зашифрованную и обратно; одной формы представления документа в другую: *.doc в *.pdf) – необходимое дополнение к задачам представления моделей.

3. *Распознавание s-моделей*. Сообщение не может быть интерпретировано, если оно не распознано получателем. Необходимым, но не достаточным условием распознавания является представление сообщения в формате, известном получателю. При выполнении этого условия для распознавания сообщения решаются задачи сопоставления с моделями-образцами, либо сопоставления свойств распознаваемой модели со свойствами моделей-образцов.

4. *Конструирование s-моделей* – это их построение по определенным правилам из ранее созданных моделей. К задачам этого класса относятся задачи конструирования моделей систем понятий, языков, систем знаний, интерпретаторов сообщений на моделях систем понятий; моделей задач [3], программирования, взаимо-

действия в *s*-среде; моделей архитектур машин, сетей, сервис-ориентированных архитектур; моделей сообщений и средств их построения, документов и документооборота. На верхнем уровне иерархии этого класса находятся задачи конструирования моделей *s*-среды и технологий *s*-моделирования.

5. *Интерпретация s-моделей* сообщений предполагает существование принятого сообщения, модели системы понятий, на которой оно должно интерпретироваться и механизма интерпретации. В этом классе изучаются различные модели трансляции (компиляция, интерпретация, ассемблирование).

6. *Обмен s-моделями*. В этом классе изучаются задачи взаимодействия в *s*-среде (человек – машина; машина – машина) с типизацией: отправителей и получателей; средств отправки, передачи и получения сообщений; сред передачи сообщений. Создаются *s*-модели систем правил обмена сообщениями, архитектур сетей, сервис-ориентированных архитектур, систем документооборота.

7. *Сохранение, накопление и поиск s-моделей*. Этот класс включает связанные между собой задачи сохранения, накопления и поиска. Изучаются и типизируются память и накопители; механизмы управления памятью и накопителями; формы сохранения и накопления; носители, механизмы сохранения, накопления и поиска; базы данных и библиотеки программ. Изучаются модели предмета поиска (по образцу, по признакам, по описанию свойств) и механизмов поиска.

8. *Защита s-моделей от несанкционированного доступа и применения*. Этот класс включает задачи предотвращения и обнаружения уязвимостей, контроля доступа; задачи защиты от вторжений, вредоносных программ, перехвата сообщений, несанкционированного применения.

4. S-модели

Система понятий. *S*-модель *sc* системы *sC* понятий определяется парой <память *mem[sc]* *s*-модели *sc*>, <семейство *rel(mem[sc])* связей, заданных на *mem[sc]*> (где [*sc*] – помета). Здесь *mem[sc]* и *rel(mem[sc])* соответствуют совокупности понятий моделируемой системы *sC* и се-

мейству связей, заданных на этой совокупности. Основным типом связей являются задачные связи по памяти, поскольку прагматика *s*-моделирования связана с решением различных задач в *s*-среде.

Система понятий может рассматриваться как некоторое метапонятие. Чтобы определить систему метапонятий, также необходимо представить ее *s*-модель и описание области применимости [1]. Подобное движение вверх логически не ограничено. То же можно заметить и относительно движения вниз: не существует никаких ограничений на детализирующие определения понятийных систем путем построения их *s*-моделей.

Система понятий с ядром <задача> является одним из центральных объектов *s*-моделирования, поскольку любая программа, которую выполняет *s*-машина – это *s*-модель алгоритма решения некоторой задачи.

Задачная область. *S*-модель *pa* задачной области *pA* определяется парой <память *mem[pa]* задачной области>, <семейство *rel(mem[pa])* связей, заданных на *mem[pa]*>. Непустое множество *mem[pa]* элементов памяти разбито на три подмножества: входов *inp[pa]* задач, выходов *out[pa]* задач и подмножества *or[pa]*, каждый из элементов которого является и входом и выходом некоторых задач. Любое одно из этих подмножеств может быть пустым; могут быть одновременно пустыми *inp[pa]* и *out[pa]*.

Система знаний. *S*-модель *kn* системы знаний *Kn* определяется триадой <модель *ca* системы *Sa* метапонятий>, <совокупность *set[lng]* языков сообщений>, <совокупность *set[intr]* интерпретаторов на модели *ca* сообщений, составленных на языках из *set[lng]*>. Интерпретация сообщения на модели *ca* системы *Sa* метапонятий – это построение выходного сообщения (извлечение информации) по заданному входному.

Необходимым условием построения *s*-модели языка сообщений является существование *s*-моделей системы метапонятий, на которой предполагается интерпретировать сообщения, составленные на языке, и базовых типов символов, композиции которых предполагается использовать для построения системы символов языка. Эти модели играют роль исходных для по-

строения языка. Построение s-модели языка сообщений – это разработка моделей: композиции базовых типов символов; системы символов языка, построенной на основе модели композиции базовых типов символов; системы правил конструирования сообщений с использованием модели системы символов.

Необходимым условием построения s-интерпретатора сообщений является существование s-моделей входного и выходного языков, а также – системы метапонятий, на которой должны интерпретироваться сообщения, составленные на входном языке. Построение s-модели интерпретатора – это разработка моделей: распознавания сообщений на принадлежность входному языку; интерпретации распознанных сообщений на модели системы метапонятий; представления результата интерпретации в виде сообщения на выходном языке.

5. Общий метод и научная продукция

Общий метод s-моделирования – конструктивное доказательство существования s-моделей, представимых в двух формах, одна из которых рассчитана на интерпретацию человеком, а другая – s-машиной. Необходимое условие реализации s-моделирования предполагает существование удовлетворяющих требованиям s-(представления, распознавания, преобразования, конструирования, интерпретации, обмена, сохранения, накопления, поиска и защиты):

1. языка описания s-моделей, рассчитанного на человека;
2. языка команд s-машины;
3. программ преобразования s-моделей на языке для человека в описания на языке команд s-машины.

Научная продукция информатики – методы решения задач s-(представления, распознавания, преобразования, конструирования, интерпретации, обмена, сохранения, накопления, поиска, защиты).

Эти методы применяются при разработке информационных технологий и построении s-среды (в том числе: конкретных архитектур компьютеров и сетей; систем автоматизации проектирования программных и аппаратных средств; интерфейсов (пользователя, программных и аппаратных); систем управления базами данных и

систем искусственного интеллекта; сервис-ориентированных архитектур и др.).

Заключение

В статье, являющейся тематическим продолжением [1-2], изложены основы концепции s-моделирования, ядро которой составляют следующие утверждения.

1. Результаты исследования закономерностей s-моделирования служат методологическим основанием для разработки технологий решения задач с помощью s-машин.

2. Изучение любых объектов сопровождается построением систем понятий в памяти человека. Чтобы обмениваться добытыми результатами, системы понятий необходимо представить в форме символьных моделей, рассчитанных на сохранение и передачу во внешней среде.

3. Обмен символьными моделями предполагает существование освоенных участниками обмена средств формирования сообщений, систем метапонятий, на которых интерпретируются сообщения, и механизмов интерпретации [2].

4. Среди средств формирования сообщений особое место принадлежит формализованным языкам (включая языки спецификаций задач [3] и программирования поведения машин). Такие языки используются для записи формализованных утверждений о понятиях и связях между ними.

5. Утверждения, состоящие из спецификаций понятий и связей между ними, составляют основание определений систем понятий в s-среде, каждое из которых должно сопровождаться спецификацией области применимости [1-2].

6. Особое значение имеют разрешимые связи между понятиями, представимые в виде спецификаций программируемых задач [3], решаемых в s-среде с помощью s-машин.

7. Классы базовых s-задач – это отображение s-моделирования в пространство типовых задач, решаемых в s-среде.

8. Повышение продуктивности разработки прикладных сервисов и программных средств достигается путем их автоматизированного конструирования на основе базовых задач s-моделирования.

Литература

1. В.Д. Ильин, И.А. Соколов. Информация как результат интерпретации сообщений на символьных моделях систем понятий. Информационные технологии и вычислительные системы, 4, 2006, с. 74-82
2. В.Д. Ильин, И.А. Соколов. Символьная модель системы знаний информатики в человеко-автоматной среде. Информатика и ее применения, 2007. Т. 1. Вып. 1. с. 6-78
3. Ильин В.Д. Система порождения программ. М.: Наука, 1989. с. 264
4. А.В. Ильин, В.Д. Ильин. Интерактивный преобразователь ресурсов с изменяемыми правилами поведения. Информационные технологии и вычислительные системы, 2, 2004, с. 67-77.

Ильин Владимир Дмитриевич. Заведующий лабораторией института проблем информатики РАН. Окончил Московский энергетический институт в 1960 году, доктор технических наук, профессор, автор более 80 публикаций. Область научных интересов – символьное моделирование, автоматизация разработки программ и сервисов.

Соколов Игорь Анатольевич. Директор Института проблем информатики РАН. Окончил Московский государственный университет им. М.В. Ломоносова в 1976 году, доктор технических наук, профессор, член-корреспондент РАН, автор более 90 публикаций. Область научных интересов – информационные технологии и автоматизация.