

ИСПОЛЬЗОВАНИЕ СИСТЕМОЛОГИЧЕСКОГО ПОДХОДА ПРИ РАЗРАБОТКЕ СЛОЖНЫХ ПРОГРАММНЫХ СИСТЕМ

*МАТОРИН С.И., КУЛИБАБА В.В.,
СОЛОВЬЕВА Е.А.*

Предлагаются классификации информационных связей и функций модификации информации, используемые для описания информационных процессов в программных системах. Модель знаний, основанная на данных классификациях, может быть использована при разработке автоматизированных интеллектуальных информационных систем, систем имитационного моделирования, стандартизации описания программных компонент и т.п.

1. Введение

Сложность создания больших программных систем, их недостаточная эффективность и надежность требуют разработки новых технологий, инструментариев моделирования и усовершенствования существующих методологий [1, 2]. Использование объектно-ориентированной методологии позволяет повысить эффективность и надежность систем, снизить возникающую сложность при разработке, но по сравнению с системным анализом не позволяет добиться высоких результатов, так как применяемые в данной методологии методы зависят от уровня знаний и взглядов специалиста [2, 3]. Эффективность используемых в объектно-ориентированном моделировании методов структурного анализа также ставится под сомнение, поскольку:

– в объектно-ориентированной методологии имеет место нестыковка объектных и структурных методов анализа и проектирования сложных систем;

– известные объектные методы разделяют статические и динамические аспекты проектирования программной системы (например, в UML имеются различные диаграммы спецификации статики и динамики, между которыми нет строго формального перехода)[4,5].

Поэтому повышение качества создаваемых моделей с помощью объектно-ориентированной методологии требует изменения предлагаемых в рамках данной методологии процедур анализа и учета разработанных подходов в системном анализе, т.е. синтеза системной и объектно-ориентированной методологий. Данная проблема была решена в работах [4-6] на базе системологии. Разработанная новая системно-ориентированная (системологическая) УФО-технология была успешно применена в области моделирования организационных систем. Полученные результаты [4, 7] при моделировании организационных процессов позволяют утверж-

дать, что её использование при разработке информационных систем дает возможность:

– провести имитацию функционирования разрабатываемой системы непосредственно на её проектной модели;

– повысить уровень: автоматизации и объективности анализа; проектирования систем; качества тестирования.

Положительный результат использования данной технологии достигается за счет применения специальных знаний о предметной области и модели манипулирования ими.

Использование УФО-технологии при разработке сложных программных систем ограничено ввиду отсутствия в модели представления знаний информации о протекающих в данных системах информационных процессах. Поэтому актуальной практической задачей является необходимость специализации (адаптации) в исследуемой предметной области модели представления знаний (параметрическая классификация) [8].

2. Постановка задачи

Для адаптации к исследуемой предметной области модели представления знаний необходимо исследовать информационные связи (виды информации: программы, данные и т.д.) в программных системах и функции преобразования информации, построить их классификации. При разработке классификаций должны учитываться критерии естественной классификации[9]. Необходимость соблюдения данных требований заключается в научной ценности естественной классификации, в которой учитываются сущностные признаки классифицируемых объектов. Таким образом, разработка классификаций должна проводиться на основе использования системологического подхода и системологического классификационного анализа.

Руководствуясь принципом системологии – рассматривать любой объект (явление, процесс и т.п.) с учетом его меры системности как систему, «в действительности не существует ничего кроме систем, различающихся мерой (мерой системности)» [9, с. 36], проведем исследование информационных связей и функций преобразования информации.

3. Исследование информационных связей

В первую очередь необходимо рассмотреть форму представления связи, выбрав в качестве исследуемой системы – программу. Тривиально её можно представить как набор данных и алгоритмов их обработки, таким образом, можно предположить, что программа – информация о некоей задаче, используемая для ее решения. Информация, представленная в машинном виде, – данные, класс данных называется объектом данных [10-13]. Объект данных представляет собой контейнер (форма представления информации) для хранения значения данных (значение, несущее некую смысловую информацию). Среди множества объектов данных выделяют определяемые программистом (перемен-

ные, константы, массивы) и системой (создаваемые автоматически виртуальным компьютером во время выполнения программы: стеки, файловый буфер, списки свободной памяти и т.д.). Мы будем рассматривать объекты данных, определяемые программистом.

Их можно разделить на два типа, по возможности модификации во время выполнения программы: переменные и константы (значение изменить нельзя). Все множество существующих объектов данных делят на элементарные и сложные. Элементарный объект данных может содержать только одно значение, хранимое значение фигурирует как единое целое и характеризуется только одним атрибутом (тип значения); для класса таких объектов в языке программирования определены различные операции над ними. В свою очередь, класс элементарных объектов данных включает класс простых и комбинированных данных. Простой (или примитивный) — тип данных, для которого определено упорядоченное множество допустимых значений, с наименьшим и наибольшим элементами. Как правило, для данного типа набор операций является частью определения языка [10].

Представленная классификация объектов данных (рис. 1) разработана на основе анализа ряда императивных, логических и объектно-ориентированных языков программирования, в ней учтены результаты исследований, проводимых студентами группы ПОАС в данном направлении. Она содержит описание существующих объектов данных, скорректированное под текущее понимание объекта данных в языках программирования. Ввиду многообразия языков программирования, используемых типов данных и разработанных методов работы с ними [10–13], классификацию необходимо адаптировать (дополнить новыми понятиями) под используемый язык.

Составленная классификация дополнит модель знаний [6, 8], которая в дальнейшем будет использована при проектировании и разработке программных моделей сложных систем.

Далее необходимо рассмотреть способ задания назначения хранимой информации. Из выделенных объектов данных для простых объектов мы можем описать их содержание: числовой тип содержит числа, логический — значения «ложь» и «истина», символьный — символы алфавита, печатные символы и т.п. Форма объекта данных определяет множество значений, принимаемых этим объектом, т.е. содержащуюся в нем информацию. Для элементарных объектов содержание может быть задано множеством путем перечисления однотипных объектов, для сложных — множеством состояний его составных частей. Трудно указать назначение для хранимой в объектах данных информации. Интуитивно мы можем предположить, что численный объект данных используется для хранения числа — оно применяется для индексации и представления величины, размера, объема и т.д. [14]. Однако это не раскрывает полной сути назна-

чения хранимого значения. Необходимость обработки данного вопроса объясняется стремлением ввести единую понятийную модель представления знаний, которая позволит в дальнейшем рассматривать не только форму представления данных (раскрывает «способ общения» с компонентом), но и цель (предназначение используемого объекта данных, что существенно для понимания представленного знания, содержащегося в объекте данных).

Задание назначения хранимой информации соизмеримо с решением проблем представления знаний обо всей предметной области. Учитывая, что семантический подход к рассмотрению информации «является наиболее трудно формализуемым и до сих пор окончательно не определившимся» [15, с. 24], искомое решение должно быть основано на использовании знаний. Можно предложить следующие пути смыслового описания хранимой информации на основе существующих моделей представления знаний — использование семантических сетей (классификации, онтологии) [9, 16–20], фасетного подхода [21], фреймы, продукции и т.д. [16]. Применение для представления знаний классификации является предпочтительным решением данного вопроса, так как:

- классификационная модель представления знаний ближе к понятийной организации памяти человека [9, 16, 22];
- наработана большая теоретическая база по разработке классификаций (УДК, ББК, онтологии, естественная классификация) [9, 16–20];
- существуют уже готовые результаты и опыт практического использования в данной области [17–20, 23–25];
- классификации можно применять совместно, дополняя и расширяя представленную предметную область (многоаспектные классификации) [23, 24].

Таким образом, классификационный подход дает возможность детально представить знания о любой предметной области, является универсальным (использование классификации позволит формализовать представление знаний, ввести стандартизированное, не зависящее от языка, описание понятий), легко сопровождаемым и не требующим серьезных капиталовложений (возможность использования уже наработанной базы) [8, 9, 20].

4. Исследование функций преобразования информации

Определив множество объектов данных, необходимо задать множество производимых над ними операций (рис. 2), которые указывают на возможные манипуляции (действия) с объектами этого типа. Эти операции могут быть элементарными (заданными явно), т.е. являться частью определения языка, или они могут определяться программистом (заданными абстрактно) в виде подпрограмм или методов, как часть определения типа данных.

Примером некоторых операций, встроенных в языки программирования, могут служить «присваивание», «создание», «инициализация», «объявление» [10].

«Присваивание» – изменение связывания объекта данных со значением. В некоторых языках операция возвращает результат, содержащий копию присвоенного значения.

«Инициализация» – присваивание переменной первоначального значения. Неинициализированный объект данных – такой объект, который был создан, но ему еще не было присвоено значение. Он может содержать любое значение из области допустимых значений, определенных для типа объекта.

«Создание» – выделение блока памяти для объекта.

Операция «Объявление» – сообщает транслятору об именах и типах, используемых в программе объек-

тов данных. Объявления бывают явными и неявными.

Некоторые операции в языках программирования унифицированы до представления их в виде операторов: сложение, умножение, операторы цикла.

В широком понимании операция является математической функцией (может быть описана математической функцией): каждому допустимому входному значению, т.е. аргументу (или набору аргументов) она сопоставляет однозначно определенный результат. Операция определяется на множествах допустимых значений аргументов (область определения) и возможных значений результата выполнения операции (область значений). Действие операции определяет, каковы будут результаты для данного набора входных значений (аргументов).

На множестве операций выделяют унарные (операции с одним аргументом и результатом) и бинарные

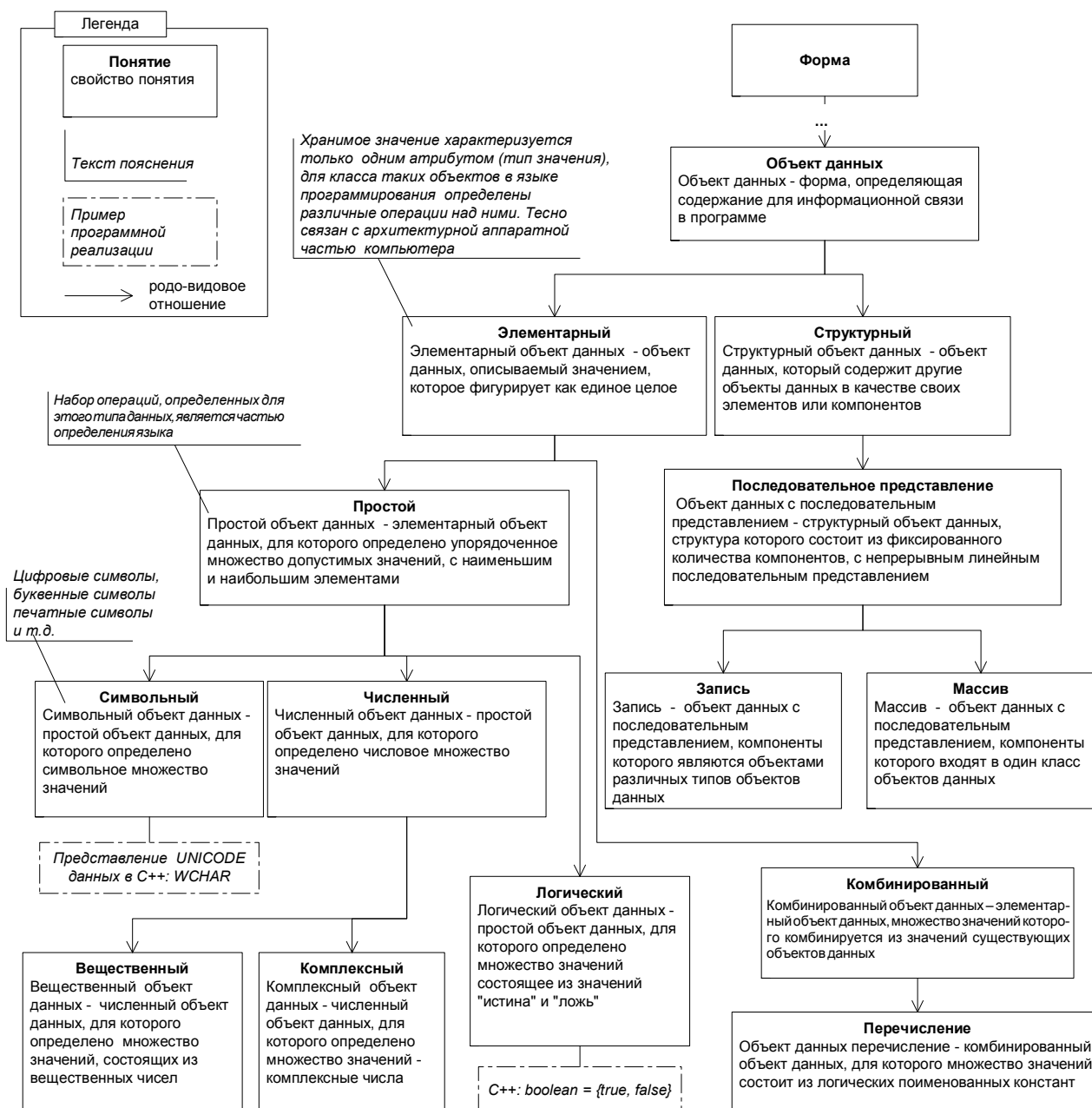


Рис. 1. Фрагмент классификации объектов данных

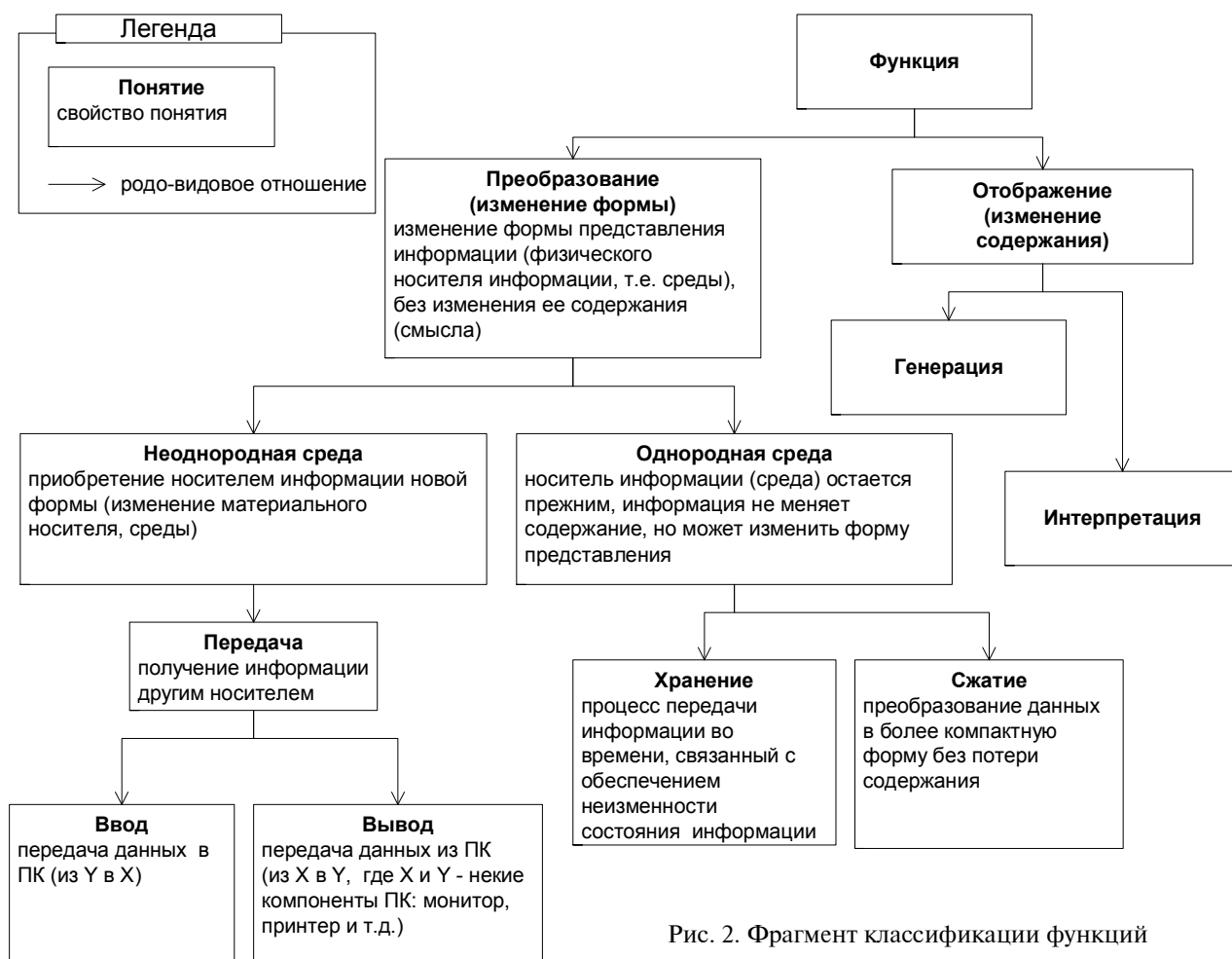


Рис. 2. Фрагмент классификации функций

(два аргумента и один результат), как наиболее распространенные. Любая операция, определенная для объекта данных заданного типа, может быть реализована одним из трех основных способов:

- 1) непосредственно в аппаратной части компьютера;
- 2) как процедура или подпрограмма-функция;
- 3) как встраиваемая последовательность кодов.

Определение операции (элементарной или абстрактной) состоит из 2-х частей: спецификации и реализации. В спецификацию операции входит: имя, сигнатура (прототип) и действие (описание той функции, которую она вычисляет). Иногда бывает непросто представить точную спецификацию операции в виде математической функции:

- 1) в программе могут использоваться неявные аргументы;
- 2) при выполнении подпрограммы могут возникнуть неявные результаты;
- 3) для некоторых возможных значений аргументов подпрограмма может быть не определена;
- 4) подпрограмма может быть чувствительна к предыстории ее вызовов [10].

Таким образом, существует необходимость разработки классификации операций (выполняемых функций), описывающей операции абстрактными

понятиями. Для построения данного фрагмента классификации были использованы следующие источники информации [10-14, 26-28]. На основании полученных результатов разделим функции по области воздействия на: изменение формы представления информации и изменение содержания информации.

Примером функций, входящих в область изменения формы информации, могут служить следующие: ввод, вывод, чтение.

Ввод – “передача данных от внешнего устройства (источника) в основную память ЭВМ” [26, с. 46].

Ввод – “задавать данные программе с помощью вводного устройства” [27, с. 134].

Ввод “ввести; место, через которое что-нибудь входит, вставляется куда-нибудь” [14, с. 64].

Чтение – “выбор данных из внешнего устройства и запись в основную память” [26, с. 437].

Вывод – “передача данных из основной памяти во внешнее устройство” [26, с. 59].

Вывод – “отображать данные на экране дисплея или бумаге, записывать их в файл или передавать их по линиям связи” [27, с. 183].

Функция “ввод” указывает на преобразование чего-либо путем *ввода (переноса)* предмета из одного состояния в другое. Поэтому под функцией

преобразования информации “ввод” мы будем понимать *процесс переноса информации из одного места в другое, связанный с изменением формы представления информации (носителя информации) и не оказывающий влияния на её содержимое.*

Приведенные выше функции указывают на различность сред, в которых находилась информация. Для функций, модифицирующих информацию по форме, изменяющих её “местоположение” в однородной среде, выделим: передача, хранение, копирование.

“Передача – перемещение, пересылка данных от источника к приемнику” [26, с. 260].

“Хранение – процесс передачи информации во времени, связанный с обеспечением неизменности состояния материального носителя” [26, с. 431].

“Копирование – процесс получения копий; воспроизведение данных с сохранением исходной информации” [26, с. 170]. Однородными из той же области можно считать: дублирование, тиражирование.

Приведенные выше функции не влияют на носитель информации и её содержание, однако влияют на форму представления информации.

Выделив “качественную” границу для функций “преобразования формы”, установим область функций, применение которых повлечет изменение в содержании информации. Примерами таких функций могут служить: формализация, анализ, структурирование, удаление, уничтожение и т.д.

“Отображение – установление соответствия между элементами двух множеств: отображение множества X на множество Y есть функция, определенная на множестве X и принимающая значения из множества Y; установление соответствия между двумя структурами” [26, с. 248].

Такую ветвь в классификации, объединяющую данные функции (функции, изменяющие содержание информации), назовем “отображением”.

“Структуризация – детализация метода путем представления его в виде структуры. Основными показателями оценки структуризации являются модульность и четкость интерфейса” [26, с. 387]

“Формализация – процесс представления информации об объекте, процессе, явлении в формализованном виде” [26, с. 427].

“Формализация данных – выбор и разработка форм входных документов для представления данных в виде, требуемом для ввода в систему” [26, с. 427].

“Формализм – соблюдение внешней формы в ущерб существу дела” [28, с. 549].

“Формализм – соблюдение внешней формы в чем-нибудь в ущерб существу дела” [14, с. 761].

“Систематизация – процесс упорядоченного определения объектов, осуществляемый по сходству или различию присущих им признаков” [14, с. 365]

“Классификация – система распределения объектов (предметов, явлений, процессов, понятий) по классам в соответствии с определенными признаками” [14, с. 143]

“Оптимизация – улучшение характеристик системы” [26, с. 241]

“Восстановление – возврат к исходному значению или нормальному функционированию” [26, с. 54].

“Восстановление данных – процесс копирования данных с носителя, содержащего защитную копию данных, на носитель-оригинал в случае нарушения в нем целостности данных” [26, с. 54].

Следует отметить, что процесс восстановления для большинства существующих информационных систем косвенно связан с процессом копирования, резервирование – с процессом дублирования. Поэтому, используя общепринятые термины, мы косвенно понимаем их связь с другими функциями. Примером могут быть функции: “резервирование”, “копирование”, “сжатие”, “хранение”, между которыми может быть поставлено отношение часть-целое, указывающее, что при процессе создания резервной копии вы воспользуетесь функцией сжатия информации (для оптимизации места ее хранения), функцией копирования данной информации на другой носитель, функцией хранения резервной копии.

“Стирание – операция, делающая файл либо его фрагмент логически или физически недоступным” [26, с. 384].

“Удаление (удаление набора данных) – исключение метки файла из оглавления тома” [26, с. 407].

Смысл значения понятия “стирание” и “удаление” – практически одинаков и характеризуется тем, что “данная операция санкционирована (каким-либо субъектом), результатом которой является логическая или физическая недоступность всей или части информации для всех субъектов, работающих с данной информацией”.

“Разрушение (разрушение информации) – стирание всей или части информации, хранящейся в памяти ЭВМ” [26, с. 322].

Для данного термина мы скорректируем его трактовку: *под разрушением информации будем понимать уничтожение всей или части информации под воздействием сторонних сил (материальный износ, электромагнитный импульс и т.п.).*

Данные термины объединяют действие, выполнение которого может привести к полной или частичной потере информации, иначе говоря, к ее уничтожению.

“Уничтожить – прекратить существование кого-нибудь, чего-нибудь, истребить” [14, с. 742].

“Уничтожиться – прекратиться, упраздниться, пропасть” [14, с. 742].

Отметим тот факт, что рассмотрение всех функций, включая те (пример: формализация), которые не

могут выполняться машиной, помогает более четко понять область функций, доступных машине.

При рассмотрении вопросов, связанных с классифицированием информации, возникают неопределенности (пример: старение информации, накопление информации, “мусор”, избыточность информации, уничтожение информации), понимание которых специфично для рассматриваемой предметной области и требует уточнения. Известно, что уничтожить информацию полностью нельзя, её полное уничтожение должно сопровождаться уничтожением всех носителей информации и ссылок (упоминаний) на неё. Следовательно, под уничтожением информации следует понимать “уничтожение информации на текущем носителе”.

На практике большинство используемых связей имеют сформировавшуюся структуру, которую проще применять и легче наполнять информацией. Таким образом, дальнейшей задачей адаптации модели знаний является:

- выделение классов для однотипных связей;
- классификация данных классов;
- разработка способов представления выделенных классов в виде шаблонов;
- организация шаблонов в единую модель представления знаний (рис. 3).

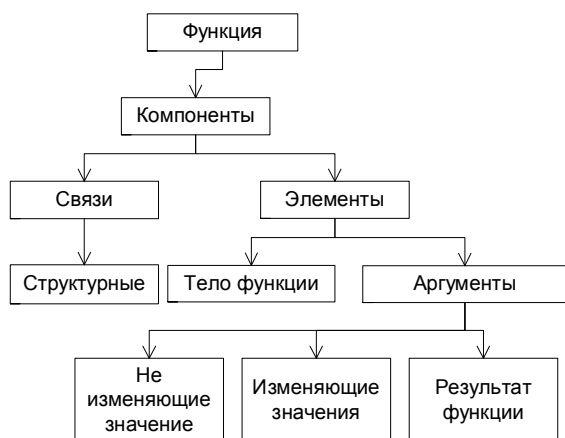


Рис. 3. Пример использования модели знаний для изучения функций

5. Выводы

Результаты, полученные при построении данных классификаций (см. рис. 1,2), использованы в адаптации предложенной в [5] модели знаний. Адаптированная модель представления знаний будет использована для формализации процессов имитационного моделирования при разработке моделей сложных программных систем, инструментария их моделирования, интеллектуальных баз данных программных компонентов.

Разработанные классификации, по сравнению с существующими [29,30], отличается ориентация на раскрытие функционального (сущностного) свойства классифицируемого объекта, что повышает качество представляемых знаний [20]. Использование при анализе предметной области предлагаемых

классификаций помогает сократить время, требуемое на анализ, по сравнению с существующей системой ARIS [31]. Значительно сократит время при разработке моделей знаний, ориентированных на использование классификаций, разрабатываемая модель знаний для информационно-образовательного комплекса «Поколение.ru» [32], применяемая модель знаний в программном инструментарии для построения обучающих систем основам дисциплин «TeachConcept» [25]. Использование классификаций в обучающих системах позволяет ориентировать обучаемого на изучение основ (сущностных свойств) изучаемого понятия, например, «TeachConcept», универсальная система дистанционного обучения «WebClass» [33].

Полученные результаты могут быть применены при решении практических задач, использующих знание-ориентированный подход, к примеру: «формализация семантики языков программирования дает дополнительное преимущество – появляется возможность доказать правильность программы» [11, с. 33].

Литература: 1. Кулибаба В.В. Обоснование использования методологии OMSAD при проектировании case-средств моделирования информационных систем // Проблемы бионики. 2002. Вып. 57. С. 30-34. 2. Одицов И.О. Профессиональное программирование. Системный подход. СПб.: БХВ-Петербург, 2002. 512 с. 3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. М.: “Издательство Бином”, СПб.: Невский диалект, 1998. 560 с. 4. Маторин С.И. О новом научном направлении системологического анализа сложных динамических объектов // Вестник ХГПУ. 2000. № 81. С. 15-18. 5. Маторин С.И. О решении проблемы согласования системного анализа с объектно-ориентированным подходом // KDS-2001. С. 481-489. 6. Маторин С.И. Анализ и моделирование бизнес-систем: системологическая объектно-ориентированная технология /Под ред. проф. Бондаренко М.Ф.. Харьков: ХНУРЭ, 2002. 322 с. 7. Маторин С.И., Попов А.С., Маторин В.С. Проектирование UFO-tolkit // Проблемы программирования. 2002. № 3-4. С. 22-25. 8. Маторин С.И. Определение и системологическое обоснование базовой иерархии классов для создания нормативной системы объектно-ориентированного анализа и проектирования // Вестник ХГПУ. Новые решения в современных технологиях. 2000. №79. С. 22-25. 9. Соловьева Е.А. Естественная классификация: системологические основания. Харьков: ХТУРЭ, 1999. 222 с. 10. Пратт Т., Зелковиц М. Языки программирования: разработка и реализация / Под общей ред. А. Матросова. СПб.: Питер, 2002. 688 с. 11. Бен-Ари М. Языки программирования. Практический сравнительный анализ: Пер. с англ. М.: Мир, 2000. 366 с. 12. Поляков Д.Б., Круглов И.Ю. Программирование в среде Турбо Паскаль (версия 5.5.): Справ. метод. пособие. М.: Издательство МАИ, А/О “Росвузнаука”, 1992. 576 с. 13. Шилдт Герберт. Справочник программиста по С/С++. Пер. с англ.: Уч. пос. М.: Издательский дом “Вильямс”, 2000. 448 с. 14. Ожегов С.И. Словарь русского языка. М.: Рус. яз., 1983. 816 с. 15. Семенов М.И., Трублин И.Т., Лойко В.И., Барановская Т.П. Автоматизированные информационные технологии в экономике. М.: Финансы и статистика, 2003.

416 с. **16.** *Бондарев В.Н.* Искусственный интеллект: Уч. пособие для вузов. Севастополь: СевНТУ, 2002. 615 с. **17.** www-ksl.stanford.edu **18.** Универсальная десятичная классификация. К.: Кн. Палата Украины, 1999. 344 с. **19.** *Всероссийский ИНИТИ* // www.viniti.ru/pro_class_udk.html **20.** *Соловьева Е.А., Маторин С.И.* О необходимости совершенствования УДК на основе естественной классификации // ХГПУ "Вестник" №82/2000. С. 45-53 **21.** *Uta Priss* Faceted Information Representation // <http://citeseer.nj.nec.com/rd/13929098,495844,1,0.25,Download/http:qSqqSqphp.indiana.eduqS~uprissqS. qSqpapersqSiccs00.ps>, 2000. **22.** *Дмитров В.И., Макаренко Ю.М.* CALS-стандарты // Автоматизация проектирования #02/97. http://www.osp.ru/ap/1997/02/16_print.htm **23.** *Гришан А.И.* Технические решения Отраслевого Фонда Алгоритмов и Программ Госкоорцентра. Тезисы доклада RELARN-2003. www.relarn.ru:8083/conf/conf2003/section1/1_09.html **24.** www.master-kniga.ru/index.html **25.** *Бондаренко М.Ф., Кулибаба В.В., Соловьева Е.А.* Базовый программный инструмент для построения обучающих систем основам дисциплин "TeachConcept" // Проблемы бионики. 2000. Вып. 53. С. 3-7. **26.** *Першиков В.И., Савинков В.М.* Толковый словарь по информатике. М.: Финансы и статистика, 1991. 543 с. **27.** *Борковский А.Б.* Англо-русский словарь по программированию и информатике. М.: Московская международная школа переводчиков, 1992. 335 с. **28.** *Словарь иностранных слов.* М.: Русский язык, 1981. 624 с. **29.** *Merkel Dieter* Content-Based Software Classification by Self-Organization // Proc. of the IEEE Int'l Conference on Neural Networks (ICNN'95). Perth. Australia. Nov 27 - Dec 1. 1995. P. 1086-1091. <http://citeseer.nj.nec.com/merkl95contentbased.html> **30.** *Girardi M.R. and Ibrahim B.* A Classification Scheme for Software Artifacts (Position Paper) // University of Geneva, C.U.I., Geneva, Switzerland. In Proceedings of Fifth ASIS SIG/CR (American Society for Information Science, Classification Research workshop), Oct. 16, 1994, Alexandria, Virginia, USA. ftp://cui.unige.ch/VISUAL_PROG/articles/

ASIS94.ps.gz. **31.** ARIS // <http://www.ids-scheer.com>. **32.** *Найханова Л.В., Аюшеева Н.Н., Евдокимова И.С.* Методологические основы отраслевой поисковой системы // http://rsi.spb.fio.ru/db/doc/get_thes.phtml?id=113. **33.** Универсальная система дистанционного обучения «WebClass» // <http://webclass.net.ru/>.

Поступила в редколлегию 05.06.2003

Рецензент: д-р техн. наук, проф. Шабанов С.Ю.

Маторин Сергей Игоревич, канд. техн. наук, доцент кафедры социальной информатики ХНУРЭ. Научные интересы: развитие системологии (системологический классификационный анализ, OMSAD, системно-объектный подход, УФО-технология). Увлечения, хобби: отдых на природе с друзьями. Адрес: Украина, 61000, Харьков, пр. Ленина, 16, к. 525, тел. 7021591. E-mail: nulpz@kture.kharkov.ua

Кулибаба Владимир Владимирович, аспирант кафедры социальной информатики ХНУРЭ. Научные интересы: применение системологического подхода (системно-объектного подхода) при изучении программных систем, исследование задач повторного использования программных компонент в программной инженерии и моделирование программных систем. Увлечения, хобби: отдых в шумных компаниях, веб-дизайн, программирование. Адрес: Украина, 61166, Харьков, пр. Ленина, 16, кв. 525.

E-mail: wolfUkr@ukr.net, nulpz@kture.kharkov.ua

Соловьева Екатерина Александровна, д-р техн. наук, кафедра социальной информатики ХНУРЭ. Научные интересы: развитие системологии, использование системологического подхода (в том числе системологический классификационный анализ) при изучении и моделировании сложных систем, развитие и изучение естественной классификации. Увлечения, хобби: отдых на природе с друзьями, встреча и прием гостей. Адрес: Украина, 61166, Харьков, пр. Ленина, 16, к. 525. E-mail: nulpz@kture.kharkov.ua