ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

11(147)
2008

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Аннотации статей размещены на сайте журнала по адресу http://www.**informika.ru/text/magaz/it/** или **http://novtex.ru/IT.**

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора наук.

УДК 621.3.019.3(004.3.06)

С. Г. Мосин, канд. техн. наук, доц.,
Владимирский государственный университет

Структурные решения тестопригодного проектирования заказных интегральных схем

Рассмотрены структурные решения, обеспечивающие проектирование тестопригодных заказных интегральных схем. Описаны особенности реализации тестирующих подсхем и правила их использования. Определены достоинства и недостатки представленных решений.

Ключевые слова: *тестопригодное проектирование, тестирующие подсхемы, сканируемые пути, встроенное самотестирование.*

Тестирование занимает существенное место в процессе проектирования и реализации электронных устройств. На мероприятия по тестированию интегральных систем (ИС) приходится порядка 40—60 % от общего времени, требуемого на разработку схемы. Высокие затраты связаны во многом с повышением сложности тестирования современных электронных изделий и необходимостью проведения тестовых мероприятий на каждом этапе процесса производства ИС. Сложность тестирования связана с такими факторами, как изменения в технологическом процессе; растущая степень интеграции; повышение функциональной сложности разрабатываемых устройств; отсутствие непосредственного доступа к внутренним компонентам интегральной схемы и др.

Сокращение затрат на тестирование и, как следствие, снижение себестоимости готового продукта связывают в первую очередь с разработкой и использованием новых, высокоэффективных тестовых стратегий, которые позволили бы упростить процесс тестирования.

Одним из перспективных направлений в данной области является использование подхода тестопригодного проектирования электронных устройств на ранних стадиях разработки (DFT — *Design for Testability*) [1, 2]. Данный подход позволяет еще на этапе проектирования заказной ИС исследовать возможности и подготовить рекомендации для последующего тестирования. Реализация подхода тестопригодного проектирования предусматривает использование внутри проекта специализированных тестирующих подсхем или на основе результатов схемотехнического анализа преобразование исходной схемы в целях повышения контролируемости ее параметров. Подход тестопригодного проектирования позволяет использовать функциональные и структурные особенности заказных ИС при формировании тестов и тестовых мероприятий.

Важная задача тестопригодного проектирования — анализ тестопригодности, в ходе которого получают количественную оценку простоты наблюдения внутреннего состояния ИС и установки произвольных значений в ее внутренних узлах, т. е. рассчитывают наблюдаемость и управляемость внутренних узлов схемы [3]. Для того чтобы подать тест в определенную функциональную часть схемы, необходимо установить соответствующую последовательность сигналов на ее входе. Для формирования каждой такой последовательности требуется приложить фиксированные значения к узлам схемы в области *A* путем назначения первичным входам ИС определенных входных наборов (рис. 1). Простота решения данной задачи для каждого внутреннего узла схемы определяет его управляемость.

После подачи входных тестов необходимо обеспечить наблюдение реакции проверяемой подсхемы относительно первичных выходов схемы. Для этого активизируют пути распространения сигналов, проходящие через область *B*, с помощью установки фиксированных значений на других узлах схемы через первичные входы. Простота решения данной задачи для каждого узла схемы определяет его наблюдаемость. Тестопригодность каждого узла рассчитывают как функцию значений его управляемости и наблюдаемости.

Для повышения легкости последующего тестирования в проектируемую схему включают дополнительные тестирующие подсхемы, обеспечиваю-

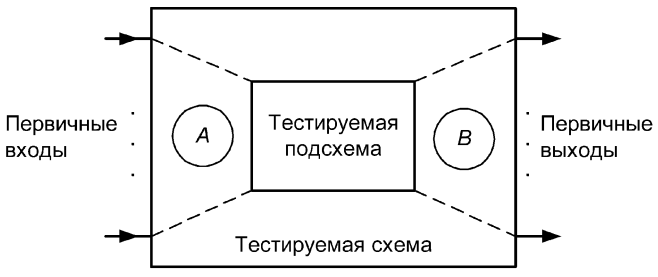


Рис. 1. Тестирование внутренних подсхем ИС

щие повышение уровня наблюдаемости и управляемости. В ходе анализа тестопригодности определяют наборы входных и выходных тестовых узлов, а также характеристики тестовых сигналов.

Предлагаемые *DFT*-решения реализуют, как правило, в виде дополнительных подсхем или функциональных элементов. Данные решения можно разделить на две категории:

- тестирование на основе реконфигурации исследуемого устройства;
- тестирование за счет кодирования внутренней информации [4].

Методы *первой категории* требуют использования дополнительных переключающих цепей и элемента выбора режима тестирования, при котором собственно и происходит реконфигурация оригинальной схемы. Такое преобразование внутренней структуры устройства, которое позволяет улучшить его тестирование, может быть выполнено двумя способами:

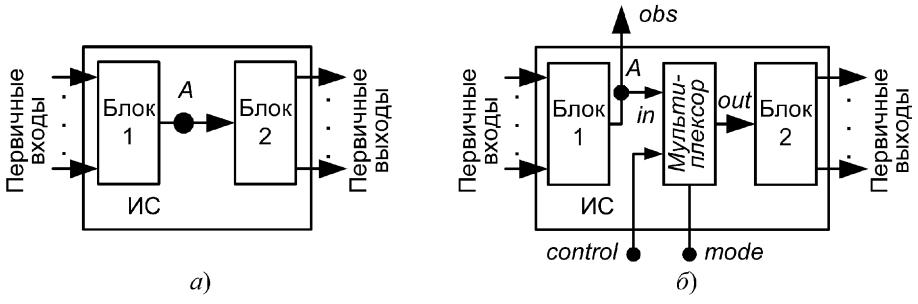
- тестируемую схему разделяют на несколько функциональных блоков таким образом, что их входы и выходы становятся непосредственно управляемыми и наблюдаемыми. Данный способ реконфигурации упрощает доступ к компонентам схемы, но зависит от конкретного приложения. Успех и эффективность его использования во многом зависит от квалификации проектировщика;
- реализуют механизмы реконфигурации оригинальной схемы для трансформирования ее внутренней структуры за счет изменения способа соединения компонентов в целях получения новой с функциональной точки зрения схемы, тестирование которой представляет более простую задачу. Отклики сформированной таким образом схемы используют для оценки работоспособности оригинальной схемы. Особенностью методов, реализованных в рамках данного способа, является их универсальность в пределах отдельных классов устройств.

Методы *второй категории* используют для осуществления интерактивного тестирования устройств, когда решают проблему измерения внутрисхемных характеристик в режиме реального времени. На основе значений измеренных параметров формируют числовой код, который используется схемой проверки для определения состояния работоспособности тестируемого устройства. Присутствие неисправности выявляют в случае, когда происходит нарушение числового кода, о чем схема проверки начинает немедленно сигнализировать. Выбор тестовых узлов и измеряемых характеристик, выполняемый еще на этапе проектирования устройства, осуществляют таким образом, чтобы как можно большее число возможных неисправностей приводило к нарушению формируемого числового кода.

Рассмотрим основные структурные решения тестопригодного проектирования заказных ИС, которые могут быть легко реализованы в виде стандартных ячеек или макроблоков и включены в топологию ИС в виде тестирующей подсхемы.

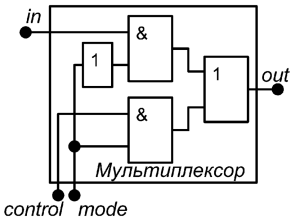
Повышение управляемости и наблюдаемости. Это одна из основных задач тестопригодного проектирования, которая обеспечивает эффективное тестирование электронной схемы [3]. Когда внутреннему узлу устройства сложно назначить определенный сигнал через первичный вход или наблюдать его состояние относительно первичного выхода, эффективным способом организации доступа к данному узлу является подключение такого узла к внешнему тестовому выходу. Например, для схемы, приведенной на рис. 2, подключение внутреннего узла *A* к внешнему выводу микросхемы *obs* позволяет обеспечить контроль выходного состояния блока 1 и управление входным сигналом блока 2. Для наблюдения состояния выходного сигнала блока 1 достаточно непосредственно подключить его к контактной площадке внешнего вывода ИС. Управление блоком 2 означает возможность устанавливать его входной сигнал в состояние логического 0 или логической 1, сохраняя при этом целостность линии передачи сигналов от блока 1 к блоку 2.

Реализовать данную логическую функцию можно с помощью мультиплексора, представленного на рис. 3. Здесь в зависимости от выбранного режима *mode* — нормальный или тестирование, на выход *out* поступает либо входной сигнал *in*, либо детерминированный сигнал 0 или 1, назначенный на управляющем входе *control*. При нормальном режиме выходной сигнал блока 1 непосредственно наблюдаем относительно внешнего вывода и поступает на вход блока 2, обеспечивая при этом взаимодействие обоих блоков. В режиме тестиро-



а) *б*)

Рис. 2. Интегральная схема:
а — без механизма управляемости и наблюдаемости внутреннего узла; *б* — с механизмом управляемости и наблюдаемости



control: 0 или 1.
mode: 0 – нормальный режим;
1 – режим тестирования.
out = (*mode* & *in*) | (*mode* & *control*)

Рис. 3. Мультиплексор управления внутренним узлом схемы

вания выходной сигнал блока 1 по-прежнему наблюдаем относительно внешнего вывода, а на вход блока 2 может быть подан детерминированный сигнал в виде логического нуля или логической единицы, что реализует управляемость данного входа.

Использование внутренних мультиплексоров. Предназначено для доступа к внутренним узлам ИС при сокращенном числе используемых внешних выводов (рис. 4). Данный подход предусматривает использование мультиплексоров и демультиплексоров, реализованных на уровне кристалла ИС, которые улучшают управляемость и наблюдаемость входов и выходов внутренних блоков устройства. Такой механизм позволяет обеспечить доступ через внешние выводы ИС к внутренним сигналам отдельного функционального блока для наблюдения или управления его состоянием, не используя при этом остальные блоки.

Метод сканируемого пути. Проектирование с применением метода сканирования следует рассматривать в качестве способа, позволяющего упростить процедуру формирования тестов для цифровых схем, в состав которых входят элементы памяти и цепи обратной связи [5, 6]. В основе этого метода лежит принцип, основанный на разделении устройства на две части — комбинационную и набор элементов памяти, которые управляются системными тактовыми импульсами. Входы комбинационной части разделяют на первичные входы (x_1, x_2, ..., x_n) и входы цепей обратной связи (y_1, y_2, ..., y_p). Таким образом, состояния на выходах устройства (z_1, z_2, ..., z_m) зависят от текущего значения сигналов на первичных входах системы и состояний элементов памяти, входящих в ее состав. Кроме того, последующее состояние элементов памяти также зависит от значений сигналов на первичных входах и состояний самих элементов памяти в текущий момент времени. Данная зависимость последующего состояния схемы от предыдущего определяет существенную сложность построения теста для последовательностных схем. На практике первичные входы устройства обеспечивают единственно доступный интерфейс непосредственного управления состоянием схемы, а первичные выходы позволяют выполнять непосредственное наблюдение за данными состояниями. Следует отметить, что управление элементами памяти и наблюдение за их состоянием возможно исключительно через комбинационную часть устройства.

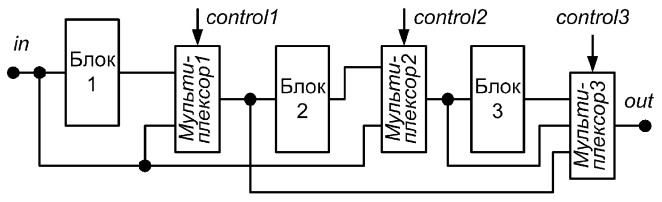


Рис. 4. Схема организации доступа к внутренним узлам на мультиплексорах

Основной проблемой при тестировании таких устройств является выбор, какую из двух частей проверять первой при условии, что ни одна из них непосредственно не управляема и не наблюдаема и обе части схемы явно зависят друг от друга.

Использование при проектировании принципа сканируемого пути позволяет уменьшить сложность структуры схемы за счет следующих дополнительных свойств:

- возможность выполнять проверку элементов памяти отдельно от комбинационной части схемы;
- возможность устанавливать внутренние переменные в любое требуемое состояние независимо от предыдущего состояния;
- возможность непосредственного наблюдения выходных сигналов комбинационной части, которые поступают на входы элементов памяти.

Для исполнения перечисленных свойств в схему включают дополнительную логику, обеспечивающую реализацию метода сканируемого пути через элементы памяти (рис. 5).

В структуру схемы включены четыре дополнительные интерфейсные сигналы:

- *SDI* (*Shift Data In*) — входные сканируемые данные;
- *SDO* (*Shift Data Out*) — выходные сканируемые данные;
- *SMC* (*Scan Mode Control*) — управление режимом сканирования;
- *CLK* (*Clock*) — системный тактовый импульс.

В данном решении каждому элементу памяти предшествует двухвходовый мультиплексор, которым управляют с помощью общего сигнала выбора режима сканирования *c* (рис. 6). Если сигнал *c* равен 0, то выходной сигнал комбинационной части поступает на вход соответствующего элемента памяти. При этом в целом схема работает в режиме нормального функционирования. Когда сигнал *c* принимает значение 1, схема функционирует в режиме сканирования, при этом происходит реконфигурирование элементов памяти в один последовательный сдвиговый регистр. Вход

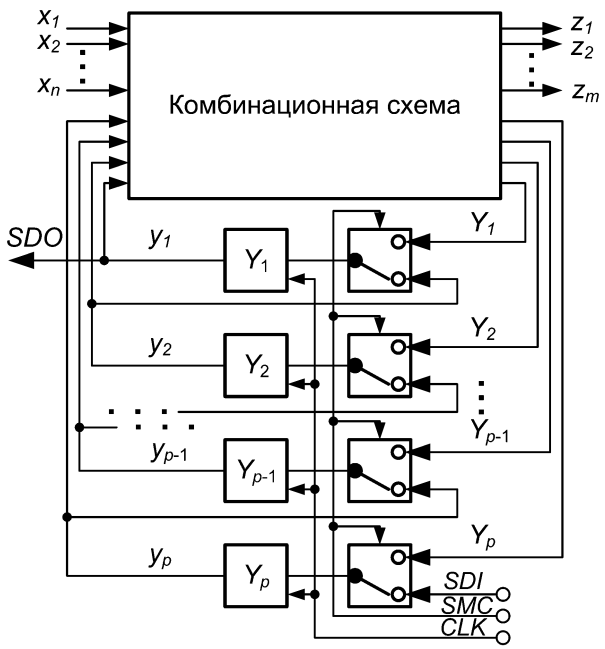


Рис. 5. Схема, использующая принцип сканируемого пути

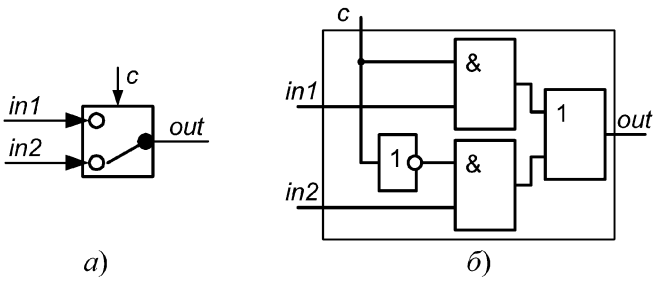


Рис. 6. Переключатель: *а* — условное обозначение, *б* — схемная реализация

регистра, на который последовательно поступают данные, называют *входом сканируемых данных*. Выход, с которого последовательно получают данные, называют *выходом сканируемых данных*. В режиме сканирования элементы памяти последовательностной схемы можно установить в любое заданное множество состояний, прикладывая последовательность сигналов к входу сканируемых данных и выполняя сдвиг регистра под действием системного тактового импульса.

Общий алгоритм тестирования с помощью сканируемого пути можно представить следующим образом.

1. Установить режим сканирования. В полученном сдвиговом регистре проверить правильность функционирования каждого элемента памяти, используя вход и выход сканируемых данных и системный тактовый импульс.

2. Сформировать множество тестовых наборов для комбинационной части схемы, принимая во внимание непосредственную управляемость как первичных входов, так и внутренних входов элементов памяти, а также непосредственную наблюдаемость всех выходов, т. е. первичных выходов и выходов элементов памяти.

3. Применить каждый тестовый набор. Для этого необходимо:

- приложить к первичным входам схемы соответствующие сигналы из тестового набора. Установить режим сканирования. В элементы памяти сдвигового регистра записать остальные тестовые сигналы;
- установить режим нормального функционирования. Полученные в ходе работы состояния внутренних выходов комбинационной части записать в элементы памяти, применяя системный тактовый импульс;
- установить режим сканирования. Синхронно с системным тактовым импульсом вывести содержимое элементов памяти сдвигового регистра через выход сканируемых данных. Провести сравнение полученных откликов с эталонными реакциями и принять решение о состоянии схемы.

В данном методе вместо проверки последовательностной схемы как единого устройства выполняют контроль отдельно каждой части — комбинационной схемы и элементов памяти. При использовании стандартных тестов для проверки элементов памяти задачу формирования тестов можно свести к процедуре построения тестов для комбинационной части устройства.

Серьезный недостаток рассмотренного метода — потребность в дополнительном оборудовании (переключательных элементах). Поскольку переключательные элементы работают как в режиме сканирования, так и в режиме нормального функционирования устройства, то происходит снижение быстродействия схемы.

Для устранения данного недостатка при проектировании последовательностных схем используют специальные структуры элементов памяти, которые позволяют формировать сканирующие пути без использования схем переключения. Одним из таких подходов является проектирование на основе метода сканирования, чувствительного к уровню тактового сигнала (*LSSD — Level Sensitive Scan Design*) [5].

Базовый триггер для *LSSD* представлен на рис. 7. Он состоит из двух триггеров-защелок L_1 и L_2. Защелку L_1 используют в нормальном и сдвиговом режимах, а L_2 — только в сдвиговом режиме. Сигнал D определяет системные входные данные; *Clk* — системный тактовый импульс; *SI* — вход сканируемых данных; A и B — тактовые импульсы сканирования (сдвига).

На основе базовых триггеров строят последовательностную схему со сканируемым путем (рис. 8).

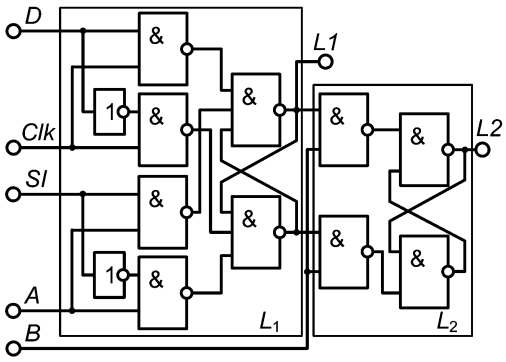


Рис. 7. Базовый триггер *LSSD*

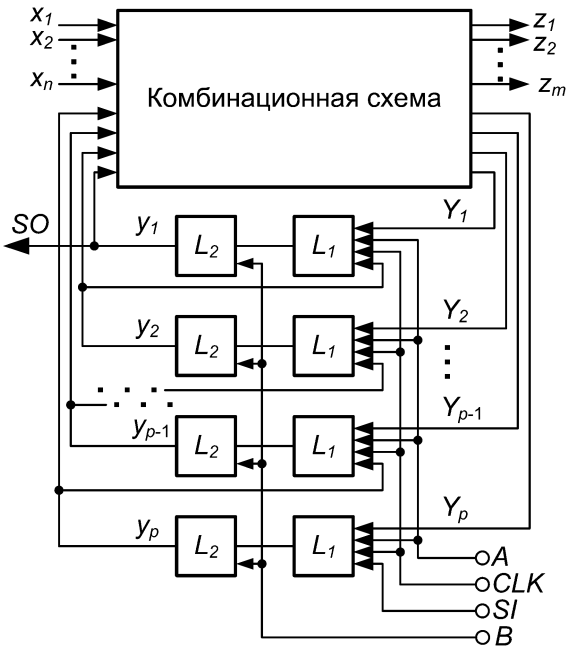


Рис. 8. Схема сканирования на основе *LSSD*

Переключение режимов в данной схеме происходит с помощью синхросигналов *A* и *B*, что позволяет не снижать общего быстродействия устройства.

При нормальной работе защелка L_1 захватывает входные данные под действием входа синхронизации *Clk* и пересылает их на выход системных данных L_1 (рис. 7). В режиме тестирования, когда защелки работают как сдвиговый регистр, используют неперекрывающиеся синхросигналы сдвига *A* и *B*. Входные сканируемые данные поступают в защелку L_1 при использовании синхросигнала сдвига *A*. Активизируя сигнал сдвига *B*, данные из L_1 поступают на выход сканирования L_2.

С помощью входа *SI* тестовые данные загружают последовательно в регистр, а с выхода *SO* — сигнал выхода сканируемых данных — содержимое регистра поступает на внешние устройства.

В практических реализациях *LSSD*-схем сканируемый путь формируют соединением защелок в сдвиговый регистр, когда выходной сигнал L_2 одного базового триггера подключают к входу сканируемых данных другого базового триггера. Сигналы *A* и *B* являются общими для всех триггеров.

Стандарт цифрового граничного сканирования IEEE 1149.1 [7]. Базовая идея данного стандарта связана с использованием технологии граничного сканирования, которая является разновидностью методов сканируемых путей и состоит в размещении специальных ячеек между каждым выводом ИС и ее внутренней логикой. Данные ячейки образуют последовательный сдвиговый регистр и содержат значения сигналов каждого вывода ИС. Граничное сканирование реализует бесконтактный метод доступа к выводам ИС, обеспечивая полный контроль состояния внутреннего функционирования схемы.

Базовая архитектура смешанной тестовой шины представлена на рис. 9. Она включает следующие основные элементы:

- цифровые граничные ячейки (*BSC — Boundary Scan Cell*);
- тестирующую схему, в состав которой входят:
  — контроллер доступа теста (*TAP — Test Access Port*);
  — регистр команд (*IR — Instruction Register*);
  — регистры данных.

На рис. 10 приведена структура цифровой ячейки граничного сканирования (*BSC*). Выделяют следующие режимы ее функционирования:

- нормальный, при котором данные проходят напрямую с входа *PI* на выход *PO*.
- режим обновления, когда данные на выход *PO* поступают с выходного регистра;
- режим захвата, при котором данные с входа *PI* поступают на сдвиговый регистр;

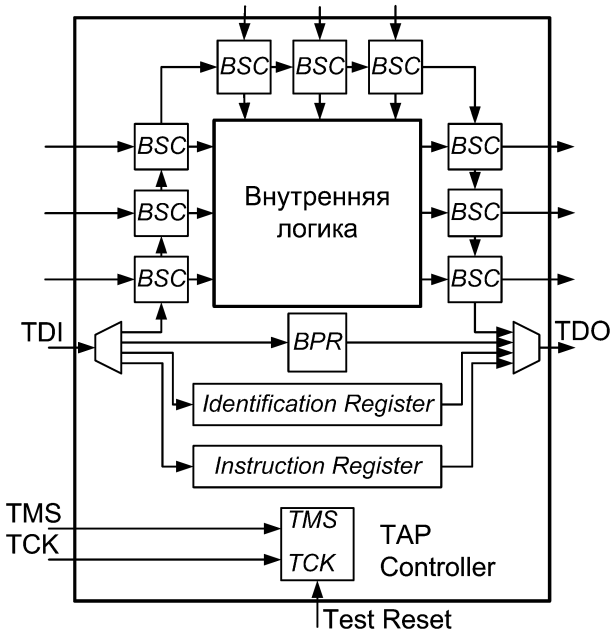


Рис. 9. Архитектура цифрового граничного сканирования

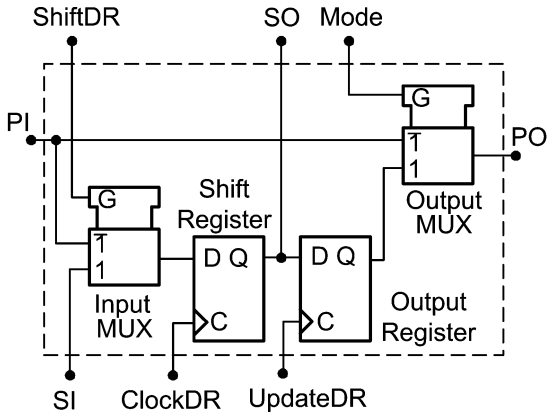


Рис. 10. Структура ячейки цифрового граничного сканирования (*BSC*)

- режим последовательного сдвига, при котором данные поступают с выхода *SO* (*Scan Out*) одной ячейки на вход *SI* (*Scan In*) следующей ячейки.

Из структуры ячейки видно, что режимы захвата и последовательного сдвига могут быть использованы совместно с нормальным режимом, а это дает возможность проводить тестирование устройства в процессе его функционирования.

Ячейки цифрового граничного сканирования образуют сдвиговый регистр граничного сканирования, который функционирует в двух режимах — последовательном и параллельном. В последовательном режиме возможна передача данных через сдвиговый регистр (см. рис. 9) от внешнего тестового входа (*TDI*) к внешнему тестовому выходу (*TDO*). В параллельном режиме происходит сохранение значений с первичных входов ИС в ячейки регистра или назначение логических состояний из ячеек регистра первичным выходам ИС.

Элементы граничного сканирования не изменяют функционирования внутренней логики устройства и не зависят от реализуемой устройством функции.

Назначение *TAP* — управление данными и режимами тестирования. Стандарт граничного сканирования предусматривает наличие следующих обязательных тестовых выводов:

- *TDI* (*Test Data In*) — последовательный вход тестовых данных;
- *TDO* (*Test Data Out*) — последовательный выход тестовых данных;
- *TMS* (*Test Mode Select*) — сигнал выбора режима тестирования;
- *TCK* (*Test Synchronizing Clock*) — сигнал синхронизации тестовой логики, независимый от системного синхросигнала.

В архитектуре предусмотрен также опциональный сигнал

- *TRST* (*Test Reset*) — асинхронный сигнал тестового сброса, независимый от *TCK*.

Логика функционирования *TAP*-контроллера реализована синхронным конечным автоматом с 16 внутренними состояниями. Контроллер реагирует на поступающий сигнал выбора режима тестирования (*TMS*) по переднему фронту сигнала синхронизации тестовой логики (*TCK*). В контроллере можно выделить шесть устойчивых состояний, которые могут поддерживаться бесконечно долго при каждом *TCK* и неизменном сигнале *TMS*: *Test-Logic Reset*, *Run-Test/Idle*, *Shift_DR*, *Pause_DR*, *Shift_IR* и *Pause_IR*. Причем первое из перечисленных устойчивых состояний будет обеспечено при равенстве сигнала *TMS* логической единице, а остальные — при равенстве *TMS* логическому нулю.

В каждый момент времени только один из внутренних регистров может быть подключен между внешними выводами *TDI* и *TDO*. В архитектуре цифрового граничного сканирования можно выделить следующие внутренние регистры:

- *BPR* (*Bypass Register*) — регистр непосредственной передачи данных с входа *TDI* на выход *TDO*;
- *BSR* (*Boundary-Scan Register*) — регистр, образованный ячейками граничного сканирования;
- *IR* (*Instruction Register*) — *n*-битный регистр команд ($n >= 2$), содержащий текущую инструкцию;
- *IdR* (*Identification Register*) — 32-разрядный идентификационный регистр, содержащий постоянный уникальный код устройства;
- *UserDR* (*User Data Registers*) — регистры данных, реализуемые на этапе проектирования во внутренней логике схемы.

Регистр команд должен обладать длиной не менее двух разрядов, чтобы обеспечить кодирование трех обязательных инструкций.

1. *Непосредственная пересылка* (*Bypass Instruction*), когда схема находится в режиме функционирования и *BPR*-регистр подключен между внешними выводами *TDI* и *TDO*. Пересылка тестовых данных происходит параллельно без влияния на работу схемы. Код данной инструкции состоит из всех единиц ("111...11"), записанных в регистр команд.

2. *Захват / Установка* (*Sample/Preload Instruction*), когда схема находится в режиме функционирования и регистр граничного сканирования (*BSR*) подключен между внешними выводами *TDI* и *TDO*. По данной команде в режиме сканирования данных выходные отклики схемы, формируемые на первичных выходах ИС, помещают в регистр *BSR*. Кроме того, данную инструкцию используют для установки тестовых сигналов в регистр граничного сканирования до проведения тестирования. Для кодирования инструкции используют последовательность "01".

3. *Внешнее тестирование* (*Extest Instruction*), когда ИС переводят в режим внешнего граничного сканирования и *BSR*-регистр подключен между выводами *TDI* и *TDO*. При выполнении данной инструкции регистр граничного сканирования используют для пересылки тестовых данных между несколькими ИС, обладающими интерфейсом IEEE 1149.1. Код данной инструкции состоит из всех нулей ("000...00"), записанных в регистр команд.

Встроенное самотестирование. Базовая идея использования встроенного самотестирования (*BIST — Built-in Self-Test*) заключается в реализации внутри проектируемой микросхемы специальных подсхем, которые обеспечивают внутреннее формирование тестовых воздействий и анализ выходных откликов. Одним из наиболее распространенных решений, обеспечивающих реализацию генераторов тестовых воздействий и подсхем компактного представления выходных откликов, является *сдвиговый регистр с линейными обратными связями* (*LFSR — Linear Feedback Shift Register*) [8, 9]. Данные регистры при реализации требуют существенно меньшей площади кристалла по сравнению с обычными двоичными счетчиками за счет использования минимальной комбинационной части на каждый триггер. Выходные состояния *LFSR*-регистра можно рассматривать как псевдослучайные последовательности.

Если у сдвигового регистра отсутствует внешний вход, то его принято называть автономным сдвиговым регистром с линейными обратными связями (*ALFSR — Autonomous LFSR*), который используют для формирования псевдослучайных тестовых шаблонов. При наличии у сдвигового регистра внешнего входа он обладает возможностью уплотнения (компактного представления) тестовых откликов исследуемых схем. Такой регистр принято называть *сигнатурным анализатором*.

Сдвиговый регистр с линейными обратными связями принято описывать характеристическим полиномом

$$P_n(X) = h_n x^n + h_{n-1}x^{n-1} + h_{n-2}x^{n-2} + \ldots + h_{1x} + h_0 = \sum_{i=0}^{n} h_i x^i, \qquad (1)$$

где n — длина сдвигового регистра; h_i — коэффициент присутствия обратной связи, принимающий значение 0, когда в i-м разряде обратная связь отсутствует, и 1, когда из данного разряда взята обратная связь.

Сигналы обратной связи складывают по модулю 2, и результат поступает на вход сдвигового регистра (рис. 11).

Автономный сдвиговый регистр формирует циклические линейные коды, снимаемые с выходов триггеров. Кодовые слова можно характеризовать длиной p (p — число неповторяющихся комбинаций) и числом информационных битов n (n — число используемых триггеров).

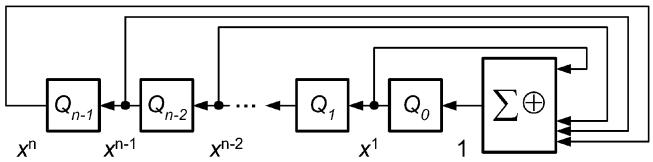


Рис. 11. Автономный сдвиговый регистр с линейными обратными связями: Q_i — триггеры

Внутренние состояния сдвигового регистра периодически повторяются. Длина максимально возможной последовательности уникальных шаблонов для n-разрядного *LFSR* составляет $2^n - 1$.

Характеристический полином может быть следующих видов:

- неприводимый, который невозможно разложить на многочлены;
- приводимый, который можно представить произведением многочленов;
- примитивный — неприводимый полином степени n, который делит полином $x^{2^n-1} + 1$ нацело.

При построении компактного генератора тестовых шаблонов с максимальным числом уникальных состояний в качестве характеристического полинома необходимо выбирать примитивный полином.

Для получения множества уникальных шаблонов максимальной длины требуется инициализировать регистр начальным значением, отличным от нуля. В противном случае *LFSR*-регистр будет сохранять исходное состояние, обеспечивая формирование множества единичной длины. Данное свойство справедливо для всех сдвиговых регистров с линейными обратными связями.

Чтобы обеспечить формирование всех возможных состояний, включая нулевое, в схему *LFSR*-регистра добавляют специальную логику декодирования, например, реализуемую элементами И-НЕ или ИЛИ-НЕ.

При использовании сдвигового регистра с линейными обратными связями в качестве сигнатурного анализатора на его вход подают двоичную последовательность тестовых откликов схемы $a(X)$, которая суммируется по модулю 2 с сигналами обратных связей (рис. 12). Характеристиче-

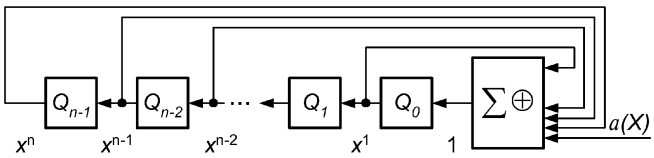


Рис. 12. Сигнатурный анализатор на *LFSR*

ский полином, описывающий сигнатурный анализатор, должен быть примитивным.

Входную двоичную последовательность можно записать в виде многочлена

$$a(X) = a_g x^g + a_{g-1}x^{g-1} + a_{g-2}x^{g-2} + \ldots + \\ + a_{1x} + a_0 = \sum_{j=0}^{g} \mathrm{S} a_j x^j, \qquad (2)$$

где a_j — коэффициенты, принимающие значения 0 или 1.

Многочлен $a(X)$ поступает на вход сумматора по модулю 2, начиная с коэффициентов высших порядков. При этом сдвиговый регистр с линейными обратными связями выполняет преобразование многочлена, эквивалентное его делению по модулю 2 на инверсию полинома обратных связей $P'_n(X)$:

$$P'_n(X) = X^{-n}P_n(X) = h_n x^0 + h_{n-1}x^1 + \\ + h_1 x^{n-1} + h_0 x^n = \sum_{\substack{i=0 \\ k=n}}^{\substack{k=0 \\ i=n}} h_i x^k. \qquad (3)$$

Деление многочлена (2) на полином (3) можно представить формулой деления многочленов:

$$\frac{a(X)}{P'_n(X)} = q_n(X) + \frac{r_n(X)}{P'_n(X)}, \qquad (4)$$

где $q_n(X)$ — частное от деления (целая часть); $r_n(X)$ — остаток от деления.

Частное $q_n(X)$ соответствует разрядам выходной двоичной последовательности, выдвигаемым из сдвигового регистра, а остаток $r_n(X)$ — двоичное число, хранимое в сдвиговом регистре.

В общем случае при тестировании необходимо контролировать состояния выходных откликов относительно нескольких выходных узлов. В этом случае можно формировать сигнатуры независимо для каждого выхода, однако такое решение потребует существенных затрат ресурсов для реализации n-наборов сдвиговых регистров, где n — число контролируемых узлов схемы.

Другой подход связан с использованием одного сдвигового регистра, на вход которого поступают значения сигналов с n-выходов схемы с временным мультиплексированием (рис. 13). В этом случае для тестирования в режиме реального времени скорость работы *LFSR* должна быть в n раз выше скорости проверяемой схемы. Для этого сдвиговый регистр тактируют с использованием внешнего синхросигнала большей частоты. При тестировании схемы в выделенном режиме можно понижать скорость работы основной логики в n раз, обеспечивая накопление сигнатуры относительно реакций от n узлов с номинальной частотой. Однако при обеих реализациях критическим показателем, ограничивающим использование временного мультиплексирования, является число контролируемых узлов n.

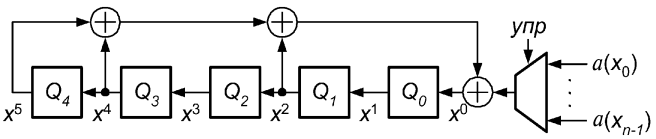


Рис. 13. Сдвиговый регистр с мультиплексированием нескольких входных сигналов

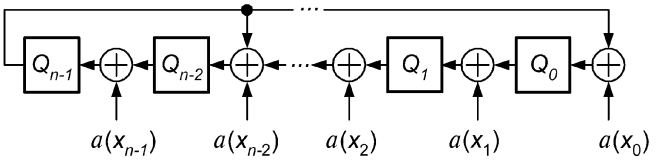


Рис. 14. Многовходовый сдвиговый регистр с линейными обратными связями

Универсальное решение формирования сигнатуры для нескольких выходов цифровой схемы — многовходовый сдвиговый регистр с линейными обратными связями (*MISR — Multiple-Input Signature Register*), представленный на рис. 14 [4]. В данной схеме применяют набор элементов *исключающее ИЛИ*, выходы которых поступают на входы триггеров сдвигового регистра. Такая организация обеспечивает обработку сигналов, получаемых с множества выходных узлов, в параллельном режиме, что определяет возможность проводить тестирование в реальном времени.

В общем случае проведение сигнатурного анализа предполагает, что исследуемую схему переводят в выделенный режим, когда на нее поступают тестовые сигналы из внутреннего генератора, а не сигналы с первичных входов. Для организации сигнатурного анализа в рабочем режиме необходимо использовать специальные, встроенные в схему, наблюдатели (*BILBO — Built-in Logic Block Observer*), которые в зависимости от режима тестирования могут функционировать как регистр данных или сдвиговый регистр с линейными обратными связями [10].

Структура *BILBO* объединяет функции регистра, сдвигового регистра, *LFSR* и *MISR*, реализованных на одном множестве триггеров-защелок (рис. 15). Кроме того, каждый модуль *BILBO* может функционировать в режиме либо генератора тестовых векторов, либо сигнатурного накопителя. Для проведения тестирования необходимы два таких блока, реализующие соответственно обе эти функции, хотя в каждом конкретном тестовом сценарии *BILBO*-генератор может выполнять функцию накопителя сигнатуры, и наоборот.

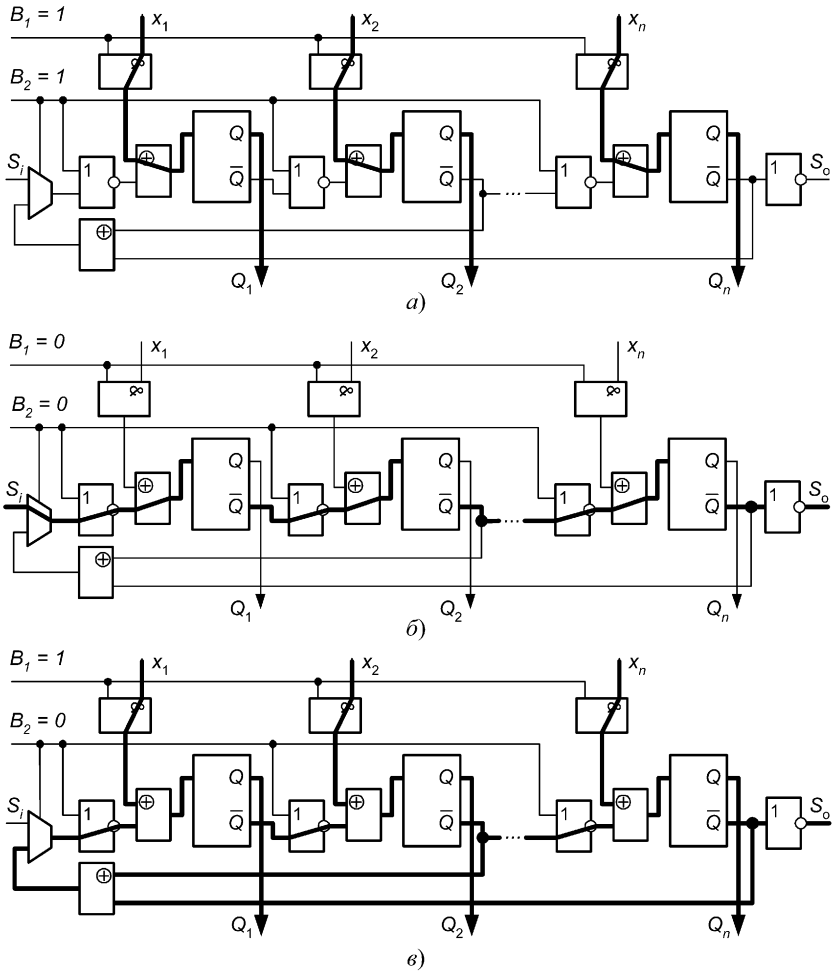


Рис. 15. Режимы функционирования *BILBO*: *а* — нормальный; *б* — сдвиг; *в* — *MISR*

Режим функционирования *BILBO* задают с помощью внешних управляющих сигналов — B_1 и B_2. В нормальном режиме ($B_1 = 1$ и $B_2 = 1$) обеспечивается механизм проверки работоспособности модуля *BILBO* (рис. 15, *а*). Сигналы x_i подают на вход соответствующего триггера через пару логических вентилей — И и исключающее ИЛИ. Изменяя значения входных сигналов x_i и контролируя при этом выход триггера Q_i, можно определить наличие константных неисправностей на входах и выходах вентилей, а также проверить способность триггера сохранять текущее значение и переходить из состояния логической единицы в состояние логического нуля, и наоборот ($0 \to 1$, $1 \to 0$).

В режиме сдвига, когда $B_1 = 0$ и $B_2 = 0$, модуль *BILBO* реализует функцию сдвигового регистра с линейной обратной связью (рис. 15, *б*). В данном режиме биты входной последовательности S_i заполняют сдвиговый регистр, обеспечивая перевод триггеров в определенные состояния. Данный режим обеспечивает механизм управляемости внутренних узлов схемы посредством единственного внешнего вывода S_i, обеспечивая инициализацию активизирующих входов внутренних компонентов проверяемой схемы.

Когда управляющие сигналы принимают значение $B_1 = 1$ и $B_2 = 0$, модуль *BILBO* функционирует в режиме *LFSR* или *MISR* в зависимости от числа поступающих на вход сигналов *x*. Если в качестве входа используют только сигнал x_1, то *BILBO* реализует сдвиговый регистр с линейной обратной связью. При подаче на вход нескольких или всех сигналов x_i модуль *BILBO* реализует многовходовый сигнатурный регистр (рис. 15, *в*). Схема *BILBO* при работе в режиме *LFSR* или *MISR* может работать как генератор тестовых сигналов, формируемых на выходах Q_i, или как сигнатурный накопитель.

Заключение

В работе рассмотрены основные структурные решения, которые легко реализовать в виде стандартных ячеек и макроблоков, обеспечивающих тестопригодность заказных ИС. Представленные решения реализуют механизмы, которые упрощают тестирование заказных ИС высокой степени интеграции, обладающих минимальным набором внешних выводов. Основной недостаток структурных решений тестопригодного проектирования — потребность в дополнительной площади кристалла для реализации тестирующих подсхем.

Список литературы

1. **Williams T. W., Parker K. P.** Design for Testability — A Survey // Proceedings of the IEEE. 1983. Vol. 71. N 12. P. 98—112.
2. **Мосин С. Г.** Подходы тестопригодного проектирования аналоговых интегральных схем // Радиоэлектроника и информатика. 2003. № 1. С. 49—59.
3. **Bennetts R. G.** Design of Testable Logic Circuits. Addison-Wesley, 1984.
4. **Novak O., Gramatova E., Ubar R., Mosin S, et al.** Handbook of Testing Electronic Systems. Czech Technical University Publishing House. 2005. 402 p.
5. **Eichelberger E. B., Lindbloom E., Waicukauski J. A. and Williams T. W.** Structured Logic Testing. Englewood Cliffs, New Jersey: Prentice-Hall, 1991.
6. **Sheth A. M., Savir J.** Scan Latch Design for Test Applications. Journal of Electronic Testing. 2004. Vol. 20. Issue 2. P. 213—216.
7. **IEEE Std 1149.1-1990.** Test Access Port and Boundary-Scan Architecture. IEEE. USA, 1995.
8. **Гуляев В. А., Кудряшов В. И.** Автоматизация наладки и диагностирования микроУВК. М.: Энергоатомиздат, 1992. 256 с.
9. **Kaligeros E., Kavousianos X., Bakalis D., Nikolos D.** A New Reseeding Technique for LFSR-based Test Pattern Generation // Proc. of 7th IOLTW On-line Testing Workshop. 2001. P. 80—86.
10. **Mourad S., Zorian Y.** Principles of Testing Electronic Systems. John Wiley & Sons, Inc. 2000. 420 p.

УДК 004.4.068

В. Г. Мокрозуб, канд. техн. наук, доц.,
Тамбовский государственный
технический университет

Представление структуры изделий в реляционной базе данных

Рассмотрены способы представления групповой спецификации изделий таблицами реляционной базы данных. Предложен формат групповой спецификации, отличающийся тем, что принадлежность деталей сборочным единицам задается логическим выражением, формируемым с использованием конструкторского обозначения сборочных единиц.

Ключевые слова: *изделие, спецификация, реляционная база данных.*

Введение

Одним из основных информационных массивов автоматизированных систем управления производством (АСУП) на промышленных предприятиях являются сведения о выпускаемой продукции. Базовым элементом этой информации служат конструкторские спецификации, которые составляются для сборочных единиц и отражают состав и структуру изделия. В дальнейшем эта информация используется всеми подразделениями предприятия на разных этапах жизненного цикла изделия (технологическая подготовка, материальное снабжение, изготовление и др.). Способы представления структуры изделия в едином информационном пространстве предприятия (ЕИПП) в значительной степени определяют работу таких модулей, как перспективное и оперативное планирование выпуска готовой продукции, материально-техническое снабжение, расчет плановой и фактической себестоимости и др.

Несмотря на то, что стандарты ISO 10303 предлагают способы представления информации об изделии, реализация конкретных проектов требует дополнительно решения практических задач по представлению структуры изделий в ЕИПП.

Разработке математического описания состава и свойств изделий посвящено значительное число публикаций. В работе [1] предлагается использовать полихроматические множества для описания свойств изделий, в работе [2] представлены способы структурного синтеза на основе различных графовых моделей, в работе [3] рассмотрено представление и проектирование семейства изделий. Эти работы носят фундаментальный характер и не дают конкретных рекомендаций по представлению состава изделий в ЕИПП. В данной работе предлагаются способы представления групповой спецификации в ЕИПП, созданном на основе реляционной базы данных.

Классический способ представления спецификации

ER-модель базы данных, когда для каждой сборочной единицы составляется своя спецификация, состоит из таблицы изделий и таблицы спецификаций (рис. 1). Таблица изделий содержит сборочные единицы и детали как покупные, так и изготавливаемые на предприятии. В таблице спецификаций поле "ID_изделия_родителя" представляет изделие, для которого составлена спецификация (куда входит изделие потомок), "количество" — определяет число входящих изделий.

Групповая спецификация

Если номенклатура типоразмеров изделий, выпускаемых предприятием, большая, и изделия сгруппированы по типам, причем изделия каждого типа имеют множество одинаковых деталей, то в целях уменьшения объема информации составляется групповая спецификация. В групповой спецификации имеется список деталей и сборочных единиц, входящих во все изделия группы (постоянные детали), и список деталей и сборочных единиц, входящих в отдельное изделие группы (переменные детали). Например, если в группу входит 10 изделий, каждое из которых содержит 100 деталей, причем 90 деталей входят во все изделия, то общее число записей — 190. При составлении спецификации на каждое изделие общее число записей — 1000.

ER-модель базы данных для групповой спецификации представлена на рис. 2. Здесь таблица "Групповые спецификации" содержит типы изделий, например, Насос НПЦ-32, Редуктор МРВ-2. Таблица "Изделия спецификаций" содержит конкретные изделия, входящие в группу, заданную полем "ID_Групповой_спецификации". Для насоса НПЦ-32 — это исполнения НПЦ.00.000, НПЦ.00.000-01, НПЦ.00.000-02, НПЦ.00.000-03 и т. д. Таблица "Содержание_спецификации" содержит перечень деталей и сборочных единиц, из

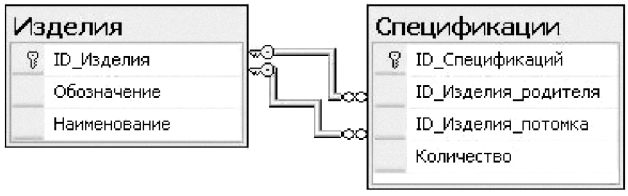


Рис. 1. ER-модель классической базы спецификаций

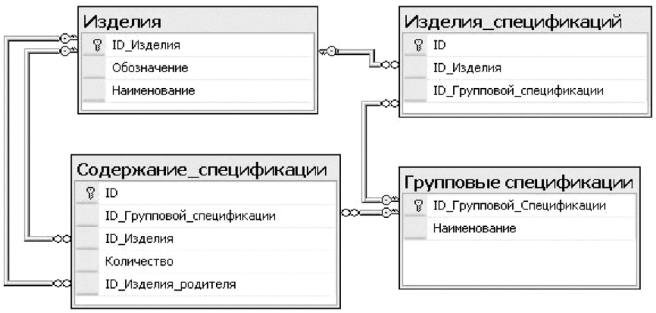


Рис. 2. ER-модель базы данных групповых спецификаций

которых состоят изделия из таблицы "Изделия_спецификаций". Если поле "ID_Изделия_родителя" не определено или ноль, то деталь постоянная, в противном случае деталь принадлежит изделию, заданному этим полем.

Групповая спецификация с полем принадлежности

Рассмотренный способ составления групповой спецификации эффективен тогда, когда число изделий группы небольшое. Если, например, число изделий группы 100 и каждое изделие состоит из 100 деталей, и 90 деталей постоянных, то общее число записей групповой спецификации 1090. Между тем, оставшиеся переменные детали могут встречаться в нескольких изделиях. Например, для группы изделий, имеющих конструкторское обозначение 700.100.01, 700.100.02, ..., 700.100.50, 700.200.01, 700.200.02, ..., 700.200.50, групповая спецификация может иметь следующий вид (табл. 1):

Таблица 1

Групповая спецификация

Обозначение	Наименование	Количество
Постоянные детали		
700.400.01	Шестерня	2
700.500.00	Хвостовик	1
...	...	...
Переменные детали		
Детали, входящие в изделия 700.100.01, 700.100.02,..., 700.100.10		
700.300.01	Крышка	1
700.310.01	Прокладка	2
...	...	...
Детали, входящие в изделия 700.100.11, 700.100.12,..., 700.100.20		
700.300.02	Крышка	1
700.310.02	Прокладка	2
...	...	...

В автоматизированной системе подготовки конструкторской документации предлагается в стандартную спецификацию добавить новую графу, которая будет определять принадлежность детали к тому или иному изделию. Запись в эту графу осуществляется по следующим правилам.

Сначала каждой лексеме обозначения изделия даются имена. Например, Z1, Z2 и т. д. Тогда обозначения изделий 700.100.01, 700.100.02, ..., 700.100.50 записываются как Z1.Z2.Z3. Соответственно запись "0 < Z3 < 11 и Z2 = 100" обозначает изделия, у которых второе поле обозначения — 100, а третье больше нуля и меньше 11, т. е. изделия 700.100.01, 700.100.02, ..., 700.100.10. Запись "Z3 = 11 или Z3 = 15" обозначает изделия 700.100.11, 700.100.15, 700.200.11, 700.200.15. Таким образом, спецификация представленная выше, примет вид (табл. 2):

Таблица 2

Групповая спецификация с графой принадлежности

Обозначение	Наименование	Количество	Принадлежность
700.400.01	Шестерня	2	Постоянная
700.500.00	Хвостовик	1	Постоянная
...	...	...	...
700.300.01	Крышка	1	0 < Z3 < 11 и Z2 = 100
700.310.01	Прокладка	2	0 < Z3 < 11 и Z2 = 100
...	...	...	...
700.300.02	Крышка	1	10 < Z3 < 21 и Z2 = 100
700.310.02	Прокладка	2	10 < Z3 < 21 и Z2 = 100
...	...	...	...

На первый взгляд составление спецификации в таком виде может показаться трудоемким, однако, эта спецификация не предназначена для ручного ввода. Ввод поля принадлежности в автоматизированной системе при наличии операции копирования или ввода по шаблону осуществляется нажатием всего одной клавиши. ER-модель базы данных в этом случае принимает вид рис. 3.

Предложенное представление групповой спецификации включает в себя описанные выше первые два способа и позволяет значительно сократить объем вводимой информации и унифицировать программный код АСУП.

Спецификация изделия в АСУП используется в алгоритмах разузлования при нормировании материалов, планировании работы цехов и т. д. Недостатком предложенного способа задания спецификации является отсутствие "ID_Изде-

Рис. 3. ER-модель базы данных групповых спецификаций с полем принадлежности

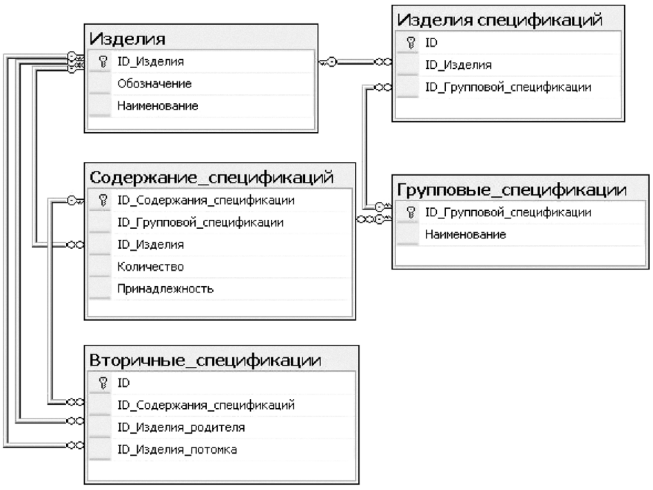


Рис. 4. ER-модель базы данных групповых спецификаций с полем принадлежности и вторичной спецификацией

лия_Родителя", что, несомненно, увеличит время работы алгоритма разузлования.

Предложенный метод был использован в автоматизированной системе технологической подготовки производства ЗАО "Тамбовский завод Полимермаш". Трехлетняя эксплуатация показала, что для редукторного производства (число деталей в изделиях порядка 100) замедление работы алгоритма разузлования не ощущается. База данных реализована в MS-SQL 2005, алгоритм разузлования — в виде хранимой процедуры. Для больших изделий (с числом деталей несколько тысяч) можно использовать дополнительно таблицу "Вторичные спецификации" (рис. 4).

Записи этой таблицы создаются программно из таблиц "Содержание_спецификаций" и "Изделия_спецификаций" при вводе (удалении, редактировании) записей в таблицу "Содержание_спецификаций". Таблица "Вторичные_спецификации" содержит "ID_Изделия_родителя" и алгоритм разузлования может быть применен к ней. Поле "ID_Содержание_спецификаций" позволяет автоматически поддерживать таблицу "Вторичные_спецификации" через таблицу "Содержание_спецификаций".

Заключение

Предложенный способ представления структуры изделий в реляционной базе данных может быть использован при разработке систем автоматизированной подготовки конструкторско-технологической документации и нормирования. Автором он использован при разработке автоматизированной системы нормирования для ЗАО "Тамбовский завод Полимермаш". Эффективность доказана трехлетней эксплуатацией.

Список литературы

1. **Павлов, В. В.** Структурное моделирование в CALS-технологиях/ В. В. Павлов; [отв. ред. Ю. М. Соломенцев]; Ин-т конструкторско-технологической информатики РАН. — М.: Наука, 2006. — 307 с.

2. **Иванова, Г. С.** Способы представления структурных моделей. / Г. С. Иванова // Наука и образование Эл. № ФС 77-30569, № Гос. регистрации 0420800025. — 2007. — № 1. http://technomag.edu.ru:8001/db/msg/30983.html.

3. **Третьяков, В. М.** Основы методики проектирования семейства изделий / В. М. Третьяков // Автоматизация и современные технологии. — 2004. — № 2. — С. 25—33.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

УДК 004.04’23

С. М. Авдошин, канд. техн. наук, зав. каф.,
Государственный Университет — Высшая школа экономики,
Е. Ю. Песоцкая, менеджер, Компания Accenture

Информационные технологии управления рисками программных проектов

На основе обзора существующих систем управления рисками рассматриваются подходы к выбору системы управления рисками программных проектов.

Ключевые слова: *риск, неопределенность, управление рисками, программный проект, информационные технологии.*

Введение

Сегодня информационные технологии (ИТ) являются одним из важнейших элементов деятельности компаний. Руководители предприятий понимают необходимость внедрения информационных систем и четко представляют себе конкурентные преимущества, которые они получат, развивая свой бизнес на основе современных ИТ.

Исследования Gartner Group, Jnc. показывают, что к 2012 г. предприятия будут иметь дело с объемами данных, в 30 раз превышающими объемы данных в 2002 г. [1]. В условиях возрастающей сложности и масштабности программных проектов отмечается готовность предприятий тратить немалые деньги на их реализацию. Как правило, в процессе реализации программных проектов руководители сталкиваются с многочисленными рисками, т. е. вероятностными событиями, связанными с неопределенностью, которые негативно влияют на проект [2]. К таким негативным влияниям можно отнести задержки в графике работ проекта, превышение бюджета, несоответствие требуемым стандартам качества или ожиданиям заказчика и др. Поэтому процесс управления рисками программных проектов необходим для их успешной реализации.

Необходимость управления рисками в программных проектах

Программные проекты — один из наиболее сложных типов проектов для управления. Для программных проектов характерен высокий уровень сложности, неопределенность при планировании, необходимость координации работ отдельных сотрудников и подразделений, возможность возникновения конфликтов между менеджером проекта, высшим руководством, руководителями вовлеченных в проект подразделений и персоналом предприятия.

Независимо от величины и масштабов программного проекта руководство компании, ставя определенные цели, постоянно сталкивается с соответствующими управленческими проблемами: как спланировать работы во времени и успеть к определенному сроку; какие потребуются ресурсы и как добиться качественного выполнения работ в рамках бюджета. Обычно информация, используемая для управления программными проектами, доступна из разрозненных источников, таких, как, например, приложения, обеспечивающие отчетность из систем мониторинга ИТ-инфраструктуры, приложения систем класса Service Desk, OSS-систем, систем управления проектами и т. д. Кроме того, такая информация зачастую представляет собой технические данные, которые еще необходимо преобразовать в бизнес-информацию, чтобы они могли быть поняты и использованы заинтересованными сторонами. Однако для многих организаций усилия по интеграции существующих инструментов, систем и источников данных для получения управленческой информации неосуществимы как с точки зрения ресурсов, так и капитальных вложений.

Проекты в специфических предметных областях, таких как ИТ, проекты, осуществляемые с применением узкоспециальных технологий, а также проекты со специфическим конечным продуктом могут содержать риски, уникальные для своей области.

Выбор классификации рисков будет зависеть от специфики программного проекта и профессиональных предпочтений менеджера проекта. Оговоримся, что одни и те же риски могут немного отличаться по содержанию для разных видов деятельности и разных типов проектов. Но в основном, по мнению авторов, риски программных проектов можно классифицировать следующим образом:

Технические риски. Практически в любом программном проекте существуют риски, связанные с техникой: отказ и сбои в работе оборудования, ошибки в монтаже и т. п.

Риски оценки сроков. Для большинства проектов по разработке и внедрению программного обеспечения характерны ошибки в оценках сроков работ проекта.

Интеграционные риски. Интеграционные риски в программных проектах, особенно в крупных компаниях, всегда высоки, поскольку любое программное решение должно быть интегрировано в существующую инфраструктуру. Наиболее характерны риски перехода на новую систему, которые включают в себя расходы на остановку предприятия во время внедрения программных решений, обучение персонала и т. д.

Риски непринятия программного продукта пользователями. Любой проект, в том числе в ИТ сфере — это, в первую очередь, изменение технологии работы. Техническая составляющая любого проекта, безусловно, важна, но не менее важна организационная часть.

Коммерческие риски. Это риски, связанные с выбором технологии и поставщика. Необходимо оценить успешность технологии на рынке, ее актуальность на протяжении жизненного цикла программного проекта, доступность необходимого аппаратного и программного обеспечения, его качество, частоту модернизации.

Риски несоблюдения технологии. Эти риски возникают в случае, если менеджер проекта имеет единоличное решение по рискам (идентификация, анализ, выбор метода реагирования). Чем больше и сложнее проект, тем выше данный риск.

Риски информационной безопасности. Это риски, связанные с недостатками в системе мер защиты информации программного продукта, несоответствия рекомендациям международного стандарта безопасности ISO 17799, а также недостатков в программно-аппаратном обеспечении системы.

Хотя ни один пакет для управления рисками не обеспечит вместо вас хорошее планирование и управление рисками всех категорий, использование встроенных функций математического моделирования, средства планирования реагирования и мониторинга изменений увеличат шансы завершения проекта в срок и в рамках бюджета за счет снижения неопределенности и повышения качества управления рисками. Технологии управления рисками позволяют использовать различные методы идентификации и оценки рисков. Спектр методик количественного анализа широк: от *Pert*-анализа и анализа "Что-Если" до сложных вычислений *Monte Carlo*, методов *Event-Tree-Analysis*, цепей Маркова и др. [2].

Программное обеспечение по управлению рисками программных проектов

При желании можно найти программные пакеты, реализующие те или иные средства управления рисками. Гораздо сложнее подобрать комплексную систему управления рисками, которая бы специализировалась на области информационных технологий и могла бы отслеживать риски проекта и контролировать проект.

В настоящее время существует большое число систем, так или иначе реализующих функции управления рисками. Некоторые из них представляют собой информационные системы управления проектами, в которых присутствует модуль управления рисками, другие являются приложениями к системам календарного планирования либо самостоятельными программными продуктами по управлению рисками. Третьи специализируются на рисках информационной безопасности, например ISO17799, а потому позволяют определить не уровень рисков программных проектов, а степень соответствия тому или иному стандарту [3].

В качестве продукта поддержки процессов управления рисками при реализации программных проектов может использоваться как специализированная система, так и модуль управления рисками многофункциональной системы управления проектами (например, *Microsoft Project, Open Plan, Primavera* и др.) или модуль системы управления предприятием (например, *Oracle, SAP, Microsoft Axapta* и др.). Также возможно создание собственной разработки в области управления ИТ-рисками программных проектов.

Построение собственной системы управления ИТ-рисками является более сложной задачей, чем выбор существующего пакетного решения, и требует не только хороших теоретических знаний в области методологии управления рисками, но и практического опыта внедрения. Следует заранее предпринять действия, чтобы не допустить типичных ошибок, которые состоят в отсутствии доверия к полученным результатам оценки ИТ-рисков со стороны руководства, недостаточной обоснованности расходов на снижение рисков, а также в сопротивлении внедрению мер снижения рисков в бизнес-подразделениях и технических службах.

При принятии решения об использовании пакетного (готового) решения в области управления рискам программных проектов следует учесть некоторые ограничения, а именно:

- возможную сложность интеграции с существующими приложениями в случае нестандартной базовой ИТ;
- возможные трудности с самостоятельным выбором подходящего поставщика и формированием объективных критериев;
- возможность избыточной функциональности и неудобство программного интерфейса;
- необходимость выполнения пользовательских настроек и проблемы локализации.

Выбор системы управления рисками

Процесс выбора поставщика программного решения по управлению рисками, как правило, происходит в соответствии со следующими основными этапами: обзор рынка; формирование требований к системе и подготовка информационного запроса поставщикам (при необходимости); рассылка информационного запроса и анализ ответов поставщиков программного обеспечения (ПО); выбор системы.

Критериями выбора системы являются следующие категории требований:

- информация о поставщике и продукте;
- функциональные требования;
- технические требования;
- цены и условия.

Информация о поставщике и продукте содержит контактную информацию о поставщике, информацию об опыте работы и финансовых показателях, масштабах деятельности, географическом охвате и направлении деятельности компании поставщика, о стратегических партнерах.

Функциональные требования содержат требования к функциональности системы управления рисками в сфере ИТ. Они содержат описание возможностей настройки данной функциональности в системе (например, возможности настраивать поля, формулы, оформление графиков, шаблонов отчетов и пр.).

Технические требования содержат критерии к аппаратному и программному обеспечению, описание конфигурации, необходимой для полнофункциональной работы информационной системы управления рисками.

Цены и условия содержат оценку стоимости покупки, внедрения и поддержки системы управления рисками с учетом приведенного примера числа пользователей.

При выборе систем следует учитывать широту охвата основных процессов управления рисками: идентификацию рисков, оценку (качественную и количественную), выбор методов реагирования, мониторинг и контроль противорисковых мероприятий.

Следует обратить особое внимание на функциональность обновления значений рисков, мониторинга эффективности используемых способов управления и способов управления остаточными рисками (например, перерасчет максимально допустимых значений рисков; процесс реагирования на инциденты и др.), качества процесса реагирования на риски. В качестве достоинства — инструмент оценки ИТ-рисков программных проектов может, например, позволить отследить связи между выявленными рисками и причинами, которые ведут к ним [4].

Обзор систем управления рисками в области программных проектов и ИТ

В качестве примера систем управления рисками в области программных проектов и ИТ остановимся более детально на нескольких наиболее актуальных, по нашему мнению, системах.

IBM Rational Portfolio Manager [5]

Система полностью поддерживает методологию COBIT® [6, 7] и предоставляет инфраструктуру для введения элементов управления, которые помогут обеспечить соответствие законодательным нормативам, таким, как закон Сарбейнса—Оксли [8, 5] и соглашение Basel II [5], и устранить разрыв между потребностями управления, техническими проблемами, бизнес-рисками и требованиями показателей производительности и реализацией инструментов финансового контроля.

По данным компании IBM *Portfolio Manager* осуществляет поддержку процессов идентификации и контроля рисков, позволяет выявить и уменьшить самые высокие воздействия рисков путем задания пользовательского триггера, который уведомит вас о возникновении риска. Можно предоставить показатели, которые покажут, например, насколько хорошо выполняется снижение риска, периодически фиксируя число событий триггера, интенсивность событий триггера, сколько времени потребовалось на разрешение события и какой уровень эскалации потребовался.

OCTAVE-S (Operationally Critical Threat, Asset and Vulnerability Evaluation) [9]

Система спроектирована на основе методологии информационной безопасности *Octave* и предусматривает высокую степень гибкости, достигаемую путем выбора критериев, которые предприятие может использовать при адаптации под собственные нужды, и позволяет поддерживать следующие процессы управления рисками:

- идентификацию критичных информационных активов;
- идентификацию угроз для критичных информационных активов;
- определение уязвимостей, ассоциированных с критичными информационными активами;
- оценку рисков, связанных с критичными информационными активами.

Risk Watch, RiskWatch, Inc [10]

Программное обеспечение специализируется на анализе и контроле информационных рисков и является мощным средством анализа и управления рисками. В семейство *RiskWatch* входят программные продукты для проведения различных видов аудита безопасности.

В качестве критериев для оценки и управления рисками используются "предсказание годовых потерь" (*Annual Loss Expectancy* — ALE) и оценка "возврата от инвестиций" (Return on Investment — ROI). В части оценки рисков система позволяет устанавливать связи между ресурсами, потерями, угрозами и уязвимостями, рассчитывать математические ожидания потерь с учетом частоты возникновения угрозы риска и стоимость ресурса, который подвергается риску.

CRAMM (CCTA Risk Analysis and Management Method) [11, 12]

Система *CRAMM* создана на основе методологии управления рисками *CRAMM* и предполагает использование технологий оценки угроз и уязвимостей по косвенным факторам с возможностью проверки результатов. В нее заложен механизм моделирования информационных систем с позиции безопасности с помощью обширной базы данных по контрмерам. *CRAMM* нацелена на детальную оценку как самих рисков, так и эффективности применяемых комбинаций методов управления ими.

Помимо анализа рисков *CRAMM* позволяет решать ряд дополнительных задач, включая:

- проведение обследования используемой/внедряемой ИТ и выпуск сопроводительной документации на всех этапах его проведения;
- проведение аудита на основе стандарта безопасности информации BS 7799:1995 (*Code of Practice for Information Security Management*);
- разработку политики безопасности и плана обеспечения непрерывности бизнеса.

КОНДОР+ [13]

Программный продукт позволяет менеджерам ИТ-проектов проверить политику информацион-

Сравнительный анализ систем по управлению рисками в области программных проектов

Критерии	IBM Rational Portfolio Manager	OCTAVE-S	Risk Watch	CRAMM	КОНДОР+	Risk Management (CEISOQ)
Учитывает специфику программных проектов	×	×	×	×	×	×
Поддерживает методологию управления рисками	×	×		×		
Идентификация (классификаторы) риска	×			×	×	×
Количественная оценка на основе моделирования	×	×		×		×
Возможности моделирования реагирования на риск			×	×		×
Гибкие настраиваемые отчеты		×		×	×	×
Поддержка информационной безопасности	×	×	×		×	×
Поддержка русского языка в системе	×				×	

ной безопасности компании на соответствие требованиям ISO 17799. *КОНДОР+* включает в себя более 200 вопросов, ответив на которые специалист получает подробный отчет о состоянии существующей политики безопасности, а также модуль оценки уровня рисков соответствия требованиям ISO 17799.

В отчете отражаются все положения политики безопасности, которые соответствуют и не соответствуют стандарту, а также существующий уровень риска невыполнения требований политики безопасности в соответствии со стандартом. Элементам, которые не выполняются, даются комментарии и рекомендации экспертов. По желанию специалиста, работающего с программой, могут быть выбраны генерация отчета, например, по какому-то одному или нескольким разделам стандарта ISO 17799, общий подробный отчет с комментариями, общий отчет о состоянии политики безопасности без комментариев для представления руководству.

Risk Management, приложение в составе *CEISOQ* (*Modeling Software Complex for Evaluation of Information Systems Operation-Quality*) [14].

Приложение *Risk Management* поддерживает задачи, решаемые с применением математических методов и моделей, в жизненном цикле системы согласно стандартам ISO/IEC, в том числе в процессе разработки проекта и системы, при проведении контроля, аудита и сертификации.

Risk Management помогает менеджеру в определении событий, которые отрицательно влияют на систему, классификации рисков, определении методов формализованной оценки рисков в терминах и показателях качества, затрат, сроков или технических характеристик [14].

Наглядные графики и дружественный пользовательский интерфейс системы *Risk Management* делают работу с системой удобной и доступной практически любому пользователю. На рис. 1 (см. четвертую сторону обложки) показаны результаты моделирования вероятности наступления различных видов рисков во временном периоде. На рис. 2 (см. четвертую сторону обложки) приведена статистическая оценка эффективности противорисковых мероприятий для заданного вида риска.

Рассмотрим сравнительные характеристики специализированных систем в области управления рисками для реализации программных проектов, представленные в таблице.

К наиболее привлекательным системам можно отнести *IBM Rational Portfolio Manager, CRAMM и Risk Management* (*CEISOQ*). Помимо перечисленных систем на российском рынке также присутствуют такие многофункциональные системы в области управления рисками, как *@Risk Professional for Project*, *Dekker TRAKKER*, *Enterprise project, ER Project 1000, Intelligent Planner, Mesa/Vista Risk Manager, Risk Track, Open Plan* [15]. Эти программные продукты специализируются больше на проектных рисках и затрагивают аспекты информационной безопасности в наименьшей степени.

Заключение

Использование информационных технологий является сегодня обязательным условием для эффективного управления промышленным предприятием и повышения его конкурентоспособности. Переход на другой качественный уровень работы с информацией и автоматизация деятельности с помощью внедрения программных проектов представляет собой достаточно трудоемкий и болезненный процесс, сопровождающийся множеством рисков и непредвиденных ситуаций.

До принятия решения о внедрении той или иной системы управления ИТ-рисками программных проектов следует убедиться, что она достаточно полно учитывает бизнес-потребности компании, ее масштабы, а также соответствует лучшим мировым практикам и имеет достаточно подробное описание процессов и требуемых действий.

Качественная система управления рисками позволит менеджеру программного проекта принимать более обоснованные управленческие решения на основе количественных данных и обеспечить лучшее взаимодействие команды проекта.

Список литературы

1. **Gartner Group, Inc.** Home Page — Gartner Group, Inc. www4.gartner.com.
2. **Рогов М. А.** Риск-менеджмент. М.: Финансы и статистика, 2001. 120 с.
3. **Guide** for developing security plans for information technology systems // NIST Special Publication. 2000. 800-18.
4. **IBM, Inc.** Home Page: www.ibm.com. Описание системы IBM Rational Portfolio Manager.
5. **Standards** for Information Systems Auditing. — ISACA Standards, 2000.
6. **CobiT:** Control Objectives. — ISACA, 3nd Edition, 2000.
8. **Sarbanes-Oxley.** Законодательный Акт об управлении предприятием, принятый в США в 2002 г. Интернет-ресурс www.sarbanes-oxley.com.
9. **Описание системы Octave.** www.cert.org/octave.
10. **Описание системы Risk Watch.** www.riskwatch.org/.
11. **A practical** guide to the use of CRAMM, 2003.
12. **Описание системы Cramm.** www.cramm.com/.
13. **Описание системы Кондор+.** www.sec4all.net/kondor.html.
14. **Костогрызов А. И.** 100 Математических моделей для эффективного управления качеством вооружения и военной техники (ВВТ) в контексте требований системообразующих стандартов. Бюллетень "Менеджмент. Вооружение. Качество", Спец. выпуск. 2005. С. 18.
15. **Дубовик М., Песоцкая Е.** Можно ли автоматизировать процесс управления рисками. www.projectmanagement.ru/mup.asp?mupid = 33.

УДК 519.685

С. В. Востокин, д-р техн. наук, доц.,
Самарский государственный аэрокосмический университет имени академика С. П. Королева

Технология интеграции приложений на основе визуальной модели GraphPlus

Рассмотрены методы интеграции приложений с точки зрения размещения процессов во взаимодействующих подсистемах. Описан способ интеграции с использованием процесса, одновременно исполняющегося в связываемых системах. Представлены возможности информационной технологии визуального программирования на основе модели GraphPlus для автоматизации данного способа интеграции.

Ключевые слова: *визуальная компонентная модель, распределенные вычисления, интероперабельность, каркас приложения.*

Введение

Цель современных крупномасштабных проектов информатизации — это не только разработка отдельных приложений, но и разработка технологий, позволяющих объединять вновь создаваемые и уже существующие приложения для совместного решения общих задач.

До недавнего времени такие проекты в основном разрабатывались в целях автоматизации научных исследований — это грид-вычисления на основе программного обеспечения GTK (Globus Alliance) [1], gLite (CERN) [2], ARC (NorduGrid) [3] и других инструментов. Однако в ближайшее время ожидается переход к интенсивному промышленному использованию подобных технологий. Об этом свидетельствует выступление одного из лидеров индустрии Б. Гейтса на недавно прошедшем Consumer Electronics Show 2008 [4]. В нем говорится о планах Microsoft по разработке платформы, с помощью которой можно будет строить приложения на основе разделения клиента, сервера и "облака" — специальной интегрирующей среды исполнения. Данная платформа под кодовым названием Oslo была анонсирована в пресс-релизах Microsoft в октябре 2007 г., а ее представление в новых версиях средств разработки (Microsoft Visual Studio "10", Microsoft System Center "5", BizTalk Server "6", BizTalk Services "1", Microsoft .NET Framework "4") ожидается в 2008 г. [5].

Новые методы программирования для интегрирующей среды исполнения ("облака" в терминах Microsoft) предусматривают использование моделирования как основы построения современных приложений. Исследования аналогичного назначения проводились в СГАУ в 2004—2007 гг. в рамках проекта GraphPlus [6—11]. Целью проекта являлась разработка и обоснование свойств компактной визуальной модели параллельных и распределенных вычислений в контексте программирования численных методов. В статье описывается, каким образом данная модель может быть использована в качестве технологии межсистемной интеграции.

Архитектуры интеграции приложений

В чем же конкретно заключаются новые подходы в интеграции программных систем и их реализация в предлагаемой модели GraphPlus? Основой интеграции гетерогенных приложений является сеть передачи данных и базовый стек протоколов (де-факто протокол TCP/IP) [12]. Взаимодействие программ-клиентов и программ-серверов Internet в контексте типовых служб происходит на основе протоколов (HTTP, SMTP,

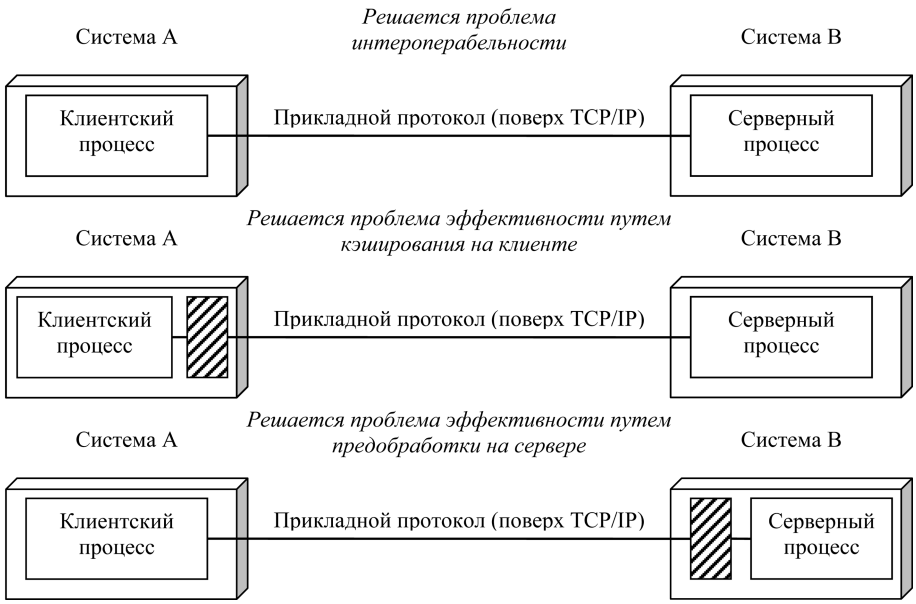


Рис. 1. Традиционные схемы интеграции приложений

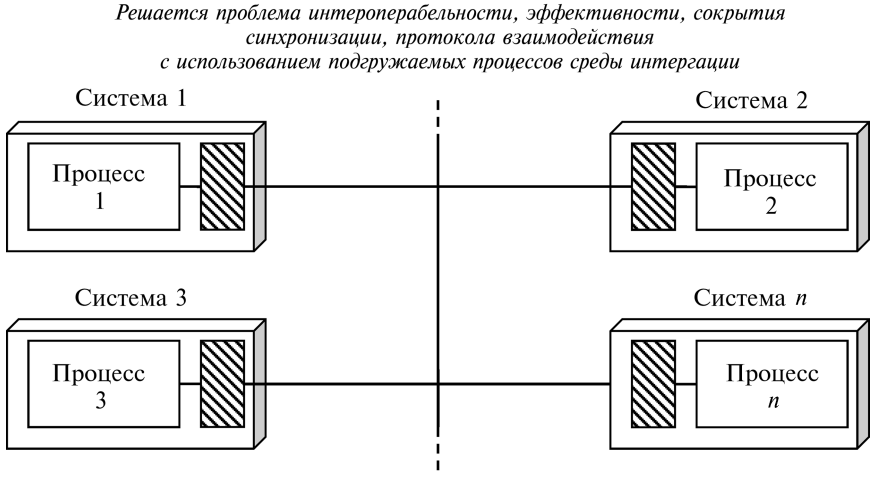


Рис. 2. Новая схема интеграции приложений

FTP и др.), реализованных поверх транспортных протоколов. Однако построение стандарта интеграции прикладных программ по аналогии с проектированием типовых служб Internet является неоправданно трудоемким как с точки зрения описания протокола, так и с точки зрения дальнейшего программирования на его основе. Поэтому были разработаны технологии (DCE PRC, CORBA, RMI, DCOM и др.), использующие модель удаленного вызова, которая представляет посылку запроса и получение ответа с сервера как вызов процедуры в адресном пространстве клиента. В настоящее время также используется подход, в котором сообщения строятся в расширяемом формате XML. Дальнейшее развитие технологий интеграции было направлено на решение проблем эффективности при использовании моделей обмена сообщениями и RPC. Для ускорения доступа к ресурсам на удаленных серверах часто бывает полезно выполнять сложное кэширование данных на стороне клиента. Это требует загрузки кода сервера на сторону клиента. Вместе с тем, для сложного манипулирования данными при подготовке и выполнении запросов необходимо, наоборот, загружать код клиента на сервер (рис. 1). Примерами использования технологий загрузки кода в адресное пространство клиента является Ajax-технологии, а в адресное пространство сервера — хранимые процедуры в языке SQL.

Таким образом, требование эффективности и гибкости межоперационного взаимодействия обусловливает новую архитектуру распределенной программной среды, в которой программные компоненты не только взаимодействуют между собой, но могут также свободно перемещаться между адресными пространствами отдельных компьютеров и устройств. Эта архитектура активно исследовалась в течение последних десяти лет, а результатом являются такие средства, как GTK, Jini, Globe и другие системы. Схема интеграции в этом подходе (рис. 2) реализуется совокупностью процессов, работающих от имени одного пользователя в адресных пространствах связываемых систем, которые взаимодействуют между собой и обеспечивают совместное поведение единой системы. При этом необходимые связывающие процессы подгружаются динамически и не являются частями связываемых систем. Эти процессы маскируют как логику связи и синхронизации, так и используемый метод передачи информации (форматы и протоколы обмена).

Результаты проекта GraphPlus

Современные языки и средства промышленного программирования не поддерживают напрямую разработку интегрирующего кода в описанной архитектуре (рис. 2). Проведенные исследования в рамках проекта GraphPlus были направлены на разработку и обоснование методов программирования для архитектуры нового типа. В ходе исследований получен ряд практически значимых результатов.

Предложена и формально описана вычислительная модель взаимодействия компонентов. Модель сочетает возможности наглядной интерпретации, понятной программисту, и строгой записи с использованием темпоральной (временной) логики, исключающей двусмысленности в толковании модели. Модель является объектной и основана на представлении вычислений в форме обхода сетевой структуры, образованной объектами P-типа (подобие процессов) одновременно несколькими объектами M-типа (подобие сообщений). При построении модели не имеет значения, в адресных пространствах каких машин будут размещаться объекты. Более того, объекты могут менять свое положение в адресных пространствах (для переноса данных, балансировки загруженности, отказоустойчивости), что невозможно обнаружить с точки зрения выполняемого алгоритма. Объектам модели среда исполнения представляется как один многопроцессорный компьютер с общей (разделяемой) памятью. Модель относится к классу асинхронных моделей. Она оптимизирована для совмещения обменов и вычислений в целях эффективной реализуемости в распределенной среде (рис. 3). На рисунке объекты P-типа обозначены прямоугольниками, а объекты M-типа — кружками. Показана реализация взаимодействия на простейшем фрагменте сетевой структуры.

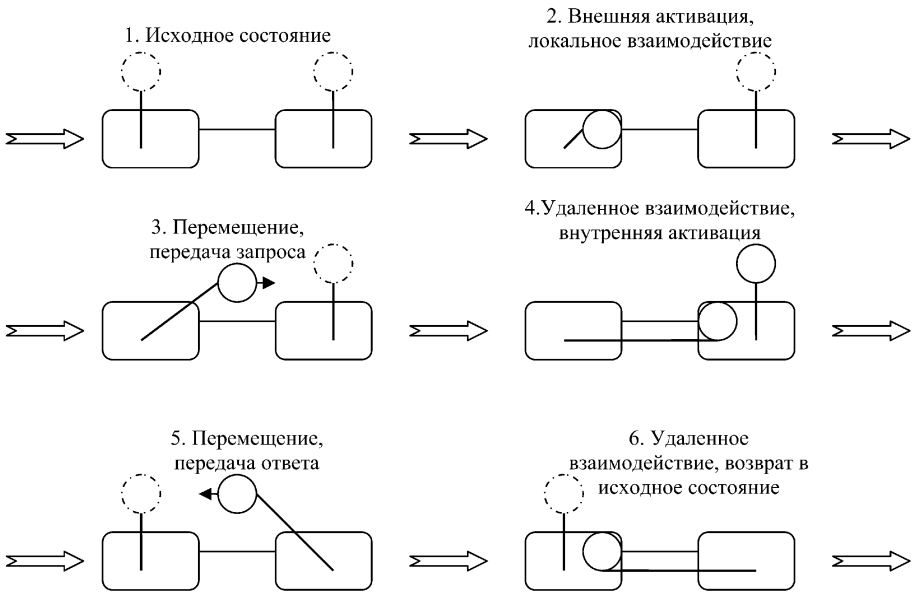


Рис. 3. Представление вычислений в модели GraphPlus

Предложен метод визуального описания модели. Для программиста удобно понимать и отлаживать программу, если объект выглядит как совокупность данных и методов их обработки. Детали, связанные с управлением вычислениями, синхронизацией, особенностями исполнения модели, логично отделить от прикладного кода. В итоге прикладной код состоит только из методов с традиционной процедурной семантикой, понятной любому программисту. Для достижения этого собственно и служит метод визуализации. В нем используются ориентированные размеченные графы специального вида, для представления которых определен набор пиктографических элементов. Отдельный граф описывает компонент и его связи с другими соседними компонентами в сетевой структуре. Для графов определена семантика в терминах введенной модели вычислений. Визуальное описание компонента не зависит от реализации, конструируется в специальном графическом редакторе и хранится в формате XML. Пример визуального представления логики компонента представлен на рис. 4. На нем кружки с символьными пометками обозначают вызовы методов компонента; сплошные дуги — методы, возвращающие логические значения, определяющие передачу управления; штриховые дуги — локальные M-объекты и их активация; двойные кружки — синхронизация.

Разработан метод генерации кода компонента (рис. 5) по его визуальному представлению. Результатом генерации является специальный код (подобъект-адаптер), являющийся частью компонента, связывающий предметно-ориентированную часть его P-объектов (подобъектов семантики) и системную часть (подобъекты среды исполнения). Каждый M-объект модели, выполняющий обход сетевой структуры, агрегирован в компонент и представлен на визуальной модели. Такой объект состоит из своего подобъекта семантики и подобъекта среды исполнения. Связь

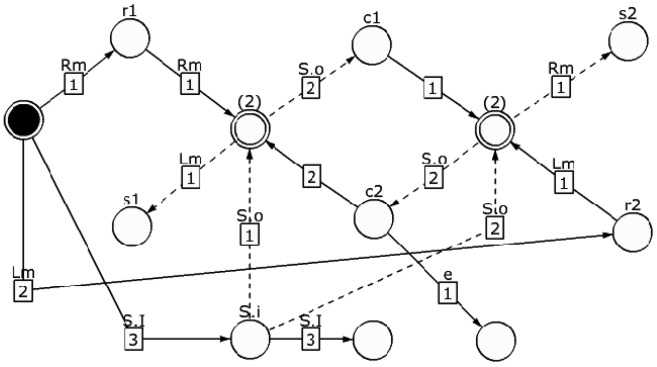


Рис. 4. Пример визуального представления компонента в модели GraphPlus, построенный в специализированном редакторе

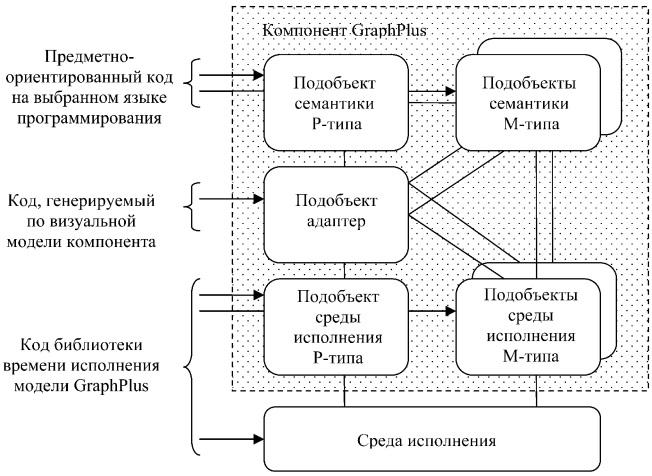


Рис. 5. Структура кода компонента в модели GraphPlus

объектов модели двух типов реализуется в подобъекте-адаптере, генерируемом по визуальной модели компонента.

Описан, программно реализован и протестирован базовый механизм исполнения модели в многопоточной среде современных многопроцессорных/многоядерных компьютеров. На его основе легко строится распределенный механизм путем добавления в среду исполнения универсальных объектов-заместителей. В отличие от традиционных методов реализации компонентных технологий, основанных на паттерне "Активный Объект" [12], предложенный метод ориентирован на максимально эффективное исполнение в случае, когда много (порядка сотен) компонентов одновременно загружены на одну машину. Это необходимо для балансировки загруженности в традиционных алгоритмах, а также может применяться в перспективных мультиагентных технологиях как операционная среда систем агентов.

В ходе реализации и тестирования описываемой технологии на примерах из области численного моделирования был апробирован процесс разработки надежных распределенных приложений для предложенной модели. Процесс предусматривает следующие этапы. Во-первых, выполняется *этап разработки и отладки эталонного последовательного теста* (*тестов*). За ним следует *этап построения визуальной модели, ее отладки в тестовой среде с последовательным исполнением*. Этот этап пройден, если результат работы эталонного теста совпадает с результатом исполнения модели. На *следующем этапе логика параллельного алгоритма тестируется более глубоко*, с учетом непредсказуемого протекания асинхронного вычислительного процесса (в пределах спецификации модели) при реальном исполнении. Как показала практика, надежность можно гарантировать при достижении большого числа повторений одного теста при условии, что модель исполняется в истинно многопоточной среде (под управлением нескольких потоков ядра операционной системы) и реальной мультипроцессорной архитектуры компьютера (гиперпоточной, многоядерной, многопроцессорной). Обычная многократная мультипрограммная реализация теста надежности не гарантирует.

Если целью проектирования является уменьшение времени счета при использовании нескольких процессоров, необходимо *исследовать модель на эффективность*. Это исследование, называемое отладкой производительности, выполняется следующим образом. Реализуется процедура, имитирующая вычислительную нагрузку в реальном алгоритме. Процедура добавляется в тест модели. Если планируется реализация модели только в пределах одной машины, достаточно выполнить измерения производительности штатными средствами операционной системы (среды программирования). Для анализа эффективности распределенных вычислений применяют специальную имитационную версию среды исполнения модели. Ее использование позволяет детально изучать источник возможных проблем производительности, измерять мгновенную интегральную эффективность вычислений, наблюдать ее изменения и влияние на нее случайных факторов среды исполнения. Такие измерения либо невозможны, либо трудно реализуемы на реальной распределенной системе (даже кластерной). На всех этапах исследований требуется проверять совпадение результатов с эталонным тестом и в случае возникновения ошибки повторить предыдущие этапы для ее исправления. Заметим, что по построению как генерируемый код подобъектов-адаптеров, так и код среды исполнения (по крайней мере, в последовательной версии для тестирования) легко отлаживать штатными средствами среды программирования. Наконец, в отлаженной модели тестовые версии подобъектов семантики заменяются рабочими версиями, и выполняется заключительное функциональное тестирование.

Как видно из описания процесса, модель, собранная со средой исполнения, но без конкретных определений подобъектов семантики, можно рассматривать как самостоятельный программный продукт. В ходе исследований для модели предложен формализованный объектно-ориентированный метод описания таких продуктов — каркасных библиотек. Эти библиотеки инкапсулируют реализацию типовых сценариев взаимодействия, пригодных как для вычислений, так и для целей интеграции. Метод апробирован на трех типовых процессах: процессе последовательно-параллельного взаимодействия; процессе "ведущий — ведомые"; асинхронном взаимодействии

линейки процессов (из области сеточных вычислений). Подобные каркасные библиотеки удобны тем, что можно ограничить свойства среды исполнения только необходимыми в данном контексте функциями. Тем самым сокращаются трудозатраты на кодирование, так как не нужно заранее разрабатывать универсальную версию среды исполнения, пригодную для всех возможных вариантов использования системы. Это актуально в силу изначально исследовательского характера описываемой технологии. В этом проект похож на известную систему Globe [11, 13].

Заключение

Проект GraphPlus выполнялся по программе научных исследований СГАУ. Теоретические и практические результаты использованы при выполнении госбюджетных работ в рамках персонального гранта Министерства образования Российской Федерации и Правительства Самарской области на проведение исследований в области гуманитарных, общественных, технических наук и естествознания; при выполнении исследовательских работ по гранту Американского фонда гражданских исследований и развития (CRDF Project SA-014-02). Основные компоненты программного комплекса официально зарегистрированы Федеральной службой по интеллектуальной собственности, патентам и товарным знакам (свидетельства об официальной регистрации № 2007611206, № 2007611201). Общий объем исходного кода данных компонентов превышает 8000 строк.

В настоящее время исследования продолжаются в двух направлениях: выполняется стандартизация технологии и проводится обоснование типовых проектных решений (каркасных библиотек) в области интеграции и управления вычислениями на базе данной технологии. Для этой цели разрабатываются и исследуются варианты реализации для платформ C/C++ для API Win32, .NET, Java. В качестве часто используемого типового решения в области параллельной и распределенной обработки детально исследуются процессы "ведущий — ведомые" и асинхронное взаимодействие линейки процессов в целях изучения метрик, предсказания производительности, удобства использования и т. д.

Несмотря на ожидаемый выпуск новых средств разработки, ориентированных на модели, мы надеемся, что в ближайшее время технология GraphPlus не потеряет своей актуальности как для исследовательских целей, так и для практического применения. В пользу этого говорит простота и компактность, гибкость и расширяемость, платформонезависимость, сочетаемость с известными технологиями программирования, ориентация на практическое применение, подтвержденные в ходе проведенных исследований.

Список литературы

1. **The Globus** Alliance. http://www.globus.org.
2. **gLite** — Lightweight Middleware for Grid Computing. http://glite.web.cern.ch/glite/
3. **NorduGrid.** http://www.nordugrid.org/
4. **Компьютерра-Онлайн**. CES 2008. http://www.computerra.ru/online/news/344678/
5. **Oslo.** http://www.microsoft.com/soa/products/oslo.aspx
6. **Официальный сайт** проекта Граф Плюс. 2004—2008. http://graphplus.ssau.ru
7. **Востокин С. В.** Особенности реализации процедур сканирования параметрических пространств для сложных численных моделей // Математическое моделирование. 2006. Т. 18. № 12. С. 125—128.
8. **Востокин С. В.** Объектно-ориентированный метод структурирования кода метакомпьютерного приложения // Информационные технологии. 2006. № 5. С. 40—45.
9. **Востокин С. В.** Технология визуального проектирования параллельных и распределенных приложений // Системы управления и информационные технологии. 2006. № 2 (24). С. 39—43.
10. **Востокин С. В.** Спецификация модели параллельных и распределенных вычислений GraphPlus на основе логики TLA // Известия Самарского научного центра РАН. 2006. Т. 8. № 3 (17). С. 875—881.
11. **Востокин С. В.** Графическая объектная модель параллельных процессов и ее применение в задачах численного моделирования. Самара: Изд-во Самарского научного центра РАН, 2007. 286 с.
12. **Таненбаум Э., ван Стеен М.** Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.
13. **Schmidt D. C., Stal M., Rohnert H., and Buschmann F.** Pattern-Oriented Software Architecture — Patterns for Concurrent and Networked Objects. Wiley, 2000.
14. **The GLOBE** Project. http://www.cs.vu.nl/~steen/globe/

УДК 621.26

Д. И. Иванов, аспирант,
И. А. Цикин, д-р техн. наук, проф., зав. каф.,
Санкт-Петербургский государственный
политехнический университет (СПбГПУ)

Реализация режима удаленного программирования в специализированной среде моделирования MATLAB

Рассматривается технология удаленного сетевого доступа к специализированной среде моделирования на примере программного комплекса MATLAB с реализацией режима удаленного программирования. Рассматриваются решения как на базе встроенных в среду весьма ограниченных средств, так и на базе специально разработанных дополнительных программных модулей.

Ключевые слова: *сетевой доступ, модель, MATLAB, MWS, WEB, HTML, HTTP, COM.*

В настоящее время как в научных исследованиях, так и при использовании современных технологий обучения широко применяются методы компьютерного моделирования физических процессов и явлений.

Эта эффективная процедура дает возможность целостного изучения поведения как реально существующих сложных систем, так и создаваемых для проверки теоретических гипотез.

Сетевая среда позволяет реализовать новые сервисы при решении задач компьютерного моделирования, что расширяет возможности в области коллективного использования моделей. Так, современные сетевые информационные технологии делают возможным реализацию моделей, требующих больших вычислительных ресурсов, без необходимости наличия мощного компьютера у конечного пользователя. При таком подходе модель централизованно размещается на сервере моделей, а на компьютере пользователя предполагается лишь установка клиентского программного обеспечения, в задачи которого входит организация взаимодействия пользователя с моделью посредством визуального интерфейса. При этом сама модель может быть исполнена с помощью различных сред проектирования, в том числе и с помощью языков программирования общего назначения. В задачу исполняющего серверного приложения при этом входит обеспечение параллельной работы пользователей с моделью с реализацией, по возможности, инвариантных к виду модели способов улучшения характеристик качества обслуживания, например механизма кэширования результатов. Вопросы коллективного сетевого доступа к таким моделям рассмотрены в работе [1], где показаны, в частности, на примере имитационной модели стохастического характера основные особенности реализации такого доступа, а также механизмы улучшения показателей качества обслуживания.

Вместе с тем, большой интерес представляет ситуация, когда на сервере установлена специализированная среда моделирования, причем размещение такой среды в ПК пользователя может оказаться затруднительным. При этом сами по себе модели могут быть достаточно простыми, так что их создание вполне возможно в режиме сетевого удаленного доступа. Подобная ситуация, например, является типичной при использовании технологий дистанционного обучения. Такой режим создания моделей будем далее называть режимом *удаленного программирования*. Реализация такого режима, так же как и собственно работа с моделями при коллективном доступе к ним, может проводиться в рамках использования HTML, http и других элементов Интернета на основе технологии "клиент—сервер" с применением браузера в качестве программного клиента.

Удаленное программирование на базе MATLAB WEB Server

MATLAB как сложный программный пакет построен по классической архитектонике модульных программных приложений. Как правило, в основе подобных сложных приложений используются базовые подходы при реализации тех или иных возможностей среды, в том числе и подходы в области ее удаленного использования. В частности, в состав программного пакета MATLAB включен базовый инструментарий для организации удаленного сетевого доступа. Основой этого инструментария является модуль MATLAB Web Server (MWS). MWS позволяет разрабатывать приложения для работы в Web, использующие стандартные компоненты MATLAB. HTML-документы служат графическим интерфейсом для распределенных приложений MATLAB. Пользователь избавлен от необходимости не только устанавливать на свой ПК данную среду, но и вникать в тонкости ее работы, так как сам принцип организации этих HTML-документов базируется на концепции ввода пользователем лишь определенного числа параметров и инициализации процесса вычислений. Таким образом, MWS позволяет развернуть MATLAB-приложение в сети, используя стандартные Web- технологии. На рис. 1 представлена блок-схема организации комплекса удаленного доступа на основе MWS.

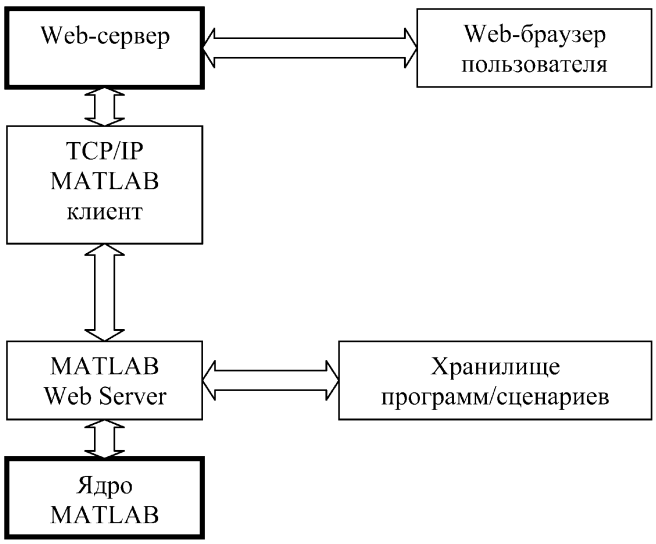


Рис. 1. MATLAB Web Server

За взаимодействие MWS с Web-сервером общего назначения отвечает подмодуль системы MATLAB TCP/IP клиент (далее TCP/IP-клиент). Такой подход позволяет размещать сам MATLAB и Web-сервер на разных компьютерах, связь между которыми осуществляется через протокол TCP/IP посредствам ЛВС.

Общая концепция MWS базируется на том, что пользователь в конечном счете инициирует запуск того или иного файла сценария (программы, *m*-файла в терминологии MATLAB) с определенным набором входных параметров, при этом сам файл пользователь менять не может. TCP/IP-клиент в данном случае исполняет роль интерфейсного коммутатора.

При развертывании системы и ее настройке строго фиксируется расположение ее элементов как по вычислительным ресурсам, так и по файловой организации. В процессе конфигурирования MWS необходимо жестко задать расположение *m*-файлов и их псевдонимы, по которым проводится их идентификация при поступлении запроса на исполнение от TCP/IP-клиента. На Web-сервере находятся HTML-документы, формирующие формы запроса параметров от пользователя, которые пользователь и заполняет перед инициализацией расчетов. В качестве Web-сервера могут выступать такие программные продукты как Microsoft IIS, Apache и др. При этом TCP/IP-клиент, работающий в тандеме с Web-сервером, представляет собой классический CGI-сценарий. CGI (Common Gateway Interface — сценарий общего шлюзового интерфейса) — программа, часто использующаяся для передачи данных из HTML-формы в приложение. CGI-сценарий выполняется на сервере, на котором размещена Web-страница с формой.

Для публикации *m*-файла с помощью MWS необходима некоторая его адаптация. Можно отметить, что такая адаптация относится в большей степени лишь к дополнительным описаниям интерфейсных взаимодействий и призвана помочь MWS разобраться, какие именно данные и в каком формате будет необходимо вернуть по средствам TCP/IP-клиента Web-серверу, который в свою очередь сформирует для браузера пользователя соответствующую HTML-страницу. В конечном итоге такая адаптация сводится к обрамлению программного кода в *m*-файле дополнительными стандартными командами.

Следует отметить, что MWS позволяет работать только с текстовым интерпретатором MATLAB и не позволяет непосредственно использовать такой интерфейсно-графический инструментарий, как *Simulink*. Возможным решением этой проблемы является использование режима терминального доступа, что, в свою очередь, повышает требования как к пропускной способности канала, так и к технической оснащенности сервера.

Недостатки MWS весьма очевидны. Так, из рассмотренного выше становится понятно, что пользователь лишен возможности самостоятельно разрабатывать MATLAB приложения, он лишь может запускать уже готовые приложения, хранящиеся на сервере, меняя значения входных параметров. Вся концепция MWS строится исключительно на фиксированности *m*-файлов на сервере. Более того, являясь коммерческим программным продуктом, MATLAB не допускает глубокого перепрограммирования работы MWS.

Тем не менее, существует возможность организовать режим удаленного программирования, лишь немного дополнив уже существующий базис. Главной идеей при реализации такого режима может стать введение дополнительных программных модулей, функционирующих в рамках Web-сервера, которые призваны к динамической переконфигурации базовых параметров работы MWS. Иными словами, можно в динамическом режиме, например в момент регистрации нового пользователя, организовать изменение главного конфигурационного файла системы, резервирование *m*-файла, интерфейс изменения этого файла пользователем и автоматическую адаптацию его для работы в среде MWS. Таким образом, пользователь получает возможность самостоятельно, например, с помощью простого текстового редактора в браузере, изменять *m*-файл и тем самым запускать собственные расчетные программы. Иными словами, существует проблема персонализации информации, хранящейся на сервере, ее изменения, а также использования в рамках данной функциональности MWS. Рассмотрим базовые принципы решения этой проблемы.

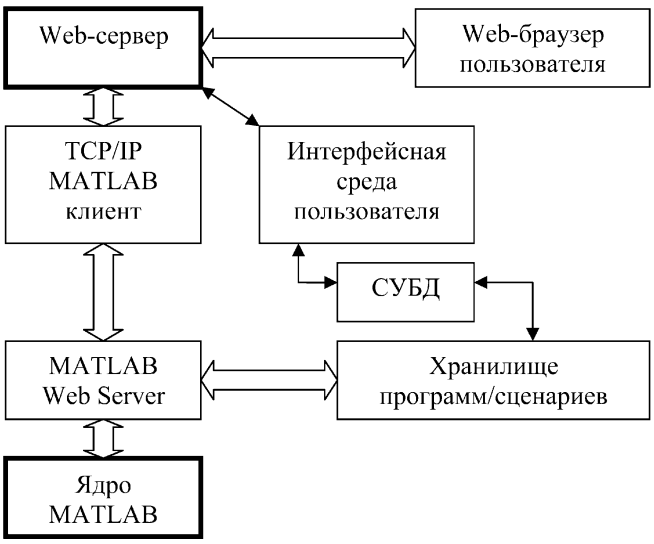


Рис. 2. Модернизация MWS

Одним из ключевых звеньев такой модернизированной системы MWS становится сайт (интерфейсная среда пользователя), роль которого сводится к организации интерфейса взаимодействия с пользователями системы, а также к поддержке базы данных пользовательских программ с организацией разграничения прав доступа к этой информации со стороны других пользователей. В предлагаемом решении можно использовать распространенную СУБД MySQL. Идея заключается в том, что все создаваемые пользователями программы сохраняются не во множество *m*-файлов, а непосредственно в БД, откуда они могут извлекаться, адаптироваться дополнительными программами для работы в режиме удаленного доступа и отправляться на исполнение в среду MWS. Схематично принципы взаимодействия отражены на рис. 2.

Интерфейсная среда пользователя реализуется классическими методами Web-программирования, в предлагаемом решении можно использовать язык сценариев PHP. Данная среда, представляющая собой сайт, реализует интерфейс составления расчетных программ (*m*-файлов) пользователями, а также их персонализируемое хранение средствами базы данных. При этом каждый пользователь может иметь сколь угодно много программ в своей базе, а также структурировано их хранить, например, в виде дерева с папками и подпапками. Доступ к интерфейсной среде осуществляется только после регистрации пользователя в системе, в результате чего пользователь получает уникальный идентификатор в системе, с помощью которого и осуществляется персонализация его данных.

Принцип работы системы следующий. Пользователь создает в текстовом редакторе интерфейсной среды свою расчетную программу, после чего осуществляет ее сохранение в БД. Далее пользователь может инициировать процесс выполнения этой программы ядром MATLAB с помощью интерфейсной кнопки из окна своего браузера. В таком случае с помощью специализированного PHP-сценария программа извлекается из БД, адаптируется для работы в Web-среде и сохраняется в зарезервированном для данного пользователя *m*-файле, и далее средствами TCP/IP-клиента дается команда MWS запустить данный *m*-файл на исполнение. Результаты исполнения *m*-файла выводятся пользователю в браузер. От пользователя требуется при составлении программы дополнять код специализированными директивами, которые в дальнейшем помогут MWS определить, какие именно параметры следует вернуть в качестве результатов обсчета.

Такой подход позволяет получать результаты не только в виде таблиц с текстовой информацией, но и в виде графиков. В этом случае за пользователем резервируется не только *m*-файл в среде MWS, но и специализированный каталог для сохранения графиков в виде графических файлов, которые могут быть добавлены в общий поток результирующей информации при оформлении отчетной HTML-страницы.

Таким образом, удаленный пользователь получает возможность не только запускать уже готовые расчетные программы в среде MATLAB, но и осуществлять составление и запуск собственных программ.

Реализация режима удаленного программирования на базе MATLAB Web Server

Рассмотрим более подробно реализацию режима удаленного программирования с использованием MATLAB Web Server. Как отмечалось выше, штатный режим работы MWS не допускает изменения *m*-файлов пользователем. Для преодоления этого ограничения был разработан набор дополнительных программных модулей на базе PHP и СУБД MySQL.

В качестве Web-сервера был выбран Apache, как бесплатно доступное ПО с достаточно гибкими возможностями конфигурирования. На базе данного Web-сервера был создан сайт со следующей структурой каталогов относительно корневой директории:

- cgi-bin — каталог CGI-сценариев, сюда будет помещен TCP/IP MWS клиент;
- images — каталог используемых для построения интерфейса пользователя изображений;
- mfiles — каталог динамически формируемых пользовательских *m*-файлов;

- templates — каталог для размещения шаблонов HTML-страниц и дополнительных PHP-библиотек, необходимых для функционирования системы.

Главным исполнительным файлом интерфейса пользователя является index.phtml, располагающийся в корневом каталоге сайта. Данный PHP-сценарий динамически формирует те или иные страницы интерфейса в зависимости от действий пользователя, в том числе отвечает и за формирование *m*-файлов пользователя и запуск TCP/IP MATLAB клиента с необходимым набором параметров.

Для корректной работы MWS необходимо изменять файл matweb.conf, который располагается в том же каталоге, что и главный исполнительный файл TCP/IP MATLAB клиента. В данном файле задается уникальное идентификационное имя и соответствующий ему *m*-файл. Поскольку *m*-файлы пользователя динамически формируются по запросу, то данный конфигурационный файл также необходимо формировать динамически. Для предотвращения множественных перезаписей этого файла применяется следующая стратегия: для каждого пользователя резервируется один *m*-файл с уникальным именем, которое и прописывается в файл конфигурации, далее каждый запуск пользователем процесса моделирования будет инициировать перезапись этого *m*-файла данными, извлекаемыми из БД.

Помимо вспомогательных таблиц БД содержит основную таблицу для хранения данных *m*-файлов, где с помощью уникальных идентификаторов формируется структурное дерево соответствия данных каждому пользователю. Таким образом, пользователь может сохранять в системе целый набор MATLAB-сценариев, формируя тем самым свою уникальную библиотеку программ (моделей). Каждая такая единица информации снабжается уникальным идентификатором и описательными данными.

После прохождения пользователем авторизации он попадает в собственный каталог MATLAB-программ, где может выбрать ранее сохраненную программу или создать новую. Если пользователь инициирует исполнение своей программы, то данный запрос поступает в сценарий index.phtml, который проводит необходимые операции для запуска нужной программы. Вначале проводится дополнение пользовательской программы вспомогательным программным кодом, который адаптирует данную программу для исполнения в среде MWS. Далее полученные данные сохраняются в уникальный зарезервированный для каждого пользователя *m*-файл, после чего дается команда TCP/IP MATLAB клиенту на исполнение данного *m*-файла. Фрагмент кода, отвечающего за выполнение описываемых операций, представлен в следующем листинге:

```
if (isAction("run"))
   if (getPostVar("id")) {
      $fields = DBFormFields(array("name", "data",
      "userid"), array("data"));
         if (preg_match_all("!outstruct\.(.*?) = !si",
         $fields["data"], $res))
            $arr_vars = $res[l];
         for ($i = 0; $i < count($arr_vars); $i++)
            $arr_vars[$i] = trim($arr_vars[$i]);
         $images = array();
         if (preg_match_all("!print(.*?);!si",
         $fields["data"], $res)) {
            $arr_pictypes = array("-djpeg" = > ".jpg",
            "-dgif" = > ".gif");
            foreach ($res[l] as $paramstr) {
               $params = explode(" ", $paramstr);
               $pic_name = $params[count($params) - 1] ;
               for ($i = 0; $i < count($params); $i++)
                  if (in_array(trim($params[$i]),
                  array_keys($arr_pictypes)))
                  $pic_type = $arr_pictypes[trim($pa-
                  rams[$i])];
               array_push($images, array("name" = >
               $pic_name, "id" = > count($images) + 1,
               "type" = > $pic_type));
            }
         }
         $mfile_dir = $dir_mfiles . "/" . getPostVar("id");
         $mfile_name = "func" . getPostVar("id") . ".m";
         $mfile_data = "function retstr = func" . getPost-
         Var("id") . "(instruct)\n";
         $mfile_data . = "cd('" . $mfile_dir . "');\n";
         $mfile_data . = $fields["data"] . "\n";
         $mfile_data . = "retstr = htmlrep(outstruct, '" .
         $mfile_dir "/template.html');";
         if (!is_dir($dir_mfiles))
            mkdir($dir_mfiles);
         if (!is_dir($mfile_dir))
            mkdir($mfile_dir);
         if (is_dir($mfile_dir)) {
            $tmpl = LoadTmpl($dir_home . "/tem-
            plates/tmpl/report.html");
            $html = MakeParsedStr($tmpl["header"],
            array("id" = > getPostVar("id"),
            "userid" = > $userid));
            $html . =$tmpl["vars_header"];
            if (is_array($arr_vars))
               foreach ($arr_vars as $varname)
                  $html. = MakeParsedStr($tmpl["variable"],
                  array("varname" = > $varname));
            $html . = $tmpl["vars_footer"];
            if (is_array($images))
               foreach ($images as $image) {
                  $image["url"] = $url_mfiles . "/" . get-
                  PostVar("id") . "/" . $image["name"] .
                  $image["type"];
                  $html . = MakeParsedStr($tmpl["image"],
                  $image);
               }
            $html . = $tmpl["footer"];
            SaveFile($mfile_dir . "/template.html", $html) or
            die("Error: Could not create/save file: 1");
            SaveFile($mfile_dir . "/" . $mfile_name,
            $mfile_data) or
            die("Error: Could not create/save file: 2");
```

```
        $url_mwswork = "/cgi-bin/matweb.exe?mlmfile =
        func" . getPostVar("id") . "&rnd = " .
        time();
    } else die("Error: Directory not found: " .
    $mfile_dir);
} else echo "Error: Invalid ID";
```

Закончив исполнение *m*-файла, MWS возвращает системе результат в виде текстовых данных. Причем, если в *m*-файле была синтаксическая ошибка, то будет возвращено ее текстовое описание, что поможет пользователю быстро устранить ошибку в тексте своей программы. Из полученных данных формируется HTML-страница с помощью соответствующих шаблонов, после чего результат отправляется в браузер пользователя.

Таким образом, система MWS, работая в рамках своей стандартной функциональности, может быть адаптирована для решения задачи удаленного программирования.

Однако в ходе проведенных испытаний реализованного способа удаленного доступа были зафиксированы и определенные недостатки работы системы в целом. В основном, главным "узким" местом системы остается сам MWS. Было установлено, что MWS не может в параллельном режиме обслуживать большое число пользователей и с увеличением числа запросов на исполнение *m*-сценариев начинает ставить задачи в очередь, что при относительно длительной обработке каждого *m*-файла ядром среды MATLAB может привести к отказу в обслуживании по истечении времени ожидания данных браузерами остальных пользователей в очереди. Число параллельно исполняемых запросов может регулироваться конфигурацией MWS, но при этом существенно возрастают требования к объему ОЗУ сервера, так как режим параллельной обработки реализуется путем удержания в памяти необходимого числа обработчиков запросов MWS, каждый из которых может занимать до 50 Мбайт ОЗУ в режиме ожидания.

Таким образом, использование концепции, навязываемой архитектоникой MWS, приводит к существенно нерациональному использованию вычислительных возможностей сервера, что автоматически снижает уровень сложности возможных сценариев *m*-файлов в системе. Также отсутствует механизм контроля работы MWS как программы, что при определенных обстоятельствах может привести к отказу работы всего сервера в целом.

Реализация режима удаленного программирования на базе MATLAB COM

Как отмечалось выше, MWS существенно ограничивает возможности по реализации режима удаленного программирования, однако можно предложить подход, который позволит, используя вышеприведенные идеи, решить обозначенную проблему. В качестве возможного решения можно предложить переход к формированию собственной программы-шлюза между ядром MATLAB и Web-сервером, а именно использование технологии COM (*Component Object Model* — модель компонентных объектов) [2] при реализации такой программы.

Технология COM предназначена для создания программного обеспечения на основе взаимодействующих распределенных компонентов, каждый из которых может использоваться во многих программах одновременно. Технология воплощает в себе идеи полиморфизма и инкапсуляции объектно-ориентированного программирования, в принципе являясь универсальной и платформенно независимой.

Предлагаемое решение для реализации сервиса удаленного доступа на базе COM предполагает замещение MWS в описанной выше схеме взаимодействия. В этом случае появляется возможность реализовать программный шлюз между ядром MATLAB и Web-сервером с учетом специфики реализуемого удаленного доступа, что было затруднительно сделать, используя MWS. При этом основная концепция построения системы остается такой же, как и при реализации системы на базе MWS, за тем лишь исключением, что компоненты MWS замещаются специально разработанными подпрограммами. Пользователь при этом работает с той же базой данных, содержащей модели, и может их запускать так же, как и в предыдущем случае, но в этом случае получает гораздо больше информации от ядра MATLAB.

В результате работа в режиме удаленного доступа становится сходной с работой в стандартной среде MATLAB при составлении *m*-файлов. Кроме того, пользователь получает больше отладочной информации при попытках запуска своих программ, чем в случае использования MWS. Тем не менее, оценка вычислительного ресурса, необходимого для обеспечения коллективного сетевого доступа при рассматриваемой реализации на базе COM, требует отдельного исследования.

Список литературы

1. **Иванов Д. И., Цикин И. А.** Сетевая реализация стохастических моделей радиотехнических устройств // Известия вузов России. Радиоэлектроника, 2006. Вып. 4. С. 34—42.

2. **Каплан А., Нильсен М. Ш.** Windows 2000 изнутри. М.: ДМК, 2000. 400 с.

УДК 004.5; 004.8

А. С. Клещев, д-р физ.-мат. наук, зав. отделом,
Институт автоматики и процессов управления
ДВО РАН, г. Владивосток

Роль онтологий в программировании. Часть 2. Интерактивное проектирование информационных объектов*

Работа завершает цикл из двух статей, посвященных обсуждению ряда направлений в программировании, где использование онтологий может привести к существенному прогрессу. В ней рассматриваются различные варианты задачи интерактивного проектирования информационных объектов, а также некоторые применения онтологий в области искусственного интеллекта.

Ключевые слова: *интерактивное проектирование информационных объектов, компьютерная хореография, интерактивное проектирование музыки, интерактивное проектирование пользовательских интерфейсов, вербальное представление знаний, индуктивное формирование знаний на основе онтологий предметных областей, специализированная оболочка, основанная на онтологии предметной области, повторное использование баз знаний, моделирование преобразований компьютерных программ.*

Интерактивное проектирование

Рассмотрим задачу построения целевых информационных объектов некоторого класса, каждый из которых обладает некоторым содержанием и имеет некоторую форму. Предположим, что имеются две группы специалистов — кодировщики (специалисты в области формы) и эксперты (специалисты в области содержания этих объектов). Кодировщики не знают содержания, которым должен обладать каждый конкретный целевой объект, но умеют строить целевые объекты нужной формы. Эксперты знают содержание целевого объекта, который нужно построить, но не умеют строить целевые объекты нужной формы (либо построение таких объектов для них достаточно трудоемко). В процессе жизненного цикла содержание целевых объектов может изменяться, что влечет за собой изменение и в их форме. Традиционный метод решения этой задачи состоит в том, что эксперты объясняют кодировщикам, каким содержанием должен обладать целевой объект, который нужно построить, после чего кодировщики строят требуемый объект. Каждый раз при изменении содержания уже построенного целевого объекта взаимодействие экспертов с кодировщиками возобновляется. Понятно, что чем сложнее содержание целевого объекта, тем более трудоемким и ненадежным является этот метод. Примером может служить разработка программ (как целевых объектов), где требуются две группы посредников — аналитики и проектировщики, чтобы наладить более или менее надежную и эффективную передачу информации между экспертами и кодировщиками.

Рассмотрим частный случай этой задачи. Предположим, что для каждого целевого объекта заданного класса может быть определена информационная структура — его проект (модель), удовлетворяющая следующим требованиям:

- проекты целевых объектов должны быть вербализуемой информацией [1];
- терминология их вербального представления должна быть известна экспертам;
- может быть разработана интерактивная программа, которая позволяет экспертам строить проекты целевых объектов в этом вербальном представлении;
- может быть разработана программа, которая преобразует вербальные представления проектов в целевые объекты.

Назовем эту задачу *задачей интерактивного проектирования*. В ней речь идет об автоматизации деятельности экспертов по построению и модификации целевых объектов заданного класса (с исключением кодировщиков из этого процесса). Проектирование может быть только интерактивным, поскольку предполагается, что эксперты не обладают навыками использования каких-либо формальных языков и им нужна интеллектуальная поддержка в этом процессе (либо интерактивный режим существенно снижает трудоемкость проектирования). Чтобы в этих предположениях автоматизировать деятельность экспертов, необходимо решить следующие подзадачи:

* Работа выполнена при финансовой поддержке ДВО РАН в рамках Программы Президиума РАН № 14 "Фундаментальные проблемы информатики и информационных технологий", проект 06-I-П14-051 "Интеллектуальные системы, основанные на многоуровневых моделях предметных областей".

- разработать онтологию и базу знаний проектов, а также онтологию деятельности по проектированию, удовлетворяющие второму из перечисленных выше требований;
- на основе этих онтологий и баз знаний разработать интерактивную САПР для экспертов, контролирующую тот факт, что создаваемые и модифицируемые с ее помощью проекты относятся к целевым объектам заданного класса;
- разработать транслятор (компилятор или интерпретатор) вербального представления проектов в целевые объекты.

Хотя задача интерактивного проектирования успешно решается в области техники, ее решение в областях, далеких от техники, менее распространено и, как правило, основывается на интуиции. Поэтому ниже приведено несколько примеров задач интерактивного проектирования, которые могут решаться (или уже успешно решаются) на основе онтологий.

Интерактивное редактирование вербального представления информации под управлением онтологий является одной из задач интерактивного проектирования. Задача состоит в разработке таких интерпретаторов онтологий (метаонтологий), которые поддерживают деятельность экспертов по созданию и редактированию информационных ресурсов различных уровней общности (баз данных, баз знаний, онтологий и т. п.). В настоящее время уже созданы такие интерпретаторы как для локальных компьютеров, так и доступные через сеть Интернет [2].

Задача интерактивного проектирования танцев с помощью компьютера является сравнительно мало исследованной [3]. Если речь идет о современных сольных танцах, то в настоящее время основными инструментами хореографа являются зеркало, в котором он видит танец, им спроектированный и исполненный, и видеокамера, которая фиксирует это исполнение и позволяет передать его ученикам хореографа. Целевыми объектами интерактивного проектирования в этой задаче являются анимированные изображения танцев определенного стиля (компьютерные мультфильмы), а экспертами — хореографы. Для разработанной системы интерактивного проектирования танцев стиля "Фанк" [4] онтология проектов состоит из двух частей — онтологии танца, в которой определяются такие понятия, как танец стиля "Фанк", движение, поза, кинетотакт и др., и онтологии биомеханики человека, в которой определяются термины анатомического строения человека, важные для хореографии, а в базе знаний — ограничения биомеханики человека. Интерактивная САПР на основе этих онтологий и онтологии проектирования танцев, движений, а также поз всего тела и отдельных его частей поддерживает графический диалог с хореографом. В процессе проектирования могут также использоваться расширяемые библиотеки ранее спроектированных танцев, движений и поз. Результатом проектирования является проект танца, который может быть превращен в компьютерный мультфильм и визуализирован (весь проект и его фрагменты) как в процессе проектирования, так и по его окончании.

Аналогичным образом может быть решена задача интерактивного проектирования и анализа музыкальных произведений. Объектами проектирования в этом случае являются музыкальные произведения в нотной записи (в форме MIDI-файлов, которые могут быть озвучены), а экспертами — композиторы и музыковеды. Онтология проектов музыкальных произведений должна содержать определения таких терминов, как музыкальная форма, план произведения и каждой его части, тональный план, тема, вариации, соединение голосов, термины гармонии и т. п. Первый шаг к созданию такой онтологии сделан в [5]. База знаний должна содержать возможные формы и их планы, ограничения на тональные планы, семантические структуры тем, типы вариаций, возможные гармонические соотношения и т. п. Задачей САПР является интерактивное построение проекта музыкального произведения, преобразование его в нотную форму и озвучивание. Система анализа должна позволять музыковеду выполнять анализ музыкального произведения в терминах той же онтологии, т. е. позволять разбирать его на такие части, чтобы с помощью обратных операций из этих частей можно было его собрать заново. Анализ реальных музыкальных произведений может быть основным способом формирования базы знаний для такой САПР.

В задаче интерактивного проектирования пользовательских интерфейсов объектами проектирования являются пользовательские интерфейсы, представленные в форме программного кода, работающего вместе с соответствующей прикладной программой. Экспертами в этой задаче выступают:
- эксперты той предметной области, для взаимодействия со специалистами которой предназначен проектируемый интерфейс;
- дизайнеры, проектирующие внешний вид интерфейса;
- эргономисты, проектирующие сценарий диалога;
- системные программисты, проектирующие способ связи интерфейса с прикладной программой;
- специалисты по юзабилити интерфейса, которые оценивают юзабилити полученного проекта.

Каждая из перечисленных групп экспертов имеет свою терминологию, которая определена соответствующей онтологией, и свои знания, ко-

торые представлены в базе знаний разработанной САПР Interdev. Пользовательский интерфейс является частью прикладной программы (информационной системы), наиболее часто требующей модификации вследствие изменения требований пользователя и условий эксплуатации, а также модификации прикладной программы [6].

Сопровождение информационной системы — это ее адаптация к изменению условий эксплуатации. Чем сложнее информационная система, тем более трудоемким является ее сопровождение. Для развивающейся информационной системы может настать такой кризисный момент, когда все усилия ее разработчиков будут уходить на ее сопровождение (адаптацию того, что уже сделано, к изменению условий эксплуатации), и не останется ресурсов на ее дальнейшее развитие (расширение ее функциональности). Примерами таких развивающихся информационных систем являются системы автоматизации управления университетами. Для таких систем изменения условий эксплуатации состоят в том, что существующие бизнес-процессы могут часто изменяться, постоянно появляются новые бизнес-процессы, а некоторые старые должны перестать выполняться, форматы документов и модели данных часто меняются (особенно в периоды реформирования системы образования). В то же время функциональность таких систем чрезвычайно широка — это управление кадрами (студентами, профессорско-преподавательским и вспомогательным составом), учебным процессом, научно-исследовательской работой, документооборотом, регулярными и разовыми мероприятиями и т. п., вплоть до контроля посещения общежитий [7].

Проблема сопровождения таких информационных систем может быть решена, если каждую такую информационную систему разрабатывать в виде специальной системы интерактивного проектирования и интерпретатора проекта. Система интерактивного проектирования поддерживает создание, развитие и сопровождение проекта информационной системы. Эта деятельность выполняется экспертами — теми специалистами, терминология которых определяется онтологиями, положенными в основу системы интерактивного проектирования (например, специалистами отделов кадров, студенческих отделов, учебно-методических управлений и т. п. в случае системы автоматизации управления университетами). Интерпретатор проекта поддерживает функционирование информационной системы.

Вербальное представление знаний

Знания, как и всякая информация, могут иметь вербальное представление [1]. Для этого необходимо, чтобы в предметной области существовала такая терминология, специально предназначенная для представления знаний, что вербальное представление знаний предметной области, основанное на этой терминологии, адекватно представляет эти знания. Онтологию, определяющую смысл такой терминологии, будем называть *онтологией знаний*, в отличие от онтологии, определяющей смысл терминов для представления ситуаций, которую будем называть *онтологией действительности*. Онтологические соглашения онтологии знаний будем называть ограничениями целостности знаний, а онтологические соглашения онтологии действительности — ограничениями целостности ситуаций. В этом случае онтология предметной области состоит из двух онтологий — онтологии действительности и онтологии знаний, которые связаны специальной системой онтологических соглашений о связи между знаниями и действительностью [8]. Примером такой онтологии является онтология медицинской диагностики [9].

Традиционным способом формирования баз знаний является их получение от экспертов. В настоящее время существуют средства редактирования баз знаний, ориентированные на экспертов и управляемые онтологиями знаний, о чем шла речь в предыдущем разделе. Другим способом является индуктивное формирование баз знаний на основе эмпирических данных. Как отметил Д. Мики [10], автоматически сформированная база знаний может быть полезна только в том случае, если она понятна специалистам соответствующей предметной области. В этом случае специалисты смогут не только сами пользоваться этими знаниями в своей профессиональной деятельности, но и будут доверять экспертной системе, использующей эту базу знаний (а также смогут проверять выводы этой системы). Кроме того, любой элемент индуктивно сформированной базы знаний должен допускать понятное специалисту предметной области объяснение (как и из каких эмпирических данных он получен). В тех предметных областях, где знания имеют вербальное представление, эти знания понятны специалистам этих предметных областей, поскольку смысл терминов соответствующей онтологии знаний им известен. В [11] предложены методы индуктивного формирования баз знаний для таких предметных областей. Эти методы предназначены для решения следующей задачи: по обучающей выборке вербально представленных ситуаций (примеров — возможных ситуаций, и контрпримеров — невозможных ситуаций) найти вербальное представление базы знаний, (наиболее) правильной (каждый пример согласуется с базой знаний) и (наиболее) точной (каждый контрпример не согласуется с базой зна-

ний) относительно обучающей выборки. Основная идея этих методов — использование при решении этой задачи онтологических соглашений о связи действительности и знаний в онтологии предметной области.

Каждая система, основанная на знаниях, состоит из двух частей — оболочки и базы знаний, а экспертная система — это система, основанная на знаниях, база знаний которой имеет высокий уровень компетентности. Традиционный, основанный на модели представления знаний подход к реализации таких систем состоит в том, что:

- инженер знаний выбирает или программист разрабатывает универсальную оболочку — интерпретатор некоторой модели представления знаний (например, системы продукций);
- инженер знаний с помощью эксперта формирует (а затем и сопровождает) базу знаний в форме, определяемой моделью представления знаний.

Если же база знаний в предметной области имеет вербальное представление, то возможен другой, основанный на модели онтологии знаний подход к реализации экспертных систем:

- инженер знаний с помощью эксперта формирует онтологию предметной области, включающую онтологию знаний;
- программист разрабатывает специализированную оболочку — интерпретатор онтологии знаний;
- эксперт с помощью редактора знаний, управляемого онтологией знаний, формирует (а затем и сопровождает) базу знаний.

Сравнение этих двух подходов показывает, что создание демонстрационного образца системы, основанной на знаниях, с помощью существующей универсальной оболочки можно выполнить значительно быстрее в рамках первого подхода, но создание экспертной системы с базой знаний, сопровождаемой в течение длительного времени, — только в рамках второго.

Если задачи, которые должна решать система, основанная на знаниях, могут быть сформулированы в терминах метаонтологии (широкой предметной области или класса предметных областей), а базы знаний (предметных областей или их разделов) имеют вербальное представление, то эта метаонтология может быть положена в основу специализированной метаоболочки, настройка которой на предметные области класса или разделы широких областей делает ее оболочкой для этих областей или разделов. Такая метаоболочка должна содержать в своем составе и многоуровневый редактор, настраиваемый с помощью онтологий. В настоящее время разработана такая метаоболочка в области медицины, настраиваемая на ее различные разделы.

Обычно для каждого проекта системы, основанной на знаниях, разрабатывается своя онтология, в терминах которой и формируются базы знаний для этого проекта. В результате, например, в области медицинской диагностики существуют десятки экспертных систем и баз знаний для них, несовместимых между собой. В то же время именно повторная используемость баз знаний была одним из стимулов выделения онтологий в самостоятельное понятие и последующее их изучение. Повторная используемость может быть достигнута, прежде всего, для баз знаний, сформированных в терминах реальных онтологий. Эти базы знаний оказываются независимыми от экспертных систем, для которых они предназначены. При этом и специализированная оболочка становится независимой от этих баз знаний — любая база знаний, сформированная в этих терминах, допускается этой оболочкой.

Создание и, особенно, сопровождение баз знаний (поддержание их в актуальном состоянии) одним экспертом или группой экспертов, специально сформированной для этих целей, в случае реальных экспертных систем редко оказывается возможным на практике, поскольку связано с большими трудозатратами. Эта проблема порождает определенный кризис экспертных систем как направления искусственного интеллекта [12]. Одним из выходов из этой ситуации является коллективное развитие баз знаний, независимых от конкретных экспертных систем. В общем виде этот подход может быть сформулирован как коллективное развитие информационных ресурсов, независимых от программ их обработки. Одним из инструментов для поддержки коллективного развития информационных ресурсов является *Многоцелевой банк знаний* [13]. Он является Интернет-системой, предоставляющей средства коллективного развития информационных ресурсов различных уровней общности с помощью редактора ИРУО, управляемого метаинформацией [2], и средства доступа к этим ресурсам для обрабатывающих программ с помощью оболочки банка.

Специализированные интеллектуальные пакеты прикладных программ также могут разрабатываться в виде систем, основанных на вербальном представлении знаний. Универсальные интеллектуальные пакеты прикладных программ ПРИЗ [14] и СПОРА [15], основанные на фреймовых моделях представления знаний, не получили широкого практического применения. Специализированные пакеты прикладных программ ориентированы на онтологии соответствующих предметных областей (например, на онтологию физической химии [16]); они предназначены для того, чтобы по постановкам задач в терминах этой онтологии и по базе знаний находить методы реше-

ния этих задач, по ним синтезировать программы, а по этим программам и исходным данным задач вычислять их решение. Метод реализации специализированного пакета прикладных программ может включать преобразование базы знаний пакета в вычислительную модель Э. Тыугу [14], по которой и ищется решение каждой задачи.

Еще одно применение интеллектуальных систем, основанных на вербальном представлении знаний, — *оптимизирующие компиляторы*, управляемые базами знаний [17]. Существующие оптимизирующие компиляторы имеют жестко встроенные наборы применяемых трансформаций и стратегии их применения. Это не позволяет использовать такие компиляторы для проведения многих компьютерных экспериментов по изучению свойств тех или иных трансформаций и стратегий их применения. Не разработаны также инструментальные средства для макетирования таких компиляторов. Оптимизирующие компиляторы, управляемые базами знаний, могут использоваться для компьютерных экспериментов в области оптимизации и преобразований программ, макетирования оптимизирующих компиляторов и создания средств активного обучения оптимизации и преобразованиям программ студентов.

В качестве теоретической основы таких компиляторов могут использоваться уже разработанные онтология трансформаций различных классов и онтология потокового анализа. Оптимизирующий компилятор, управляемый базой знаний, может включать:

- синтаксически управляемый анализатор (или редактор) программ;
- интерпретатор вербального представления проекции исходного языка на универсальное представление программ;
- интерпретатор вербального представления стратегии применения трансформаций;
- интерпретатор вербального представления методов потокового анализа программ;
- интерпретатор вербального представления трансформаций;
- интерпретатор вербального представления проекции универсального представления программ на язык целевой платформы;
- соответствующие редакторы баз знаний, управляемые онтологиями знаний.

Синтаксически управляемый анализатор (или редактор) программ предназначен для настройки такого компилятора на разные входные языки программирования.

Интерпретатор вербального представления проекции входного языка на универсальное представление программ служит для преобразования дерева разбора программы на входном языке в ее универсальное представление.

Интерпретатор вербального представления стратегии применения трансформаций предназначен для настройки компилятора на заданную стратегию применения трансформаций. Стратегия определяет порядок применения методов потокового анализа и трансформаций из базы знаний.

Интерпретатор вербального представления методов потокового анализа программ служит для выполнения потокового анализа, определяемого стратегией, т. е. обогащения универсального представления программы вычисляемыми значениями атрибутов.

Интерпретатор вербального представления трансформации предназначен для выполнения определяемых стратегией трансформаций в обогащенном универсальном представлении.

Интерпретатор вербального представления проекции универсального представления программ на язык целевой платформы служит для настройки компилятора на разные целевые платформы.

Наконец, редакторы баз знаний предназначены для формирования в базах знаний описаний различных языков программирования, их проекций на универсальное представление, стратегий применения, методов потокового анализа, наборов трансформаций, а также проекций универсального представления на разные целевые платформы.

Заключение

Вышеизложенное позволяет сделать вывод, что онтологии могут стать еще одним средством в борьбе с преодолением некоторых видов сложности при разработке программного обеспечения. С ними связан новый подход к анализу и формализации информации о сложных предметных областях, необходимый при проектировании сложных информационных систем. Использование онтологий приводит к прорыву при разработке ряда систем интерактивного проектирования нетехнических объектов. Использование онтологий знаний при разработке проблемно-ориентированных методов индуктивного формирования знаний в вербальном представлении позволяет реализовать тезис Д. Мики о том, что индуктивно сформированные знания должны быть не только по содержанию, но и по форме в равной мере доступны как специалистам предметной области, так и экспертным системам. Основанные на знаниях системы, разработанные в виде специализированных оболочек, теоретической базой которых являются повторно используемые онтологии, включают механизмы, позволяющие передать сопровождение таких систем от программистов к экспертам и специалистам предметной области.

Использование онтологий позволило также решить проблему создания трансформационной машины для преобразования программ с изменяемым набором трансформаций. Систематическое использование онтологий при разработке программного обеспечения естественно дополняется средствами поддержки коллективного развития баз знаний и других информационных ресурсов.

Список литературы

1. **Клещев А. С.** Роль онтологий в программировании. Часть 1. Аналитика // Информационные технологии. 2008. № 10. С. 42—46.
2. **Клещев А. С., Орлов В. А.** Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации // Информационные технологии. 2006. № 5. С. 25—31.
3. **Перцовский С. Л.** Построение САПР современного сольного танца. Обзор литературы. Владивосток: ИАПУ ДВО РАН, 2006. 62 с.
4. **Перцовский С. Л., Варнина А. С.** Разработка интеллектуальной САПР современного сольного танца на основе онтологий // Вестник ДВО РАН. 2006. № 3. С. 163—169.
5. **Кузин-Алексинский А. С.** Генератор вариаций на заданную музыкальную тему // Информатика и системы управления. 2004. № 1. С. 107—116.
6. **Грибова В. В., Клещев А. С.** Использование методов искусственного интеллекта для проектирования пользовательского интерфейса // Информационные технологии. 2005. № 8. С. 58—61.
7. **Крюков В. В., Шахгельдян К. И.** Корпоративная информационная среда вуза // Владивосток: Дальнаука, 2007. 308 с.
8. **Клещев А. С., Артемьева И. Л.** Математические модели онтологий предметных областей // Научно-техническая информация. Сер. 2. 2001. Ч. 1. № 2. С. 20—27; Ч. 2. № 3. С. 19—28; Ч. 3. № 4. С. 10—15.
9. **Клещев А. С., Москаленко Ф. М., Черняховская М. Ю.** Модель онтологии предметной области "Медицинская диагностика" // Научно-техническая информация. Сер. 2. Ч. 1, 2005. № 12. С. 1—7; Ч. 2. 2006. № 2. С. 19—30.
10. **Michie D.** Expert systems // Computer Journal. 1980. V. 23. № 4. P. 369—376.
11. **Клещев А. С.** Задачи индуктивного формирования знаний в терминах непримитивных онтологий // Научно-техническая информация. Сер. 2. 2003. № 8. С. 8—18.
12. **Артемьева И. Л., Гаврилова Т. Л., Грибова В. В.** и др. Мультидисциплинарная система управления информационными ресурсами различных уровней общности // Проблемы управления. 2006. № 4. С. 64—68.
13. **Клещев А. С., Орлов В. А.** Компьютерные банки знаний. Многоцелевой банк знаний // Информационные технологии. 2006. № 2. С. 2—8.
14. **Тыугу Э. Х.** Концептуальное программирование. М.: Наука, 1984. 285 с.
15. **Бабаев И. О., Новиков Ф. А., Петрушина Т. И.** Язык Декарт — входной язык системы СПОРА // Прикладная информатика. 1981. Вып. 1. С. 35—73.
16. **Артемьева И. Л., Цветников В. А.** Фрагмент онтологии физической химии и его модель // Электронный журнал "Исследовано в России". 2002. № 3. С. 454—474. http://zhurnal.ape.relarn.ru/ articles/2002/042.pdf.
17. **Клещев А. С., Князева М. А.** Управление информацией о преобразованиях программ // Известия РАН. Теория и системы управления. 2005. Ч. 1. № 5. С. 118—127; Ч. 2. № 6. С. 121—130.

УДК 004.416.6

Г. Б. Евгенев, д-р техн. наук, проф.,
МГТУ им. Н. Э. Баумана

Многоагентная методология — новая информационная технология создания прикладных систем

Описывается новая информационная технология создания интеллектуальных прикладных систем без привлечения программистов, обеспечивающая резкое сокращение трудоемкости разработки и эксплуатации этих систем.

Ключевые слова: *информационные технологии, многоагентные системы, теория программирования, теория искусственного интеллекта, модули знаний, язык UML, интеллектуальные агенты.*

В настоящее время, по мнению ряда ведущих специалистов, наметилась стагнация в определенных областях информационных технологий. В частности, концептуальный застой явно характерен для систем автоматизированного проектирования. Изначально CAD-системы были задуманы как компьютерный аналог механических кульманов. И несмотря на длительный путь развития, они остаются пусть мощными, но все же электронными кульманами, предназначенными для ручной работы. При этом инженерные знания, на основе которых проводится проектирование, остаются в форме книг, т. е. пассивных информационных ресурсов, пригодных для использования человеком, но непригодных для эксплуатации на компьютере. В то же время, если бы эти знания были превращены в активные информационные ресурсы в форме баз знаний, то для типового вариантного проектирования можно было превратить САПР в системы полуавтоматического проектирования. В таких системах достаточно ввести техническое задание, ответить на вопросы компьютера и автоматически сгенерировать весь комплект геометрических моделей и конструкторской документации. При этом трудоемкость проектирования может быть сокращена в десятки раз. Однако традиционные алгоритмические методы программирования малопригодны для ввода знаний в компьютер, а средства искусственного ин-

теллекта не могут быть применены для создания комплексных систем с использованием сетевых баз данных, параметризованных графических и геометрических образов и т. д.

Отсюда вытекает необходимость разработки принципиально новых, "прорывных" информационных технологий создания интеллектуальных прикладных систем. Такие технологии должны впитать в себя все лучшие достижения как из области традиционного алгоритмического программирования, так и из области искусственного интеллекта. Нужно, чтобы была возможность формирования баз знаний непосредственно экспертом без привлечения таких посредников, как инженер по знаниям. Инструментальные средства новой технологии должны обеспечивать как генерацию текстового представления баз знаний на языке деловой прозы в форме документов, доступных для чтения и понимания специалистами, так и генерацию эффективных программных кодов, для улучшения которых не нужно будет привлекать программистов. При этом трудоемкость создания прикладных систем должна стать на порядок ниже, чем при традиционном программировании.

Для разработки концептуальных основ подобных новых информационных технологий целесообразно воспользоваться методами эволюционного моделирования, и в частности операторами генетических алгоритмов. Как известно, в их число входят операторы селекции (выбора), кроссинговера (скрещивания) и мутации.

Из изложенного выше следует, что для создания новой технологии необходимо отобрать объекты из двух областей: теории программирования и теории искусственного интеллекта. Результаты такой селекции представлены в таблице. Мутация должна обеспечить изменение свойств выбранного объекта в целях использования его в новом качестве. В таблице мутация изображена в форме односторонних стрелок. Скрещивание дает взаимный обмен свойствами для получения нового качества. В таблице скрещивание представлено в виде двухсторонних стрелок.

Концептуальная основа многоагентной технологии

Селекция из теории программирования	Селекция из теории искусственного интеллекта
Оператор языка программирования	← Продукционное право
Диаграмма деятельности	← Машина логического вывода
Метод класса объектов	← Продукционная система
Класс объектов	↔ Фрейм-прототип
Экземпляр объекта	↔ Экземпляр фрейма
Диаграмма классов	↔ Семантическая сеть
Диаграмма объектов	↔

1. Модули знаний

Универсальная модель представления знаний — это модель представления знаний, применимая для большинства проблемных областей. В искусственном интеллекте основными универсальными моделями представления знаний являются семантические сети, фреймы, продукционные системы и логические модели. В описываемой технологии логические модели не используются.

Наиболее популярным средством представления знаний являются продукционные правила. Продукционное правило — это правило вида "УСЛОВИЕ—ДЕЙСТВИЕ", т. е. если "УСЛОВИЕ", то "ДЕЙСТВИЕ" и т. п. Продукционные правила работают совместно с рабочей памятью, которая в описываемой технологии представляет собой словарь, используемый при создании правил, совместно с текущими значениями свойств из этого словаря. Продукционное правило может применяться только в том случае, когда текущее состояние рабочей памяти продукционной системы удовлетворяет условию "УСЛОВИЕ". Выполнение продукционного правила заключается в изменении информационной структуры продукционной системы в соответствии с заключением "ДЕЙСТВИЕ".

Оператор языка программирования задает полное описание некоторого действия, которое необходимо выполнить. Таким образом, продукционное правило в принципе может выполнять функции оператора программирования. Однако, чтобы язык, построенный на таких операторах, был функционально полным, продукционные правила должны быть способны выполнять все виды действий, необходимых для построения прикладных систем. Основной набор таких действий включает:

- вычисление по формулам (в том числе присвоение значений переменным);
- определение значений по таблицам;
- выбор значений из баз данных;
- обновление значений в базах данных;
- занесение значений в базы данных;
- вычисление значений с использованием подпрограмм;
- вычисление значений с помощью методов, сгенерированных из модулей знаний;
- вычисление значений с помощью исполняемых *exe*-модулей или *dll*-библиотек, сгенерированных другими системами.

В структурном программировании есть фундаментальная теорема о структурировании, в которой доказано, что любая простая программа функционально эквивалентна структурной программе, составленной из элементов базисного множества {последовательность, ifthenelse, whiledo}.

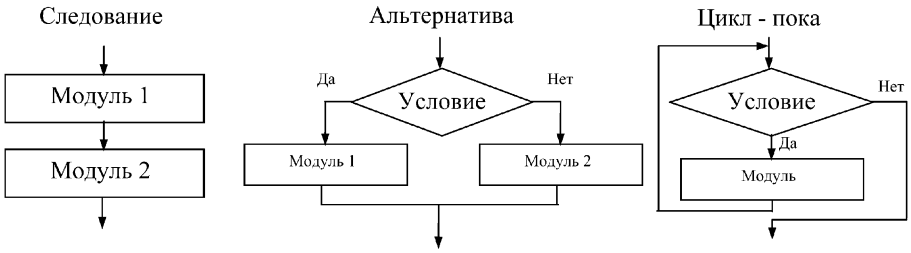


Рис. 1. Основные структуры процедурной декомпозиции

Это означает, что любая программа, представляющая собой один функциональный узел, может быть составлена из структур трех базовых типов (рис. 1).

Продукционные правила, рассматриваемые как модули, естественным образом реализуют структуры "Следование" и "Альтернатива". Следование правил устанавливается по мере определения переменных, необходимых для их исполнения, а альтернативы реализуются предусловиями правил. Цикл - пока в описываемой технологии реализован с помощью выделенной переменной *Fincalc*, при появлении которой организуется циклическое выполнение набора правил вплоть до выполнения определенного условия, изменяющего в одном из правил значение *Fincalc* из исходного, равного 1, на 0.

Новым в описываемой технологии является рассмотрение продукционного правила в качестве динамической объект-функции [1]. Графически такая объект-функция в соответствии с международным стандартом IDEF0 представляется в форме прямоугольника (рис. 2). Каждая из четырех сторон прямоугольника имеет определенное назначение: левая — входы, правая — выходы, верхняя — управление, нижняя — механизмы. Входы представляют собой информацию, необходимую для выполнения функции, и в результате ее выполнения преобразуются в выходы. Входы показывают все характеристики, которые необходимы для выполнения функции, и она не может быть выполнена без получения их значений. Управление описывает условие, оказывающее влияние на выполнение функции, но само не подвергается переработке.

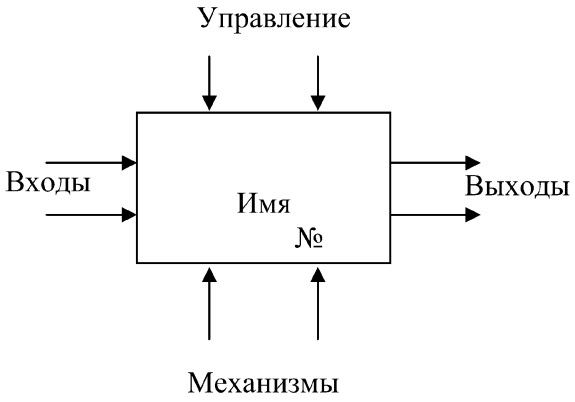


Рис. 2. Объект-функция

К нижней части изображения объект-функции присоединяются стрелки механизмов, обозначающие программное средство, которое обеспечивает выполнение функции. Входы и выходы показывают, что делается функцией, управление — почему это делается, а механизмы — с помощью чего делается.

Понятие "объект" имеет две взаимосвязанные и в то же время относительно самостоятельные стороны: набор свойств, значения которых определяют состояние объекта, и набор правил поведения, из которых формируется метод объекта. В данном случае набор свойств включает все переменные, входящие во входы, выходы и управление, а правила поведения определяются предусловиями и механизмами.

Описанная объект-функция относится к классу динамических, поскольку она обеспечивает выполнение действия, т. е. атомарного вычисления, которое приводит к изменению состояния системы или возврату значения.

На базе описанных принципов разработан новый вид программирования, получивший название экспертного программирования [1, 2]. В экспертном программировании динамическая объект-функция получила название "модуль знаний".

Как показывает опыт, этот метод экспертного программирования доступен для непрограммирующих экспертов, обеспечивая генерацию эффективных программных кодов при повышении производительности процесса в 7—10 раз в сравнении с опытными программистами. Для реализации экспертного программирования разработана инструментальная система SprutExPro.

Ниже приведен пример внешнего представления модуля знаний на языке деловой прозы (рис. 3). Такой язык не является литературным, но вполне доступен для понимания экспертами.

Приведенный пример представляет собой модуль знаний с табличным механизмом реализации. Модуль имеет заголовок с идентификатором, который определяет имя подпрограммы, генерируемой при трансляции, а также описание функции на литературном языке. Далее следует таблица предусловий запуска. В ней определено, что модуль будет запускаться для редукторов типа "цилиндрический" при числе ступеней, равном 2. Модуль имеет две входные переменные символьного типа: переменную "Схема первой ступени", которая принимает значения "одинарная" и "раздвоенная", и переменную "Расположение входной

МИЗ: "NzRsKl12" - Задание положения колеса 1-й ступени относительно опор и кода схемы 2

Предусловия запуска

имя	наименование	тип	условие
Nst	Количество ступеней	INTEGER	2
TypRed$	Тип редуктора	STRING	цилиндрический

Входные свойства

имя	наименование	тип	значение
ShSt1$	Схема первой ступени	STRING	
RaspOs$	Расположение входной и выходной осей	STRING	

Механизм - Таблица

Конфигурация свойств в таблице

	RaspOs$
ShSt1$	PolKol1$
	KodSh1

Таблица

	параллельное	соосное
одинарная	несимметричное	несимметричное
	4	5
раздвоенная	несимметричное	симметричное
	3	5

Выходные свойства

имя	наименование	тип	значение
KodSh1	Код схемы передачи 1-й ступени	INTEGER	
PolKol1$	Положение зубчатых колес относительно опор 1-ст	STRING	

Рис. 3. Пример внешнего представления модуля знаний

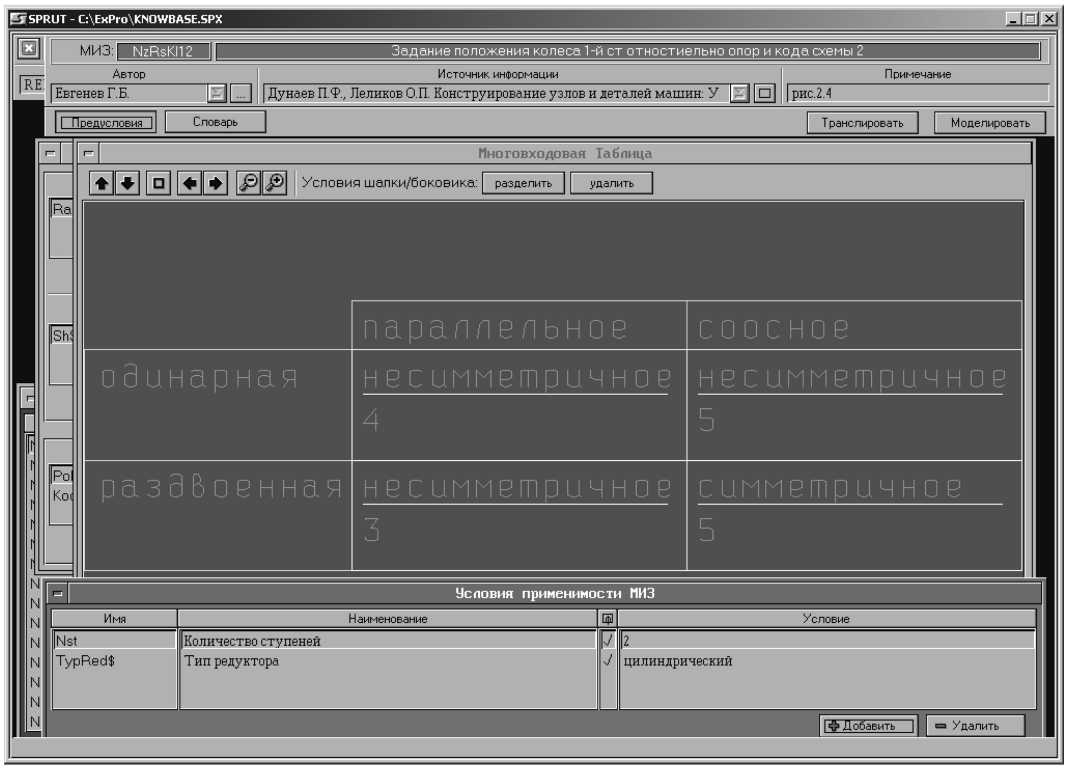


Рис. 4. Интерфейс системы SprutExPro при разработке модуля знаний

и выходной осей" со значениями "параллельное" и "соосное". Выходные переменные "Код схемы передачи 1-й ступени" и "Положение зубчатых колес относительно опор 1-ст" принимают значения в соответствии с таблицей.

На рис. 4 представлен интерфейс системы SprutExPro в процессе разработки описанного модуля знаний.

2. Интеллектуальные методы классов объектов

Возможность скрещивания методов теории программирования и искусственного интеллекта появилась только после того, как было разработано объектно-ориентированное программирование (ООПр) [3]. Это связано с тем, что процедурный, алгоритмический подход в принципе не приемлем для методов искусственного интеллекта. Однако в существующих методах ООПр при разработке методов классов объектов по-прежнему используется алгоритмический подход [3]. Первый шаг в замене процедурного подхода на непроцедурный объектный описан в предыдущем разделе. Чтобы создать полноценный метод объекта следует построить структуру на необходимом множестве модулей знаний.

Согласно таблице, приведенной выше, методы классов объектов следует реализовать на основе продукционных систем. Продукционная система — это способ представления знаний в виде неупорядоченной совокупности продукционных правил, рабочей памяти и механизма логического вывода. Механизм логического вывода обрабатывает неупорядоченную совокупность правил в режиме интерпретации.

Продукционные системы часто использовались при создании экспертных систем (ЭС). При построении ЭС были учтены уроки предшествующих исследований в области искусственного интеллекта, что определило успехи их практического применения. Важнейшим из этих уроков явилось положение, согласно которому мощность ЭС обусловлена в первую очередь мощностью базы знаний, содержащей правила принятия решений, и только во вторую очередь методами логического вывода, основанными на этих правилах. Как показал опыт развития искусственного интеллекта важнее иметь разнообразные специальные знания, а не общие процедуры вывода. Это положение использовано в экспертном программировании, в котором программы, реализующие функции вывода, генерируются применительно к каждому конкретному набору правил, составляющих базу знаний.

Чтобы превратить продукционную систему искусственного интеллекта в систему программирования, разработана методика, при которой осуществляется компиляция как правил вывода, так и экземпляра решателя, осуществляющего вывод на конкретном наборе правил, упорядоченных в соответствии с порядком определения значений переменных.

Общепринятым языком ООПр является UML. В этом языке структуру методов объектов удобно описывать с помощью диаграмм состояний [3].

На рис. 5 представлен пример диаграммы состояний. В экспертном программировании такие модели строятся автоматически на основе анализа

Рис. 5. Диаграмма состояний метода проектирования привода

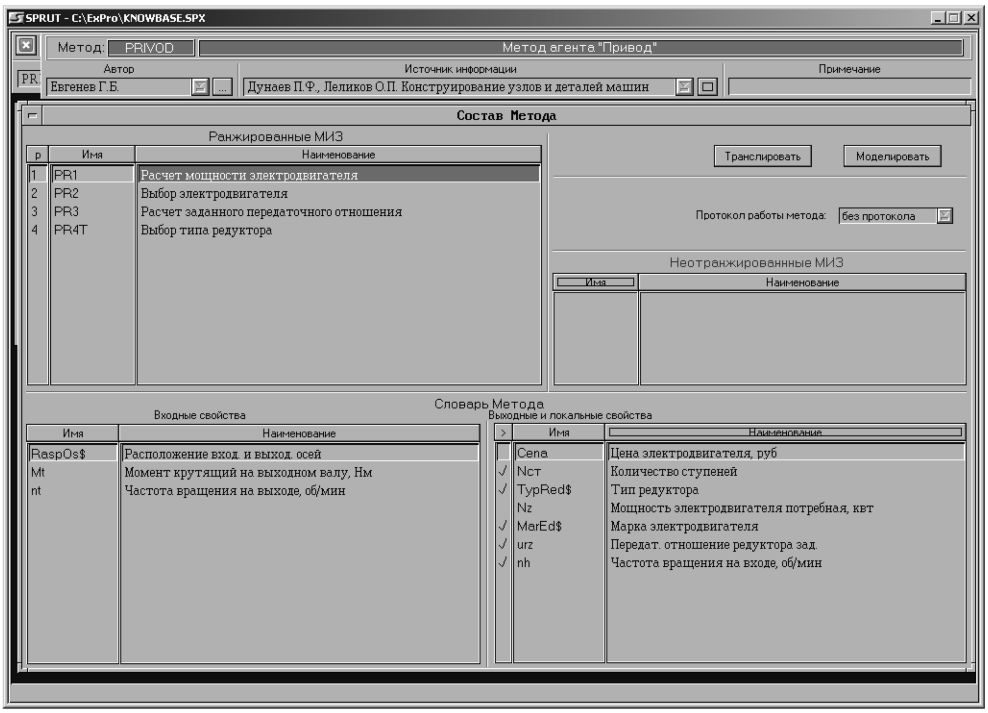


Рис. 6. Интерфейс системы SprutExPro при генерации метода

функциональных связей объект-функций. По сути дела, это означает автоматизацию генерации алгоритмов применительно к конкретному множеству объект-функций. С точки зрения искусственного интеллекта такая диаграмма представляет собой семантическую сеть.

Пример генерации метода в SprutExPro приведен на рис. 6. В левом верхнем окне отображены четыре модуля знаний, из которых состоит метод. Они автоматически упорядочены в необходимой последовательности. В левом нижнем окне приведен автоматически сформированный список входных переменных. В их число входят переменные, не определяемые ни в одном модуле метода. В правом нижнем окне отображены выходные и локальные переменные, т. е. переменные, определяемые в модулях метода. Выходные переменные отмечаются разработчиком галочками. В итоге метод, как и модуль знаний, представляет собой динамическую объект-функцию (см. рис. 2).

3. Интеллектуальные агенты

Согласно принципам ООПр описанные выше методы не могут существовать автономно. Каждый метод является составной частью некоторого объекта. Объект — это абстракция множества предметов реального мира, в которой: 1) все предметы множества (экземпляры) имеют одни и те же характеристики (свойства); 2) все экземпляры подчинены и согласовываются с одним и тем же набором правил поведения. Состояние объекта характеризуется перечнем его свойств и текущим значением каждого из этих свойств, а поведение объекта определяется методом, состоящим из набора правил. При объектно-ориентированном подходе к проектированию программных систем есть две относительно самостоятельные и вместе с тем тесно взаимосвязанные стороны моделирования: статическая и динамическая. Статическое моделирование определяет структуру классов и объектов, а динамическое — их поведение. В предыдущих разделах рассматривались динамические компоненты. В данном и последующем разделах анализируется статическое моделирование.

В теории искусственного интеллекта имеется способ представления знаний, практически эквивалентный понятию "объект", — это фрейм. Фрейм представляет собой логическую запись, каждому полю (слоту) которой соответствуют основные элементы понятия. В формальных фреймовых моделях слотам ставятся в соответствие значения, присоединенные процедуры или другие фреймы. Фреймы используются для описания объектов, событий, ситуаций, прочих понятий и взаимосвязей между ними. Они могут иметь две разновидности — фрейм-прототип и экземпляр фрейма. Фрейм-прототип — это фрейм, в котором значения слотов не определены. Экземпляр фрейма представляет собой фрейм, в котором определены значения слотов.

Таким образом, фрейм изначально предназначался для описания объектов. В ООПр имеются понятия класса и экземпляра [3]. Класс — это описание множества объектов, имеющих общие атрибуты, операции, отношения и семантику. Экземпляр представляет собой конкретную материализацию абстракции. Эта сущность обладает состоянием, в котором запоминаются результаты операций, реализуемых методами.

Из приведенных выше определений следует, что имеются две пары для скрещивания — фрейм-прототип с классом объектов и экземпляр фрейма с экземпляром объекта. Для превращения фрейма-прототипа в класс объектов необходимо разделить слоты на две группы: слоты-атрибуты для хранения значений свойств и слоты-методы для указания на присоединенные процедуры. Экземпляр фрейма практически эквивалентен экземпляру объекта, в котором определены значения атрибутов.

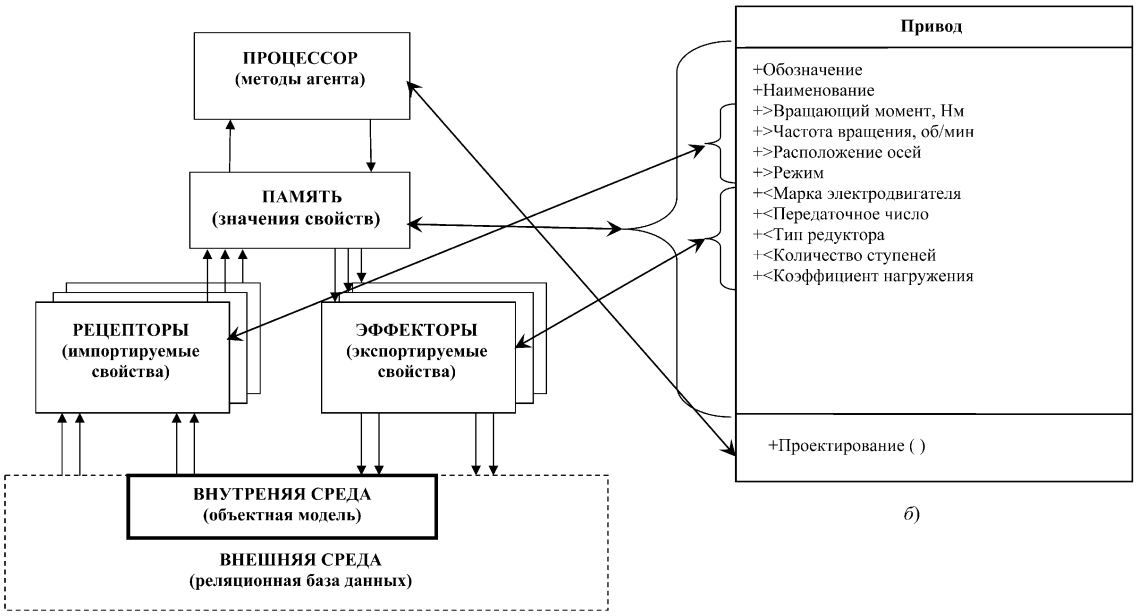
Обобщенная модель класса искусственных агентов приведена на рис. 7, *а* [4]. Любой агент представляет собой открытую систему, помещенную в некоторую среду. Этой средой является проект, формируемый в базах данных, в качестве которых целесообразно использовать базу данных объектного типа для представления модели сущностей (внутренняя среда) и реляционную базу данных для поиска информации (внешняя среда). Внешняя среда, как правило, является сетевой.

Свойства агента могут принадлежать трем различным категориям: импортируемые, экспортируемые и внутренние. Импортируемые свойства являются рецепторами агента, формирующими его систему восприятия. Экспортируемые свойства агента являются его эффекторами, функция которых состоит в воздействии на среду, т. е. на состояние проекта.

Свойства агента всех трех категорий образуют его память, в которой хранится текущее состояние агента.

Процессор агента формирует его методы, обеспечивающие объединение и переработку разнородных данных, выработку соответствующих реакций на информацию о состоянии среды (проекта), принятие решений о выполнении тех или иных действий [4]. В целом процессор определяет поведение агента. Его можно наблюдать, используя инспектор модели агента, с помощью которого пользователь следит за состоянием свойств агента, либо в графическом окне, в котором отображаются результаты.

На рис. 7, *б* приведена модифицированная модель класса объектов в нотации UML. Из рисунка



а) *б*)

Рис. 7. Архитектура агента

следует, что процессор агента соответствует операциям класса, память агента — значениям атрибутов класса. Несоответствие заключается в том, что агент имеет импортируемые и экспортируемые свойства, а в нотации класса UML это не предусмотрено. Знаком плюс в UML отмечаются свойства, видимые извне. Простейшим решением этой проблемы может быть пометка импортируемых свойств знаком >, а экспортируемых — знаком <, как это показано на рис. 7, *б*. При этом агент принимает форму статической объект-функции.

4. Многоагентные системы

Решение сколько-нибудь сложной задачи требует использования многоагентной системы.

Многоагентные системы принадлежат к классу интеллектуальных систем распределенного решения задач. Их основу составляет иерархическая метасистема агентов, проектируемая сверху вниз.

Многоагентные системы содержат следующие основные компоненты [4]:

- множество системных единиц, в котором выделяются подмножества активных единиц — агентов, манипулирующих подмножеством других агентов и пассивных единиц — объектов;
- среду, т. е. некоторое пространство, в котором существуют агенты и объекты;
- множество задач (функций, ролей), которые поручают агентам;
- множество отношений (взаимодействий) между агентами;
- множество организационных структур (конфигураций), формируемых агентами и объектами;
- множество действий агентов (например, различных операций над другими агентами и объектами).

На основе изложенного выше формальную модель многоагентной системы (МАС) можно представить так:

$$MAS = (A,\ E,\ R,\ ORG),$$

где A — множество агентов; $E = \{e\}$ — среда, в которой находится данная МАС; R — множество взаимодействий между агентами; ORG — множество базовых организационных структур, соответствующих конкретным функциям (ролям) агентов и устанавливающих отношения между ними.

В многоагентных системах множество агентов A формируется из сущностей, подлежащих разработке. В качестве среды $E = \{e\}$ выступает разрабатываемый проект, состоящий из множества экземпляров e из классов задействованных в проекте агентов. Задачи, порученные агентам, определяются инкапсулированными в них методами. Методы обеспечивают решение задач, порученных агентам, и выполнение операций, оказывающих воздействие на другие агенты.

Множество взаимодействий R между агентами определяется ребрами графа экспорта и импорта свойств агентов и объектов. Эти взаимодействия носят как вертикальный, так и горизонтальный характер. Вертикальные взаимодействия осуществляются между агентами, связанными друг с другом по иерархии организационной структуры ORG, а горизонтальные взаимодействия — между иерархически не связанными агентами.

Наконец, организационная структура ORG в многоагентной системе представляет собой ие-

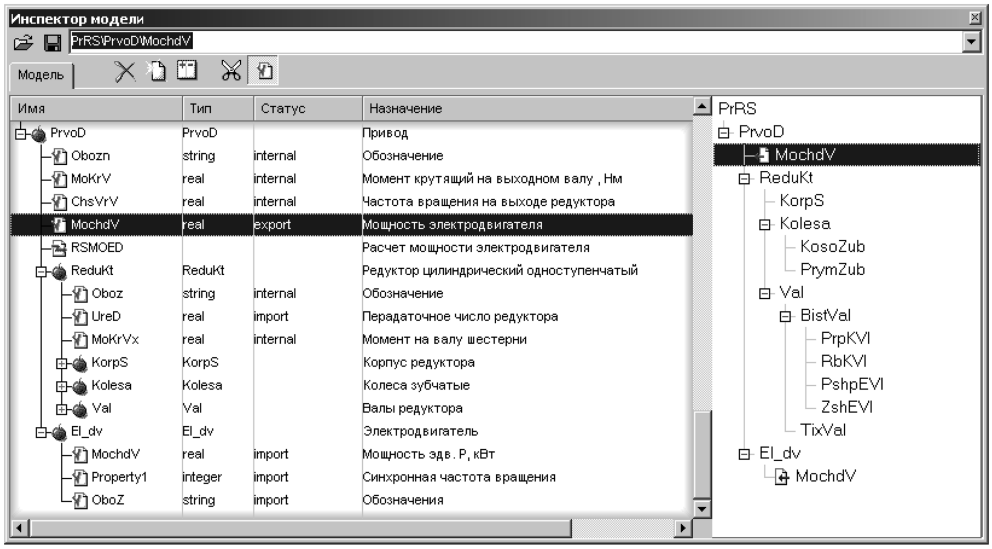


Рис. 8. Генерация многоагентной метасистемы в SprutX

рархическую метасистему, моделируемую И/ИЛИ графом. Связки типа И описывают отношения класса "целое—часть", а связки типа ИЛИ — отношения класса "род—вид". Последние используются при принятии решений в процессе структурного синтеза.

Для моделирования многоагентных систем в теории искусственного интеллекта наиболее подходят семантические сети. Семантическая сеть — структура данных, состоящая из узлов, соответствующих понятиям, и связей, указывающих на взаимосвязи между узлами. В ООПр этому понятию при представлении знаний практически однозначно соответствует диаграмма классов, а при представлении результатов — диаграмма объектов. Обе диаграммы содержат множество сущностей (агентов или экземпляров) и организационную структуру, которая в диаграмме классов моделируется связками типа И/ИЛИ, а в диаграмме объектов связками типа И.

Недостатком диаграмм классов для моделирования многоагентных систем является отсутствие возможностей моделирования взаимодействий агентов при экспорте и импорте свойств.

Для реализации многоагентной методологии разработана инструментальная среда SprutX. С ее помощью генерируется многоагентная метасистема (рис. 8).

На рис. 8 представлен фрагмент метасистемы проектирования электромеханического привода. Корневым агентом здесь является "Привод", который имеет входные свойства "Обозначение", "Момент крутящий на выходном валу, Н·м", "Частота вращения на выходе редуктора" и выходное свойство "Мощность электродвигателя", которое экспортируется в объект "Электродвигатель". Привод имеет метод "Расчет мощности электродвигателя". Привод включает агенты "Редуктор цилиндрический одноступенчатый", "Корпус редуктора", "Колеса зубчатые" и "Валы редуктора".

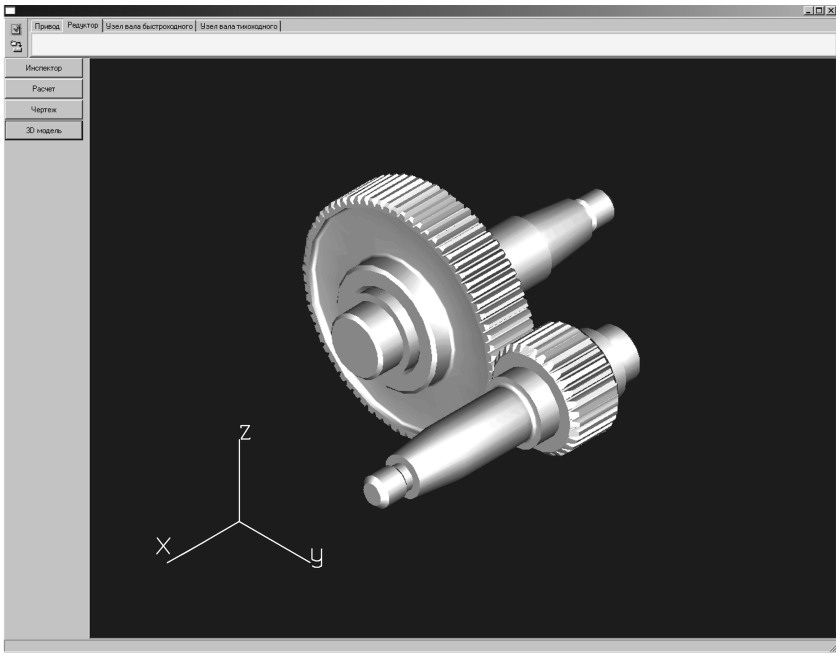


Рис. 9. Интерфейс пользователя в полуавтоматической системе проектирования электромеханических приводов

На рис. 9 представлен интерфейс пользователя в полуавтоматической системе проектирования электромеханических приводов. Верхние закладки отображают иерархическую структуру метасистемы: привод включает редуктор, который состоит из узлов вала быстроходного и тихоходного. На левом поле отображены методы выбранного с помощью закладки агента — в данном случае редуктора. Редуктор, как и любой другой объект, имеет кнопку "Инспектор", с помощью которой можно увидеть текущие значения свойств. С помощью метода "Расчет" проводится кинематический и силовой расчет редуктора. Вывод в правое графическое поле чертежа редуктора по результатам расчета проводится

методом "Чертеж", с помощью которого запускается параметризованная графическая база знаний. На рис. 9 представлен результат запуска метода "3D-модель"

Заключение

Разработаны теоретические основы построения интеллектуальных прикладных систем с использованием методов теории многоагентных систем. Как показал опыт применения изложенной теории на практике, многоагентные системы обеспечивают реализацию основных концепций создания современных систем: интеграцию, интеллектуализацию и индивидуализацию.

На основе разработанной теории с использованием комплекса инструментальных средств операционной среды "Спрут", создана новая информационная технология генерации многоагентных прикладных систем практически без использования труда прикладных программистов. С помощью этой новой информационной технологии, обеспечивающей резкое сокращение трудоемкости создания специализированных систем, разработаны системы проектирования асинхронных электродвигателей и проектирования технологических процессов СПРУТ ТП.

Список литературы

1. **Евгенев Г. Б.** Системология инженерных знаний: Учеб. пособие для вузов. М.: Изд-во МГТУ им. Н. Э. Баумана, 2001. 520 с.

2. **Евгенев Г. Б., Кобелев А. С., Борисов С. А.** Технология экспертного программирования // Информационные технологии. 2002. № 3. С. 2—9.

3. **Буч Г., Рамбо Дж., Джекобсон А.** Язык UML. Руководство пользователя / Пер. с англ. А. А. Слинкин. 2-е изд., стер. М.: ДМК Пресс; СПб.: Питер, 2004. 432 с.

4. **Тарасов В. Б.** От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. М.: Эдиториал УРСС, 2002. 352 с.

УДК 004.031.43

Л. А. Жуков, канд. техн. наук, доц.,
О. В. Корчевская, ст. преподаватель,
Сибирский государственный
технологический университет, г. Красноярск

Метод плоскостей: численный эксперимент для задач двух- и трехмерной ортогональной упаковки

Представлена модель трехмерной ортогональной упаковки. Описан общий подход для решения задач двух- и трехмерной упаковок (метод плоскостей). Приведены результаты численного эксперимента, демонстрирующие преимущества данного метода для некоторого класса задач.

Ключевые слова: *оптимизация, ортогональная упаковка, двухмерная упаковка, трехмерная упаковка, методы ортогональной упаковки, модель ортогональной упаковки.*

Введение

Задача раскроя/упаковки является обобщением нескольких известных задач: упаковки в контейнеры, упаковки прямоугольников в одну полосу или листы, распила пиломатериала, *m*-процессорном расписании и др.

В связи с *NP*-сложностью данного класса задач [1] наибольший интерес вызывают методы, позволяющие достичь глобального экстремума — алгоритмы полного перебора. Однако в задачах принятия комбинаторных решений применительно к сложным оптимизационным задачам, для которых характерна большая размерность исходных данных, возможности полного перебора весьма ограничены.

В связи с этим на практике стали применяться, так называемые, "эвристические" алгоритмы, при разработке которых используются интуитивные соображения, не подкрепленные соответствующим математическим обоснованием. Они сильно зависят от специфики задачи, предполагают отказ от поиска оптимального решения и нахождение "хорошего" решения за приемлемое время.

1. Модель трехмерной упаковки

Под задачами раскроя/упаковки понимается широкий класс задач, допускающих различное прикладное толкование, общим для которых является наличие двух групп объектов, между которыми устанавливается и оценивается соответствие [2, 3]. В качестве основной принято рассматривать следующую задачу упаковки: имеются малые элементы, их необходимо разместить без взаимного перекрытия внутри больших объектов так, чтобы заданная целевая функция достигала экстремума.

В работе [2] приведена классификация задач по факторам задания объектов и способам их разделения на элементы: задача упаковки прямоугольников в полосу, в контейнеры, в открытую область. Кроме того, если допустимыми являются только сквозные линии (разрезы), параллельные кромкам материала, то такие задачи называют *гильотинными*.

Входная информация для задач трехмерного ортогонального раскроя/упаковки в общем виде может быть задана в следующем виде:

$$< W, L, H, k, u, w, l, h, m, b, \varepsilon, \gamma, g, V >,$$

где $W = (W_1, W_2, ..., W_k)$ — ширина, $L = (L_1, L_2, ..., L_k)$ — длина, $H = (H_1, H_2, ..., H_k$ — высота параллелепипедов; k — число параллелепипедов; $u = (u_1, u_2, ..., u_k)$ — число параллелепипедов определенного вида; $w = (w_1, w_2, ..., w_m)$ — ширина, $l = (l_1, l_2, ..., l_m)$ — длина, $h = (h_1, h_2, ..., h_m)$ — высота объектов; m — число типов объектов; $b = (b_1, b_2, ..., b_m)$ — число объектов определенного типа; ε — признак направления: $\varepsilon = 1$, если объекты можно поворачивать на 90°, $\varepsilon = 0$ — в противном случае; γ — признак гильотинности: $\gamma = 1$, если задачу решают с учетом признака гильотинности, и полагают равным 0 в противном случае; $g = (g_1, g_2, ..., g_m)$ — масса объектов; V — набор технологических ограничений; $c = \sum_{j=1}^{k} u_k$ — общее число параллелепипедов; $n = \sum_{i=1}^{m} b_i$ — общее число объектов.

Выходной информацией является карта раскроя, представленная в виде следующего набора:

$$< X, Y, Z, S, E >,$$

где $X = (x_1, x_2, ..., x_n)$, $Y = (y_1, y_2, ..., y_n)$, $Z = (z_1, z_2, ..., z_n)$ — векторы минимальных координат объектов; $S = (s_1, s_2, ..., s_n)$ — номера заполненных объектов (s_i — номер параллелепипеда, в который упакован i-й объект); $E = (e_1, e_2, ..., e_n)$ — признаки поворота.

Зададим k — коэффициент раскроя (процентное отношение суммарного объема всех упакованных объектов к занятому объему), а τ — время решения задачи.

Таким образом, если задан параллелепипед неограниченной высоты ($H = \infty$), необходимо найти такое отображение Ω

$$< X, Y, Z, S, E > = \Omega(< W, L, H, k, u, m, w, l, h, b, \varepsilon, \gamma, g, V >), \quad (1)$$

$$\begin{matrix} L \to \min \\ k \to 100 \\ \tau \to \min \end{matrix}$$

которое преобразует входные данные в выходные, причем соблюдаются следующие условия:

- ортогональное размещение объектов в контейнерах;
- неперекрытие объектов между собой;
- неперекрытие объектами граней параллелепипедов.

Для задачи с фиксированными сторонами параллелепипедов, необходимо найти их минимальное число S, т. е. к уравнению (1) добавляется еще одно условие $S \to \min s_j$.

На данную систему оказывают влияние следующие факторы:

- способ (метод) решения задачи;
- приоритетный список (последовательность объектов);
- параметры оптимизации.

Данная модель трехмерной ортогональной упаковки легко может быть преобразована к двумерной и одномерной.

В работах [2, 3] подробно приведены математические модели и обзор методов решения задач одно- и двухмерного раскроя/упаковки.

Вероятно, можно предложить задачу упаковки для пространства большей размерности, например четырехмерную, учитывающую ресурс времени. Предполагается недеструктивное размещение объектов с освобождением какого-либо ресурса.

Способ (метод) решения задачи. Фундаментальные научные разработки в области решения задач раскроя/упаковки в условиях массового производства принадлежат Л. В. Кантаровичу и В. А. Залгаллеру. Результаты дальнейших исследований в этой области отражены в работах Э. А. Мухачевой, А. Ф. Валеевой, И. П. Норенкова, В. М. Картака, А. С. Филипповой, А. В. Чиглинцева, И. В. Романовского, В. А. Кузнецовой и др.

В работах [2, 4] выделена тенденция развития методов решения задач упаковки, которые развиваются в двух направлениях. Для первого направления характерно то, что поиск локально-оптимальных решений ведется в некоторой окрестности исходного решения с применением декодеров — алгоритмов, которые по закодированному решению восстанавливают эскиз упаковки.

Другим направлением является разработка конструктивных методов, имеющих дело с покомпонентным построением упаковки, а именно: к частично построенной упаковке добавляется новый компонент до тех пор, пока упаковка не будет построена полностью.

В настоящее время разработано множество эвристических методов, от однопроходных до метаэвристик (поиск с запретами, имитация отжига, "муравьиная колония", генетические алгоритмы и др.), что затрудняет их выбор и оценку качества полученных решений.

Представляют интерес унифицированные методы, позволяющие на общей теоретической основе создать математическое обеспечение для решения задач одномерной, прямоугольной и параллелепипедной упаковки/раскроя. Такой подход приведен в работе [4]: на базе модифицированного метода решения задачи "0—1 рюкзак" создано инвариантное математическое обеспечение для решения задач n-мерной упаковки ($n = 1, 2, 3$).

Формирование приоритетного списка. Эффективность применения разработанных методов напрямую зависит от приоритетного списка — очередности размещения прямоугольников.

В зависимости от стратегии установления порядка упаковки выделяют два основных класса: *offline* — когда заранее известны все объекты и их упорядочивают согласно некоторому критерию, *online* — когда объекты поступают в заданном порядке.

Для задачи двумерной ортогональной упаковки обычно приоритетный список формируют по невозрастанию площади заготовок. Предлагаются и другие способы. В работе [5] показана возможность применения алгоритма "метод ветвей и границ" и внесены предложения, позволяющие применить алгоритм более эффективно. В частности, поскольку время решения задачи зависит от приоритетного списка, И. В. Романовским предложено ранжировать прямоугольники по убыванию значений $\lambda_i = w_i + l_i + \sqrt{w_i l_i} + w_i^2/W + l_i^2/h$, $i = 1, 2, ..., m$.

Ю. Г. Стоян и М. В. Новожилова в работе [6] предложили методы поиска глобального минимума задачи двухмерной упаковки, если рассмотреть описание размещения прямоугольников относительно их полюсов системой из $4m$ линейных неравенств. Верхняя оценка числа уравнений равна m^{2m}. В работе также отмечается, что в случае использования приоритетного списка, поиск глобального минимума сводится к перебору только $m!/2$ перестановок вариантов размещения объектов.

В работе [7] предложен способ формирования приоритетного списка с использованием нейросетевых технологий. В частности, для задачи двухмерной ортогональной упаковки с использованием нейронной сети Хопфилда была сконструирована энергетическая функция и определена матрица весовых коэффициентов, что позволило получить приоритетный список квазиоптимальной упаковки.

Параметры оптимизации. Использование однопроходных методов, как правило, приводит к нахождению локальных решений. В целях исключения полного перебора для улучшения полученного решения, ряд авторов включает в свои алгоритмы схемы перебора, которые зависят от специфики алгоритма. И здесь следует найти компромисс между числом параметров, по которым проводится оптимизация, и временем решения задачи.

В общем случае для задачи трехмерной ортогональной упаковки можно предложить следующее:

1) сортировка входных данных: по объему, по максимуму значения каждой из сторон объектов с учетом переворотов;

2) приоритетное заполнение по какой-либо оси (X, Y или Z) параллелепипеда; это имеет смысл при различающихся параметрах контейнеров;

3) анализ возможности объединения и удаления из дальнейшего рассмотрения пустот;

4) заполнение двумя способами — с созданием слоев по осям выбранных направлений (когда следующая коробка не может выйти за пределы слоя), либо без них, когда слоем является весь параллелепипед, т. е. ограничениями для заполнения являются только размеры контейнера.

2. Метод плоскостей

В данном методе предполагается осуществлять заполнение контейнера объектами (коробками) с созданием слоев и без них, в результате чего образуются плоскости пустот, находящихся на разной глубине. Они описываются пятью параметрами: координатами плоскости (y_1, z_1), (y_2, z_2); глубиной x и флагом (0 — слой не заполнен, 1 — полностью заполнен, -1 — игнорировать плоскость при заполнении данного слоя).

Первая коробка из приоритетного списка разбивает исходный параллелепипед на три плоскости, одна из которых заполнена.

Дальнейшее заполнение может осуществляться по различным осям, что оказывает влияние на конечное решение. В связи с этим поочередно выбираются различные приоритетные направления.

Если алгоритм предусматривает формирование слоев, следующей выбирается коробка, которая помещается на данной плоскости полностью, не выходя за границу слоя. В зависимости от размера второй коробки и размеров плоскости, в которую она помещена, происходит дальнейшее разбиение плоскости на одну или три.

По мере заполнения параллелепипеда происходит изменение координат пустых плоскостей и анализ возможности их объединения. Эта процедура повторяется до тех пор, пока все пустые плоскости будут иметь флаг, равный 1 (полное заполнение), либо 1 и -1 (игнорировать). Это значит, что слой заполнен и невозможно больше поставить ни одной из оставшихся в списке коробок. Если остались коробки и есть место в контейнере, то происходит создание нового слоя, в котором прежние слои могут также дополняться.

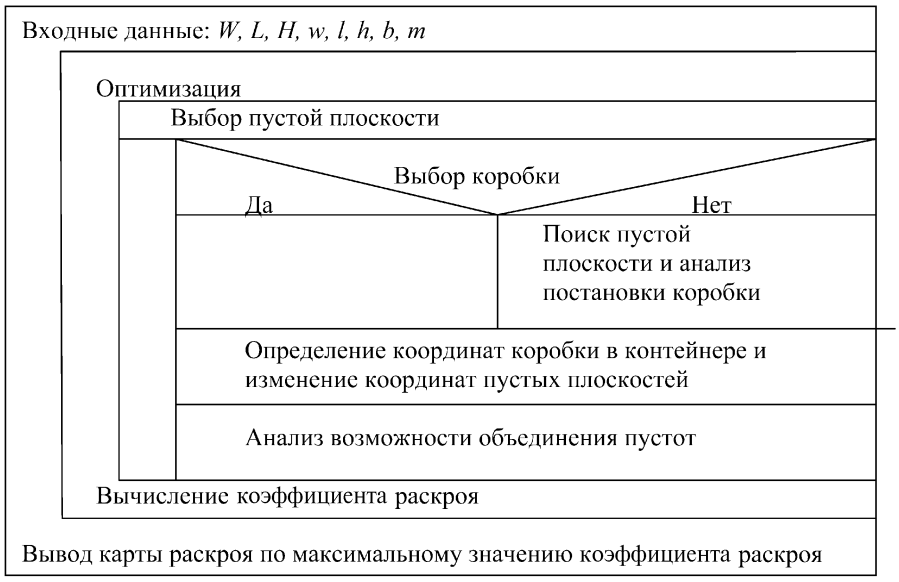


Рис. 1. Структурограмма метода плоскостей без создания слоев

Процедура с использованием метода без формирования слоев отличается от рассмотренной выше тем, что толщина слоя принимается равной длине контейнера, т. е. отсутствует блок, отвечающий за формирование слоев. Заполнение может осуществляться по трем направлениям — задняя стенка справа налево (по осям Z, Y), правая стенка (по осям Z, X), либо дно параллелепипеда (по осям X, Y).

На рис. 1 приведена структурограмма метода плоскостей для случая, когда заполнение осуществляется без создания отдельных слоев.

На структурограмме под оптимизацией понимается перебор входных параметров, условий заполнения по приоритетным направлениям.

Вычислительная сложность метода плоскостей составляет $O(n^5)$.

3. Численный эксперимент для задачи трехмерной ортогональной упаковки

Для оценки эффективности разработанного метода при решении задачи параллелепидной упаковки была проведена серия расчетов на основе известной методики Г. Вешера.

За основу разбиения было взято следующее: нижнее ограничение длины предметов — v_1, верхнее ограничение длины предметов v_2 ($v_1 W \leqslant l_i \leqslant v_2 W$, $i = 1, \ldots, n$); нижнее ограничение ширины предметов w_1, верхнее ограничение ширины предметов w_2 ($w_1 W \leqslant w_i \leqslant w_2 W$, $i = 1, \ldots, n$); нижнее ограничение высоты предметов η_1, верхнее ограничение высоты предметов η_2 ($\eta_1 W \leqslant \eta_i \leqslant \eta_2 W$, $i = 1, \ldots, n$).

Серия № 1. Целью данного эксперимента была проверка метода плоскостей и определение на разной группе предметов необходимости включения в оптимизацию разбиения на слои.

Выполнены численные эксперименты, параметрами которых являлись высота основания параллелепипеда $H = 150$; ширина $W = 100$; число предметов $n = 50$, 100, 200, 250, 500, 1000. Предметы были отсортированы по классам: "мелкие" ($v_1 = 0{,}1$, $v_2 = 0{,}2$; $w_1 = 0{,}1$, $w_2 = 0{,}2$; $\eta_1 = 0{,}2$, $\eta_2 = 0{,}25$), "средние" (v_1, $= 0{,}2$, $v_2 = 0{,}4$; $w_1 = 0{,}2$, $w_2 = 0{,}4$; $\eta_1 = 0{,}2$, $\eta_2 = 0{,}25$) и "разнородные" ($v_1 = 0{,}1$, $v_2 = 0{,}3$; $w_1 = 0{,}1$, $w_2 = 0{,}2$; $\eta_1 = 0{,}2$, $\eta_2 = 0{,}25$).

Для каждого класса задач было сгенерировано по 10 тестовых примеров, усредненные значения коэффициента раскроя приведены на рис. 2 (см. третью сторону обложки).

Таким образом, вне зависимости от класса предметов и их количества, способ дальнейшего заполнения параллелепипеда следует выбирать с формированием слоев.

На рис. 3 (см. третью сторону обложки) приведены усредненные значения коэффициентов раскроя различного типа деталей для алгоритма с формированием слоев.

При трехмерном раскрое метод плоскостей показывает лучшие результаты при большом числе деталей и в основном для разнородных предметов. Время решения задачи на ПК (Celeron (R) CPU 3,2 ГГц, 1 Гбайт ОЗУ) от нескольких секунд до 2 мин (для $n = 1000$).

4. Численный эксперимент для задачи двухмерной упаковки

Метод плоскостей с некоторой модификацией может быть применен и для решения задачи двухмерной упаковки.

Для оценки эффективности разработанной методики и программного обеспечения были использованы тестовые примеры из работы [4] на основе методики Г. Вешера и данные из электронной библиотеки OR-library, размещенной на сайте http://mscmga.mc.ic.ac.uk/info.html.

В электронной библиотеке OR-library приведены, так называемые, безотходные примеры упаковки, разработанные E. Hopper, для которых заранее известно оптимальное значение длины занятой полосы и оптимальный приоритетный список, отвечающий безотходной упаковке.

Серия № 2. Были протестированы задачи E. Hopper из серии N, содержащие по семь групп

Задачи	n	Значение среднего коэффициента раскроя, %	
		Оптимальный приоритетный список	Измененный приоритетный список
N1a—N1e	17	100	92,58
N2a—N2e	25	100	92,65
N3a—N3e	29	100	94,14
N4a—N4e	49	100	93,98
N5a—N5e	73	100	94,9
N6a—N6e	97	100	96,71
N7a—N7e	197	100	97,04

примеров. Каждый из примеров серии включает пять различных вариантов. В тестовых примерах в качестве входной информации, кроме размеров прямоугольников и ширины полосы, подавался оптимальный приоритетный список и измененный приоритетный список (последовательность прямоугольников изменена случайным образом). Полученные средние значения коэффициента раскроя для заданного и измененного приоритетного списка приведены в таблице.

Как видно из таблицы, метод плоскостей обладает свойством "реставрации", т. е. с его помощью по известному оптимальному приоритетному списку формируется оптимальная безотходная упаковка. В реальных условиях в связи с отсутствием в алгоритме схем полного перебора данный метод позволяет получить упаковку с коэффициентом раскроя более 90 %.

Серия № 3. Целью данного эксперимента было сравнение метода плоскостей с алгоритмом рандомизированного динамического перебора (DSR) для решения задачи прямоугольной упаковки. Применялась методика Г. Вешера. Были сгенерированы следующие классы задач: "малые предметы" ($v_1 = 0{,}05$, $v_2 = 0{,}1$; $w_1 = 0{,}1$, $w_2 = 0{,}15$); "средние предметы" ($v_1 = 0{,}25$, $v_2 = 0{,}35$; $w_1 = 0{,}35$, $w_2 = 0{,}45$) и "малые и большие предметы" ($v_1 = 0{,}05$, $v_2 = 0{,}25$; $w_1 = 0{,}05$, $w_2 = 0{,}95$). Для каждого класса задач было сгенерировано по 10 тестовых примеров. Критерием оценок также выступал коэффициент раскроя, значения которого приведены на рис. 4 (см. третью сторону обложки).

Согласно данным, приведенным в работе [4], значения коэффициентов раскроя для класса "малых предметов" составили 92...96 %; "средних" — 98...100 %; "малых и больших" — 93...95 %. Таким образом, метод плоскостей показывает не худшие результаты для классов "малых" и разнородных ("малых и больших") предметов.

Заключение

Авторы считают, что в данной работе на основании проведенных численных экспериментов можно сделать следующие выводы.

1. Экспериментальные данные показали, что при решении задачи трехмерной упаковки следует использовать процедуры с формированием слоев. Использование приоритетного заполнения по различным осям повышает качество получаемых решений.

2. При тестировании предложенного метода параллелепидной упаковки, наилучшие значения коэффициента раскроя (85...90 %) были получены в классе средних предметов.

3. Как для задач двух-, так и трехмерной упаковки, метод плоскостей показывает лучшие результаты при увеличении числа объектов, причем время вычисления составляет от нескольких секунд до 2 мин.

4. Метод плоскостей обладает свойством реставрации, т. е. восстанавливает карту раскроя по оптимальному приоритетному списку.

5. Сравнение усредненных коэффициентов раскроя метода плоскостей с одним из лучших алгоритмов — методом рандомизированного динамического перебора (DSR), позволяет сделать вывод, что в классах малых и разнородных предметов метод плоскостей не уступает DSR по эффективности для некоторого класса задач.

Список литературы

1. **Мухачева А. С., Валеева А. Ф., Картак В. М.** Задачи двухмерной упаковки в контейнеры: новые подходы к разработке методов локального поиска оптимума. М.: Изд. МАИ, 2004. — 193 с.

2. **Мухачева Э. А., Мухачева А. С., Валеева А. Ф., Картак В. М.** Модели и методы решения задач ортогонального раскроя и упаковки: аналитический обзор и новая технология блочных структур // Информационные технологии. 2004. № 5. Приложение. 32 с.

3. **Гери М. П., Джонсон Д. С.** Вычислительные машины и трудно разрешимые задачи. М.: Мир, 1982. 416 с.

4. **Валеева А. Ф.** Конструктивные методы решения задач ортогональной упаковки и раскроя: Автореф. дис. ... на соискание степени д-ра техн. наук. Уфа, 2006. 32 с.

5. **Романовский И. В.** Алгоритмы решения экстремальных задач. М.: Наука, 1977. 88 с.

6. **Стоян Ю. Г., Яковлев С. В.** Математические модели и оптимизационные методы геометрического проектирования. Киев: Наукова думка, 1986. 268 с.

7. **Корчевская О. В., Жуков Л. А., Больсявичус А. В.** Конструирование функции энергии сети для задачи ортогональной упаковки // Компьютерная интеграция производства и ИПИ-технологии: Сб. материалов III Всероссийской научно-практической конференции. Оренбург: ИПК ГОУ ОГУ, 2007. С. 352—357.

УДК 001.891.57:621.1.016.4

А. П. Мельник, аспирант,
С. Н. Чувашев, д-р. физ.-мат. наук, проф.,
"МАТИ"-РГТУ имени К. Э. Циолковского,
И. Г. Зорина, доц.,
МГТУ им. Н. Э. Баумана
andrei_melnik@mail.ru

Моделирование процессов теплопередачи для определения реальных теплофизических характеристик зданий

Рассматривается разработанный автоматизированный программный комплекс, который позволяет вычислить необходимые для определения теплофизических характеристик здания данные на основе измерения температур и яркостных температур. Математическая модель теплообмена базируется на законах радиационного переноса и газодинамики с учетом теории подобия.

Ключевые слова: *теплопередача, теплообмен, контроль теплофизических характеристик ограждающих конструкций зданий.*

Введение

Определение комплекса теплофизических параметров реальных ограждающих конструкций зданий и сооружений представляет значительный практический интерес. Во-первых, важно знать теплоизоляционные свойства для того, чтобы определять необходимую для отопления тепловую энергию и при необходимости принять меры к утеплению зданий и сооружений. Во-вторых, необходимы сведения о времени вымораживания домов при аварийном отключении теплоснабжения в зимний период: если время ремонта окажется больше времени замерзания теплоносителя, то произойдет дополнительный аварийный выход из строя теплосети дома с тяжелыми последствиями; в то же время крайне нежелательно и проведение лишних трудоемких и длительных операций по сливу и последующему заливу дорогостоящего теплоносителя в систему отопления.

Теплоотвод через крышу связан с внутренним и трансграничным движением воздуха (вентиляцией) и управляется отдельно; теплоотвод через фундамент обычно мал. Наиболее важно знать теплофизические свойства вертикальных ограждающих конструкций зданий и сооружений (стен, окон, дверей и др.).

В 2006 г. принят федеральный закон, который обязывает иметь энергетический паспорт, содержащий, в частности, количественные данные по теплофизическим свойствам зданий, на каждое сдаваемое в эксплуатацию здание. В ряде случаев это требование распространяется и на давно построенные здания. Паспортные или рассчитанные по конструкционным чертежам данные часто оказываются слишком неточными для решения указанных задач: они не учитывают наличие дефектов строительства, отклонений от проекта, намокания и/или оседания теплоизоляции и т. п.

Теплофизические свойства в принципе можно определить, проведя ряд измерений (температуры и/или тепловых потоков и характеристики поля излучения), описав процессы теплообмена для конвективного, кондуктивного и радиационного механизмов [1—6] и решив обратную задачу нахождения реальных свойств ограждающих конструкций. В общем случае это достаточно сложная задача.

В настоящее время применяются упрощенные методики, основанные на нормативных документах (СП 23-101—2000, СНиП 23-02—2003, СанПиН 2.1.2.1002—00 "Санитарно-эпидемиологические требования к жилым зданиям и помещениям"; ГОСТ Р 51379—99 и др.) В них вместо научно обоснованных закономерностей процессов теплопереноса, учитывающих физические свойства конструкций, для простоты применяются сильно упрощенные формулы и эмпирические коэффициенты. В результате ряд важных физических эффектов, влияющих на теплоперенос, учитывается неточно или не учитывается совсем. Так, не учитываются (или слишком неточно, с ошибками до нескольких раз): перенос энергии излучением, в частности, игнорируется отклонение степени черноты от единицы [4—6]; встречные потоки излучения, влияние облачности (известно, как резко различаются скорости охлаждения поверхностей в ясные и облачные ночи); влияние масштабного фактора на интенсивность конвективного теплообмена, что дает ошибки до 60—80 % и более [3] и др. В результате использование упрощенных методик может привести к большим ошибкам.

Правда, их достоинством является простота реализации, все вычисления могут проводиться на калькуляторе. Однако распространение электронной техники и информационных технологий

устраняет необходимость упрощений, снижающих надежность определения теплофизических свойств ограждающих конструкций. Современные методы вычислительной математики позволяют применять более сложные, но и значительно более надежные методы описания тепловых потоков.

В данной статье описаны методика, основанная на относительно более подробном учете физических законов теплообмена, и программный комплекс, обеспечивающий автоматизированное определение теплофизических свойств вертикальных ограждающих конструкций и предсказание их поведения при аварийном отключении теплоснабжения при заданных погодных условиях.

Решение задачи

Методика и программы разделяются на две части: в первой определяются теплофизические свойства вертикальных ограждающих конструкций, а во второй на основе полученных данных рассчитывается динамика остывания здания при заданных погодных условиях.

Методика первой части базируется на решении обратной задачи теплообмена. Ограждающие конструкции представляются при этом как совокупность участков различных типов: стены одной, другой, третьей и т. д. конструкции (различные материалы, толщина, цвет поверхности), окна, витражи, двери и т. п. Эти типы участков ограждающих конструкций с точки зрения тепловых потерь представляют собой параллельные термосопротивления тепловым потокам изнутри здания в окружающую среду (рис. 1). Теплопередача рассчитывается отдельно для каждого типа участков ограждающих конструкций с учетом: тепловой конвекции и переноса излучения от внутренней части здания к внутренней поверхности стенки; кондукционного теплопереноса (теплопроводности) через стенку и тепловой и вынужденной (под действием ветра) конвекции и переноса излучения от внешней поверхности стенки здания в окружающую среду.

Таким образом, при расчете локального теплового потока учитываются три указанные последовательные термосопротивления тепловым потокам изнутри здания в окружающую среду (рис. 1). Основные уравнения модели для каждого типа участков ограждающих конструкций представляют собой выражения равенства потока теплоты через три последовательных термосопротивления и соотношение, связывающее истинную и яркостную температуры на внешней поверхности и соответствующую степень черноты. Из уравнений модели, погодных условий и результатов измерений для каждого типа участков ограждающих конструкций получается и численно решается замкнутая система уравнений относительно значения термосопротивления и степени черноты внешних и внутренних поверхностей.

Конструктивные особенности каждого типа участков ограждающих конструкций (толщины и теплопроводности слоев, тепловые мостики, трещины и т. д.), влияющие на кондукционный теплоперенос, учитываются исходя из прямых измерений для конкретного здания. Также из прямых измерений определяются реальные значения степени черноты ограждающих конструкций. Это позволяет уже в модели первого приближения избавиться от описанных выше упрощений, которые могут приводить к многократным ошибкам при расчете мощности теплового потока.

При вычислениях компоненты коэффициента теплоотдачи, связанной с конвекцией, для вертикальных ограждающих конструкций применяются критериальные зависимости [1] для турбулентного движения воздуха у нагретой или холодной вертикальной стенки и продольно обдуваемой пластины. Они получены с помощью теории подобия на основе уравнений газодинамики с учетом турбулентного переноса массы, импульса и энергии:

$$\partial\rho/\partial t + \mathrm{div}(\rho v) = 0;$$

$$\rho\,\mathrm{d}v/\mathrm{d}t = \rho g - \mathrm{grad}\,p + \rho(\nu_t + \nu)\Delta v +$$

$$+ [\zeta + \rho(\nu_t + \nu)/3]\mathrm{grad}(\mathrm{div}\,v);$$

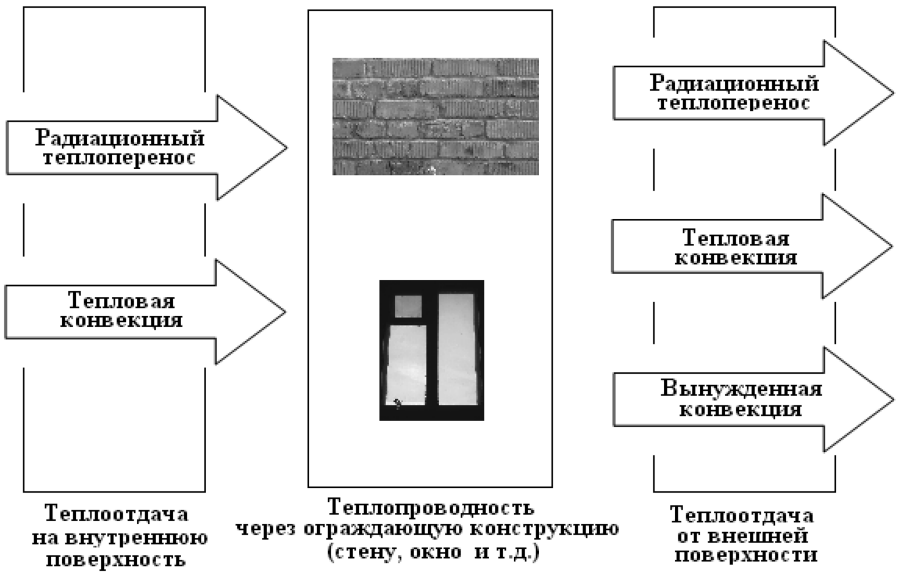


Рис. 1 Схема учета термосопротивлений тепловым потокам изнутри здания в окружающую среду на различных типах участков ограждающих конструкций (стен окон, дверей и др.)

$$\rho d\varepsilon/dt = -p\,\mathrm{div}\,v + \mathrm{div}[(\lambda + \lambda_t)\mathrm{grad}\,T] +$$

$$+ \eta\{2[(\partial v_x/\partial x)^2 + (\partial v_y/\partial y)^2 + (\partial v_z/\partial z)^2] +$$

$$+ [(\partial v_x/\partial y + \partial v_y/\partial x)^2 + (\partial v_x/\partial z + \partial v_z/\partial x)^2 +$$

$$+ (\partial v_z/\partial y + \partial v_y/\partial z)^2] - 2/3(\mathrm{div}\,v)^2\} + \zeta(\mathrm{div}\,v)^2,$$

где ρ — плотность; v — скорость; ε — внутренняя энергия; T — температура; g — гравитационное ускорение; p — давление; ν и ν_t — молекулярная и турбулентная вязкость; λ и λ_t — молекулярная и турбулентная теплопроводность; η — коэффициент внутреннего трения; ζ — вторая вязкость; Δ — оператор Лапласа; div — дивергенция; grad — градиент; x, y, z — декартовы координаты.

Из теории размерности и подобия к этой системе уравнений следуют строго обоснованные степенные зависимости между физическими величинами, объединенными в безразмерные критерии подобия (Нуссельта, Рэлея, Прандтля, Грасгофа, Рейнольдса и др.). Эти зависимости имеют фундаментальный характер и выполняются с точностью, превышающей обычную экспериментальную точность. Коэффициенты в этих зависимостях полуэмпирические, полученные в результате многочисленных тщательно проведенных экспериментов, выполненных при различных условиях и в различных лабораториях разных стран.

Для описания теплоотдачи при излучении применяется имеющая фундаментальный характер формула Планка для абсолютно черного тела. Влияние состояния излучающей поверхности учитывается тем, что в расчете участвуют реальные степени черноты. Таким образом, мощность потока энергии от теплообмена свободной тепловой конвекцией в поле сил тяжести определяется выражением

$$q = \alpha(T_1 - T_2);\ \ \alpha = \mathrm{Nu}\lambda/x,$$

где x — характерный вертикальный размер; λ — теплопроводность газа, определяемая в зависимости от средней арифметической температуры воздуха и стенки; Nu — критерий Нуссельта, $\mathrm{Nu} = 0{,}13\mathrm{Ra}^{(1/3)}$; $\mathrm{Ra} = \mathrm{PrCr}$ — критерий Рэлея, Pr — критерий Прандтля, для воздуха $\mathrm{Pr} = 0{,}73$, Gr — критерий Грасгофа, $\mathrm{G}r = g\Delta\rho x^3/\nu^2$, $\Delta\rho = \rho(T_1 - T_2)/273$, g — ускорение свободного падения ($g = 9{,}81$ м/с^2).

Мощность потока энергии от теплообмена вынужденной конвекцией со скоростью v для плоской стенки определяется выражением

$$q = \alpha(T_1 - T_2);\ \ \alpha = \mathrm{Nu}\lambda/x,$$

где λ — теплопроводность газа, определяемая в зависимости от средней арифметической температуры воздуха и стенки; критерий Нуссельта $\mathrm{Nu} = 0{,}0288\mathrm{Pr}^{0,4}\mathrm{Re}^{0,8}$; критерий Рейнольдса $\mathrm{Re} = vx/\nu$, ν — молекулярная вязкость газа, определяемая в зависимости от средней арифметической температуры воздуха и стенки.

Для теплообмена излучением применяются закономерности локально термодинамически равновесного теплового излучения серых тел, т. е. выражение для мощности потока энергии от теплообмена излучением между поверхностями с температурами T_1 и T_2 имеет вид

$$q = \beta\sigma(T_1^4 - T_2^4),$$

где β — степень черноты; σ — постоянная Стефана—Больцмана.

Приведенные выше формулы подставляются в выражения для суммирования последовательно-параллельных тепловых потоков: в каждой точке S ограждающей конструкции суммарный поток $q = q(S)$ находится как

$$q(S) = q_{in}^r(S) + q_{in}^T(S);$$

$$q_{in}^r(S) + q_{in}^T(S) = q^{\mathrm{c}}(S);$$

$$q^{\mathrm{c}}(S) = q_{ex}^r(S) + q_{ex}^v(S) + q_{in}^T(S),$$

где индексами "*in*" и "*ex*" отмечены тепловые потоки в данной точке на внутренней и внешней поверхностях ограждающей конструкции; q^{c} — поток теплопроводности через саму ограждающую конструкцию; верхние индексы "*r*", "*v*" и "*T*" соответствуют радиационному теплообмену и вынужденной и тепловой конвекции.

Для теплообмена излучением применяются закономерности локально термодинамически равновесного теплового излучения серых тел, т. е. мощность потока энергии от теплообмена излучением между поверхностями с температурами T_1 и T_i определяется выражением

$$q_{ex}^r = \beta_1\sigma T_1^4 - \sum_i \beta_i\Omega_i T_i^4,$$

где первый член в правой части формулы описывает тепловое излучение, испускаемое поверхностью самой ограждающей конструкции, а второй член — потоки излучения, падающие на эту поверхность. Они испускаются окружающими объектами: Землей, облаками и открытыми участками неба, стенами других домов и др.; Ω_i — телесные углы, под которыми видны эти объекты; β_i и T_i — их степени черноты и температуры. Если нет более точных данных, то в расчетах можно принять $\beta_i = 1$, для Земли и облаков — температуру воздуха, для стен других домов — T_1, а для открытых участков неба — достаточно низкую температуру, например, −50 °C. Для расчета телесных уг-

лов задаются высота соседних домов, расстояние до них, процент облачности и т. п. При расчете теплообмена на внутренних поверхностях температуру внутренних стен можно считать равной температуре воздуха в помещении.

Суммарные теплопотери через вертикальные ограждающие конструкции с учетом теплопроводности и теплоотдачи на всех поверхностях определяются интегралом по всей площади вертикальных ограждающих конструкций

$$Q_{tr}^{v} = \int q(S)dS.$$

При вычислениях безразмерных критериев подобия используются характеристики воздуха с учетом их зависимостей от температуры и давления по справочным данным [2]. Получающаяся система уравнений решается численными методами.

Методика второй части использует полученные теплофизические свойства здания для решения прямой задачи определения динамики остывания при заданных погодных условиях. Система трансцендентных и дифференциальных уравнений решается численно. Процессы теплопереноса описываются в тех же приближениях, что и в методике первой части.

Кроме методик расчетов разработана и занесена в информационно-справочную часть программного комплекса методика проведения измерений, в которой определяется порядок действий при измерении характеристик, аппаратное обеспечение измерений, описаны различные требования, предъявляемые к погрешности измерений, к операторам, которые будут проводить испытания, к оформлению результатов.

Разработанный программный комплекс содержит несколько форм. При запуске программы открывается основная форма (рис. 2), в которой пользователь может выполнить следующие действия:

- создать новый документ;
- открыть ранее созданный документ;
- указать тип конструкции строения;
- внести основные данные инспектируемого объекта, такие как адрес, дата обследования, данные основных ограждающих конструкций, информация из представленной заказчиком документации и другая информация;
- внести результаты измерений температур и яркостных температур ограждающих конструкций и их участков со степенью черноты, равной единице, характеристик окружения и др.;
- вызвать окно редактора модели ограждающих конструкций строения;
- вызвать диалоговую форму для задания параметров расчетов теплопотерь и остывания выбранного строения, а также анализа результатов выполненных расчетов;
- создать отчет заданного образца.

Форма "Параметры ограждающих конструкций" (рис. 3) предназначена для задания основных геометрических и физических характеристик строения, требуемых для расчетных модулей: габаритных размеров строения, а также состава и размещения различных типов элементов ограждающих конструкций на вертикальных поверхностях строения, данные о материалах и т.д.

В этой форме реализована возможность изменения модели исследуемого строения: добавления или удаления секций и зон вертикальных ограждающих конструкций. Геометрическая модель ограждающих конструкций строения состоит из нескольких примыкающих друг к другу секций разной высоты. Редактор позволяет задать состав типов элементов ограждающих конструкций, используемых в обследуемом строении для каждой из выбранных секций. Определенному типу ограждающей конструкции может соответствовать фрагмент стены, окно, дверь, витраж. Предусмотрен вариант, при котором в одной и той же секции существуют стены или окна с разными материалами.

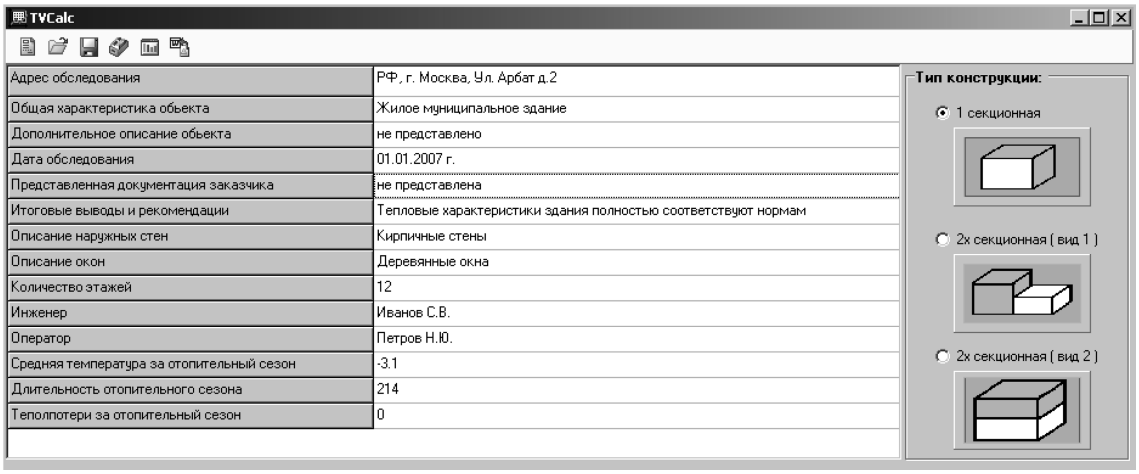


Рис. 2. Основная форма программы

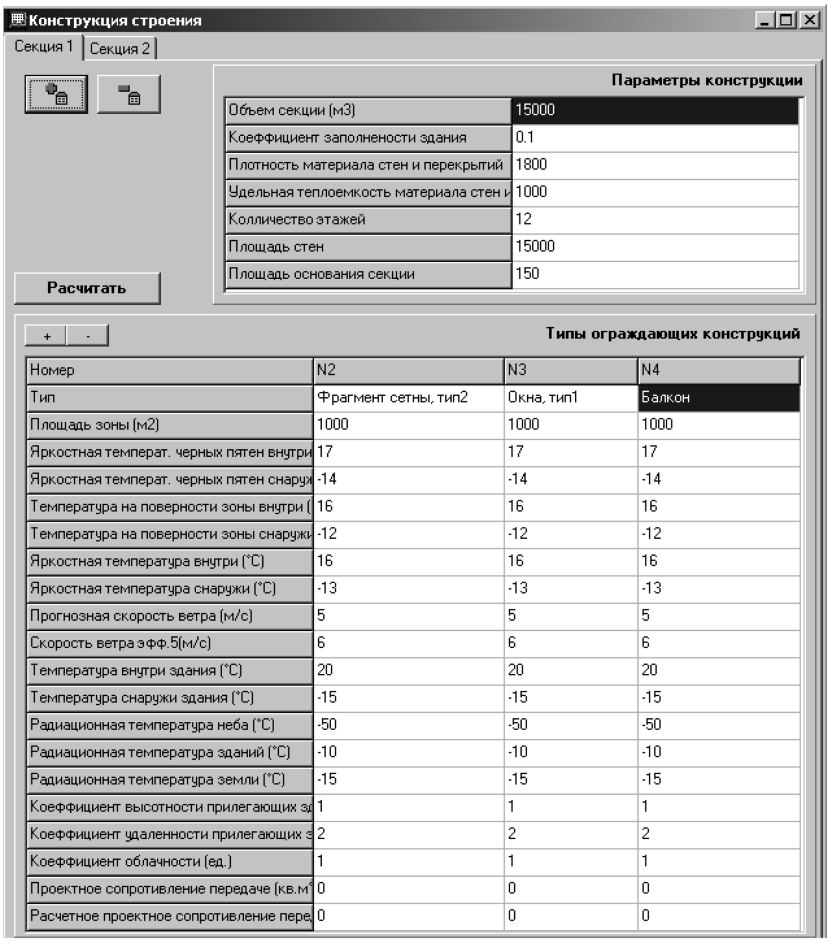


Рис. 3. Форма "Параметры ограждающих конструкций"

В состав параметров типа ограждающей конструкции входят геометрические размеры — ширина и высота. Геометрические размеры типа элемента предназначены для определения общей площади элементов данного типа на соответствующей стене строения. Для внесения этих данных в программу необходимо выбрать нужную секцию строения, а затем зону, указав тип элемента ограждающей конструкции, и ввести данные измерения в соответствующее поле. При нажатии кнопки "Рассчитать" программа проводит вычисления. Результаты расчета появляются в нижней строке таблицы в этом же диалоговом окне (рис. 3).

После задания параметров ограждающих конструкций и результатов натурных измерений можно перейти к расчету теплопотерь обследуемого строения в течение отопительного периода, а также к анализу процессов остывания строения при различных погодных условиях. Диалоговое окно управления расчетами теплопотерь и остывания строения позволяет задать основные параметры расчетов и проанализировать полученные результаты в интерактивном режиме.

Для запуска расчета остывания необходимо нажать кнопку "Рассчитать", которая расположена над таблицей параметров расчета остывания. Результаты расчетов остывания выводятся в виде девяти графиков временных зависимостей температуры для различных погодных условий (рис. 4, см. третью сторону обложки). Погодные условия для расчета задаются в виде наибольшего и наименьшего значений для наружных температур и прогнозных скоростей ветра. Третьи значения температуры и скорости ветра, применяемые для построения графиков, равны среднему арифметическому заданных экстремальных значений.

При нажатии пиктограммы "Создание отчета" в главной форме программы все экспериментальные и расчетные данные, а также графики остывания заносятся в автоматически сгенерированный программным комплексом отчет заданной формы, который после распечатки, утверждения подписями должностных лиц и заверения печатями уполномоченных организаций может заноситься непосредственно в энергетический паспорт здания в виде вкладыша.

Выводы

Применение методов и средств информатики и прикладной математики позволило отойти от упрощенных подходов СНиП 23-02—2003, СП 23-101—2000 и др. при определении теплопередачи через вертикальные ограждающие конструкции здания. Разработанная методика, алгоритмы и программы позволили описать процессы теплопереноса на основе теории газодинамического подобия и теории радиационного переноса энергии, что обеспечивает большую надежность результатов.

Разработан программный комплекс, позволяющий по ряду измерений автоматически определять теплофизические характеристики существующих ограждающих конструкций и прогнозировать как энергозатраты на отопление, так и время остывания при аварийном отключении теплоснабжения с учетом реальных погодных условий. Созданный программный комплекс имеет дружественный интерфейс, снабжен справочной системой информационной поддержки операто-

ра и системой автоматической генерации итогового отчета, готового к утверждению. Он прошел этапы опытной эксплуатации, уточнения методик и доработки интерфейса, проведено обучение операторов. Комплекс применяется уполномоченной организацией г. Москвы при официальном обследовании зданий и комплектации энергетических паспортов. Возможно его применение другими организациями и предприятиями, в том числе в других регионах РФ и за рубежом.

Список литературы

1. **Кутателадзе С. С.** Теплопередача и гидродинамическое сопротивление. М.: Энергоатомиздат, 1990. 366 с.
2. **Григорьев И. С., Мейлихов Е. З.** Физические величины: Справочник. М.: Энергоатомиздат, 1991. 1232 с.
3. **Wind** Effects on Buildings and Structures // Proc. Symposium. London: Her Majesty's Stationery Office. 1965. N 16.
4. **Излучательные** свойства твердых материалов: Справочник / Под ред. А. Е. Шейндлина. М.: Энергия, 1974.
5. **Новицкий Л. А., Степанов Б. М.** Оптические свойства материалов при низких температурах: Справочник. М.: Машиностроение, 1980.
6. **Kaspar J.** Radiometry. NJ etc.: McGraw-Hill, 1972.

УДК 004.942

С. В. Шалагин, канд. тех. наук, докторант,
Ф. Х. Кайбушев, аспирант,
Казанский государственный технический университет им. А. Н. Туполева

Реализация схем умножения элементов поля Галуа в базисе ПЛИС класса FPGA

Предложена модель схемы умножения элементов поля Галуа вида $GF(2^n)$, которая позволяет уменьшить оценки емкостной и временной сложности для этой схемы за счет параллельного вычисления разрядов произведения. Построены функциональные модели схемы умножения в базисе ПЛИС/FPGA семейства Stratix фирмы Altera при использовании языка описания аппаратуры VHDL. Результаты экспериментального исследования позволяют сделать вывод об эффективной реализации предложенной модели.

Ключевые слова: *схема умножения, сложность, поле Галуа, параллельное вычисление, ПЛИС.*

Введение

Операция умножения элементов $GF(2^n)$ в базисе программируемых логических интегральных схем (ПЛИС) класса FPGA [1, 2] является базовой для синтеза систем обработки информации, в частности, при вычислении цифровой свертки или дискретного преобразования Фурье [3—5].

Цель работы — исследование реализации схемы умножения (СУ) вида $GF(2^n)$ в базисе ПЛИС/FPGA на основе предложенной модели, предполагающей параллельное вычисление разрядов произведения и ориентированной на VHDL-реализацию.

Решена задача синтеза модели СУ элементов поля Галуа вида $GF(2^n)$, ориентированная на базис ПЛИС класса FPGA при использовании языка VHDL. Проведено сопоставление предложенной модели с известной моделью СУ элементов $GF(2^n)$ [6, 7] в плане адекватности [1, 6] базису ПЛИС/FPGA семейства *Stratix* фирмы *Altera* (при использовании специализированной системы автоматизации процесса проектирования [8] Quartus II vesion 4.2 (далее САПР)).

Теоретический анализ

Операция умножения элементов поля Галуа.

Модель 1. Для схемы умножения элементов $GF(2^n)$ — СУ/$GF(2^n)$, предложенной в работе [7], теоретические оценки емкостной сложности равны

$$Q_{\text{TEOP}} = n^2 + n(n-1) + f(n), \qquad (1)$$

где $f(n) = \sum_{i=0}^{2n-1} (S_i - 1)$, S_i — число единичных элементов в i-й строке матрицы $\hat{\mathbf{D}}$, которую получим из $\mathbf{D} = (\mathbf{I}, \mathbf{A}^1, ..., \mathbf{A}^{n-1})^T$ путем вычеркивания повторных строк [6, 7], $\mathbf{A}$ — сопровождающая матрица для образующего многочлена поля $GF(2^n)$, $\mathbf{I}$ — единичная матрица; n — порядок $GF(2^n)$. Теоретические оценки временной сложности выполнения операции умножения элементов поля Галуа СУ будут равны [6, 7]

$$T_{\text{TEOP}} =]\log_2(S_{\max})[+1 +]\log_2 n[, \qquad (2)$$

где $S_{\max} = \max_i S_i$.

Модель 2. Преимущество данной модели СУ/$GF(2^n)$ заключается в том, что значения разрядов произведения элементов поля Галуа вычисляются параллельно и независимо друг от друга. Это позволяет снизить оценки временной сложности для модели 2 по сравнению с моделью 1.

Утверждение. Для описания операции умножения элементов $GF(2^n)$ — $\alpha = (\alpha_0\ \alpha_1\ ...\ \alpha_{n-1})^T$,

$\beta = (\beta_0\ \beta_1\ \ldots\ \beta_{n-1})^T$ в целях получения произведения $c = (c_0\ c_1\ \ldots\ c_{n-1})^T$, $\alpha_i, \beta_i, c_i \in GF(2)$, $i = \overline{0, n-1}$, применима следующая формула [9]:

$$c_i = \begin{pmatrix} \alpha_0 \\ \ldots \\ \alpha_i \end{pmatrix}^T \begin{pmatrix} \beta_i \\ \ldots \\ \beta_0 \end{pmatrix} + \sum_{j=1}^{n-1} \left(\begin{pmatrix} \alpha_j \\ \ldots \\ \alpha_{n-1} \end{pmatrix}^T \begin{pmatrix} \beta_{n-1} \\ \ldots \\ \beta_j \end{pmatrix} \xi_i^{j+n-1} \right), \qquad (3)$$

где ξ^m, $m = \overline{n, (2n-2)}$ — степени примитивного элемента $\xi \in GF(2^n)$, $P(\xi) = 0$, т. е. постоянные величины. Обозначим $\xi^m = (\xi_0^m\ \ \xi_1^m\ \ \ldots\ \ \xi_{n-1}^m)^T$, $\xi_i^m \in GF(2)$, $i = \overline{0, n-1}$.

Определим оценки временной и емкостной сложности для модели 2 СУ/$GF(2^n)$, представленной на абстрактном уровне в виде (3) — $T_{\text{ТЕОР}}$ и $Q_{\text{ТЕОР}}$. Оценки временной сложности для СУ/$GF(2^n)$, реализованной согласно (3), имеют вид

$$T_{\text{ТЕОР}} = 1 + \max_{i=\overline{0,n-1}} \left(\max_{j=\overline{1,n-1}} ([\log_2(i + 1 + (n-j)\xi_i^{j+n-1})])\right), \qquad (4)$$

где ξ_i^{j+n-1} определены согласно (3). Теоретической оценкой временной сложности данной СУ является максимальная оценка вычисления c_i, где $i = \overline{0, n-1}$ согласно (3). Оценки емкостной сложности вычисляются исходя из числа элементарных схем — двухвходовых логических элементов, выполняющих произвольную булеву функцию. Данные оценки для СУ/$GF(2^n)$ определены по формуле

$$Q_{\text{ТЕОР}} = 2n(n-1) + 1 + \sum_{i=0}^{n-1} \sum_{j=n}^{2n-1} \xi_i^j, \qquad (5)$$

где ξ_i^j определены согласно (3).

На основе изложенного выше сформулирована следующая теорема.

Теорема. Оценки временной и емкостной сложности, вычисленные на основе элементарных схем, которые задействованы для реализации СУ/$GF(2^n)$, заданной согласно (3), определяются по формулам (4) и (5) соответственно.

Доказательство теоремы вытекает непосредственно из формулы (3). Число операций двоичного умножения (конъюнкций) равно $Q_\otimes = \frac{1}{2}n(n+1) + \sum_{j=n}^{n-1}((n-j))$ и операций поразрядного сложения по модулю 2 — $Q_\oplus = \frac{1}{2}n(n-1) + \sum_{j=n}^{n-1}\left((n-1-j)\sum_{j=0}^{n-1}\xi_i^{j+n-1}\right)$.

Базис ПЛИС. Рассмотрим ПЛИС, которая используется для реализации СУ. Семейство Stratix — программируемое логическое устройство семейства PLD, класса FPGA [2]. Устройства Stratix выполнены на основе 0,13-микрометровой технологии. Логический элемент содержит четырехвходовую таблицу, которая является генератором булевых функций от четырех переменных. Кроме того, каждый логический элемент содержит программируемый регистр и цепочку переноса со способностью выбора переноса [2]. Оценки реальных затрат логических ресурсов при реализации функциональный модели проектируемой схемы на ПЛИС Stratix определяются по формуле вида [1]

$$Q = 3\sum_{i=1}^{4} N_{\Gamma\Phi(i)}, \qquad (6)$$

где $N_{\Gamma\Phi(i)}$ — общее число генераторов, реализующих функции на i булевых переменных, задействованных для реализации функциональной модели проектируемой схемы.

Методика реализации цифровых схем в базисе ПЛИС/FPGA

Методика реализации цифровых схем включает два этапа.

1. Расчет теоретических оценок для СУ/$GF(2^n)$ и создание VHDL-описания проектируемой схемы. САПР позволяет реализовать СУ/$GF(2^n)$, определенную согласно (3), на основе VHDL-описания на структурном, потоковом и поведенческом уровнях описание цифровых схем ввиду имеющейся теоретической проработки указанной схемы (см. теорему).

2. Оценка адекватности проектируемой схемы базису ПЛИС/FPGA. При реализации на ПЛИС широкого класса устройств обработки цифровых данных (регистров, счетчиков, мультиплексоров и т. п.). доля ресурсов взаимосвязи в общих затратах логических ресурсов ПЛИС составляет 20 — 30 % относительно теоретических оценок емкостной сложности функциональных моделей комбинационной схемы [10]. Введем в рассмотрение крите-

рий, характеризующий долю ресурсов взаимосвязи [6, 1],

$$K_{PT} = \frac{Q - Q_{\text{TEOP}}}{Q} 100\ \%, \qquad (7)$$

где Q — оценка реальных затрат логических ресурсов при реализации СУ/$GF(2^n)$ в базисе ПЛИС, которая вычислена согласно (6); Q_{TEOP} — оценка емкостной сложности СУ/$GF(2^n)$ (теоретическая), полученная согласно (1) для модели 1 и согласно (5) для модели 2.

Функциональная модель комбинационной схемы, реализованная в базисе ПЛИС, является адекватной базису ПЛИС по критерию вида (7), если его значение не превышает 30 % [1].

Для ПЛИС сопротивление проводников межсоединений (МС) растет по мере увеличения степени интеграции. Так, в случае 0,5-микрометровой технологии доля задержки МС в общей задержке проекта составляет 60... 70 %, а при 0,25-микрометровой технологии — 80...90 % [8]. Возникает задача снижения времени задержки функционирования проекта за счет выбора соответствующей конфигурации МС, определяющей порядок прохождения сигналов между логическими элементами. Для ее решения введем критерий, характеризующий вклад МС в оценку временной сложности функциональной модели цифровой схемы (ФМЦС) [1, 6],

$$K_t = \frac{T_{MC}}{T} 100\ \%, \qquad (8)$$

где T_{MC} — оценка времени задержки МС, задействованных для реализации СУ/$GF(2^n)$ на ПЛИС; T — общая оценка времени задержки функционирования при реализации СУ/$GF(2^n)$ на ПЛИС.

ФМЦС, реализованная в базисе ПЛИС, является адекватной базису ПЛИС по критерию вида (8), если его значение не превышает 90 % для ПЛИС, имеющей степень интеграции более, чем 0,25 мкм на один элементарный вентиль [1].

Эффективность задействования логических ресурсов ПЛИС при реализации СУ/$GF(2^n)$ оценивается по времени задержки прохождения сигнала внутри них в пересчете на один уровень элементарной схемы; число уровней определено согласно (3):

$$t_{\text{зЭС}} = \frac{T - T_{\text{MC}}}{T_{\text{TEOP}}}, \qquad (9)$$

где T — время задержки функционирования СУ/$GF(2^n)$; T_{MC} — вклад задержки МС в общее время T; T_{TEOP} — теоретическая оценка временной сложности СУ/$GF(2^n)$, вычисленная согласно (2) и (4) для моделей 1 и 2 соответственно. Чем ниже значение критерия (9), тем выше эффективность задействования логических ресурсов ПЛИС.

Решение задачи оценки адекватности реализации СУ/$GF(2^n)$ в базисе ПЛИС сводится к вычислению критериев (7)—(9) [1], а задача сравнительной оценки — к сопоставлению указанных критериев для моделей 1 и 2 СУ/$GF(2^n)$, реализованных на ПЛИС.

Пример. Построим СУ/$GF(2^5)$ для модели 1. Пусть $P(x) = x^5 + x^2 + 1$ примитивный многочлен. Сопровождающая матрица [11] имеет следующий вид:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Вычислим степень A^2, A^3, A^4 и построим матрицу $\mathbf{D} = (I, A, A^2, A^2, A^3, A^4)^T$. Вычеркнем из матрицы $\mathbf{D}$ все повторные строки и получим матрицу $\hat{\mathbf{D}}$. Тогда СУ имеет вид:

$$\alpha\beta = \alpha_0(\beta_0\ \beta_1\ \beta_2\ \beta_3\ \beta_4) + \\ + \alpha_1(\beta_4\ \beta_0\ (\beta_1 + \beta_4)\ \beta_2\ \beta_3) + \\ + \alpha_2(\beta_3\ \beta_4\ (\beta_0 + \beta_3)(\beta_1 + \beta_4)\ \beta_2) + \\ + \alpha_3(\beta_2\ \beta_3\ (\beta_2 + \beta_4)\ (\beta_0 + \beta_3)\ (\beta_1 + \beta_4)) + \\ + \alpha_4((\beta_1 + \beta_4)\ \beta_2\ (\beta_1 + \beta_3 + \beta_4)\ (\beta_2 + \beta_4)\ (\beta_0 + \beta_3)).$$

Согласно (1) и (2) получим $Q_{\text{TEOP}} = 50$, $T_{\text{TEOP}} = 6$.

Построим СУ/$GF(2^5)$ для модели 2. Пусть $P(x) = x^5 + x^2 + 1$ примитивный многочлен. Согласно выражению (3) СУ имеет следующий вид:

$$c_0 = \alpha_0\beta_0 + (\alpha_1\beta_4 + \alpha_2\beta_3 + \alpha_3\beta_2 + \alpha_4\beta_1) + \alpha_4\beta_4;$$

$$c_1 = \alpha_0\beta_1 + \alpha_1\beta_0 + (\alpha_2\beta_4 + \alpha_3\beta_3 + \alpha_4\beta_2);$$

$$c_2 = \alpha_0\beta_2 + \alpha_1\beta_1 + \alpha_2\beta_0 + (\alpha_1\beta_4 + \alpha_2\beta_3 + \\ + \alpha_3\beta_2 + \alpha_4\beta_1) + (\alpha_3\beta_4 + \alpha_4\beta_3) + \alpha_4\beta_4;$$

$$c_3 = \alpha_0\beta_3 + \alpha_1\beta_2 + \alpha_2\beta_1 + \alpha_3\beta_0 + \\ + (\alpha_2\beta_4 + \alpha_3\beta_3 + \alpha_4\beta_2) + \alpha_4\beta_4;$$

$$c_4 = \alpha_0\beta_4 + \alpha_1\beta_3 + \alpha_2\beta_2 + \alpha_3\beta_1 + \alpha_4\beta_0 + \\ + (\alpha_3\beta_4 + \alpha_4\beta_3);$$

$$\xi^5 = (1\ \ 0\ \ 1\ \ 0\ \ 0);$$
$$\xi^6 = (0\ \ 1\ \ 0\ \ 1\ \ 0);$$
$$\xi^7 = (0\ \ 0\ \ 1\ \ 0\ \ 1);$$
$$\xi^8 = (1\ \ 0\ \ 1\ \ 1\ \ 0).$$

Тогда согласно выражениям (4) и (5) $T_{\text{TEOP}} = 5$, $Q_{\text{TEOP}} = 50$.

Экспериментальное исследование

Для оценки адекватности схемы умножения базису ПЛИС архитектуры FPGA по предложенной методике необходимо получить заданный набор данных. Для выполнения этапа 1 требуется рассчитать теоретические оценки емкостной сложности $Q_{\text{ТЕОР}}$. Для модели 1 СУ значения данной оценки рассчитаны согласно формуле (1), а для модели 2 СУ — согласно формуле (5). Расчет оценок $T_{\text{ТЕОР}}$ для модели 1 и модели 2 проводится согласно (2) и (4) соответственно. Для реализации этапа 2 проводится расчет оценок реальных затрат согласно формуле (6). Критерии K_{PT}, K_t и $t_{\text{з.ЭС}}$ для каждой из реализаций СУ вычислены согласно (7), (8) и (9) соответственно.

Для исследования адекватности реализации СУ/$GF(2^n)$ в базисе ПЛИС строим функциональную модель комбинационной схемы посредством САПР. Результатом работы САПР является протокол, который содержит оценки $N_{\text{ГФ}(i)}$, T и $T_{\text{МС}}$.

Таблица 1

Сопоставление теоретических оценок емкостной и временной сложности

n	$Q_{\text{ТЕОР}}$		$T_{\text{ТЕОР}}$	
	М1	М2	М1	М2
8	164	141	6	6
16	599	552	8	7
32	2 323	2 183	10	9
64	8 514	8 324	10	9

Таблица 2

Сопоставление теоретических оценок емкостной сложности и оценок реальных затрат

n	$N_{\text{ГФ (4)}}$		$N_{\text{ГФ (3)}}$		$N_{\text{ГФ (2)}}$		Q		$K_{\text{РТ}}$, %	
	М1	М2	М1	М2	М1	М2	М1	М2	М1	М2
8	47	42	8	5	4	4	177	153	7,34	7,84
16	179	170	24	13	17	9	660	576	9,24	4,17
32	706	682	54	30	28	23	2 364	2 205	1,73	1,00
64	2 777	2 726	40	53	93	28	8 730	8 421	2,47	1,15

При реализации функциональных моделей М1 и М2 $N_{\text{ГФ (1)}}$ не задействованы.

Таблица 3

Сопоставление вклада межсоединений в общую задержку функционирования

n	K_t, %		T_{MC}, нс		T, нс		$t_{\text{з.ЭС}}$	
	М1	М2	М1	М2	М1	М2	М1	М2
8	62,48	68,00	7,552	9,795	12,088	14,405	0,756	0,768
16	66,43	68,20	9,293	10,994	13,99	16,12	0,587	0,732
32	70,84	75,23	13,167	14,786	18,588	19,655	0,542	0,541
64	78,15	76,43	16,065	15,755	20,557	20,613	0,449	0,540

Оценки временной сложности T и $T_{\text{МС}}$ вычислены в наносекундах.

Значения теоретических оценок емкостной и временной сложности для моделей 1 и 2 СУ/$GF(2^n)$ (оценки реальных затрат логических ресурсов, время задержки функционирования, а также критерии (7)—(9)) приведены в табл. 1—3 для $n = 8$; 16; 32; 64. В табл. 1 сравниваются теоретические оценки сложности моделей 1 и 2. В табл. 2 приведены данные для функциональных моделей 1 (М1) и 2 (М2) СУ, реализованных в базисе ПЛИС класса FPGA серии EP1S10B672C6 семейства *Stratix* фирмы *Altera*.

Обсуждение результатов

Анализируя результаты, представленные в табл. 1, наблюдаем, что теоретические оценки емкостной сложности для модели 2 СУ меньше, чем для СУ, предложенной в работе [7] (модели 1). Теоретические оценки временной сложности для модели 2 СУ элементов поля Галуа также будут меньше, чем для модели 1. Что касается поля $GF(2^n)$, $n = 8$, наблюдаются одинаковые показатели теоретических оценок временной сложности.

При сопоставлении коэффициента, характеризующего вклад МС в общую задержку проекта, наблюдается следующее: модель 1 имеет большую степень адекватности по критерию вида (8), так как для нее при $n = 8$, 16, 32 она имеет меньшие значения, чем для модели 2. При $n = 64$ наблюдается обратное — для модели 2 значение критерия (8) будет иметь меньшее значение, по сравнению с моделью 1. Наблюдается уменьшение доли ресурсов МС, что объясняется наличием на ПЛИС семейства *Stratix* гибких структур как логических ячеек, так и системы взаимосвязи МС.

На основе сопоставления оценок $K_{\text{РТ}}$, приведенных в табл. 3, отметим следующее. Гибкость логических ресурсов ПЛИС семейства *Stratix* проявляется для модели 2 следующим образом. Наблюдается снижение оценок емкостной сложности и критерия вида (7) по сравнению с моделью 1. Различия оценок емкостной сложности для ФМЦС СУ, реализованных на ПЛИС EP1S10B672C6, объясняется тем, что их логические ресурсы — логические элементы, задействованы не в полном объеме при реализации модели 1.

При сопоставлении адекватности реализации ФМЦС СУ на ПЛИС *Stratix* по критерию вида (9) наблюдается следующее: значения $t_{\text{з.ЭС}}$ для модели 1 несколько меньше, чем для модели 2 при $n = 8$ и 16, что объясняется меньшим вкладом задержки МС в общую задержку ФМЦС для модели 1, чем для модели 2. Для $n = 32$ и 64 наблюдается об-

ратная ситуация, что связано с увеличением доли задействованных логических элементов ПЛИС.

Таким образом, можно сделать вывод о большей адекватности реализации на ПЛИС модели 2 по сравнению с моделью 1. Это достигается за счет того, что модель 2 близка к однородной вычислительной структуре, элементы которой сходны со структурой межсоединений и логических элементов ПЛИС семейства *Stratix*. Причем с ростом порядка $GF(2^n)$, для СУ, реализованной на базе модели 2, доля ресурсов взаимосвязи (7) и вклад МС в общую задержку функционирования (8) для нее снижаются: критерий (7) — на 1,32 %, а критерий (8) — на 1,72 % (см. табл. 2 и 3).

Заключение

Моделирование специализированных систем обработки информации в базисе ПЛИС является актуальным научным направлением. В данной связи исследование цифровых схем в плане оценки адекватности базису ПЛИС класса FPGA является полезным для оптимизации их физической реализации. Предложенная в работе схема умножения в $GF(2^n)$ (модель 2) позволяет повысить эффективность использования ресурсов ПЛИС/ FPGA и автоматизировать процесс синтеза СУ при их конфигурировании.

Список литературы

1. **Шалагин С. В.** Экспериментальное исследование методики синтеза комбинационных схем на программируемых микросхемах класса FPGA // Микроэлектроника. 2004. Т. 33. № 6. С. 421—432.
2. **Altera.** Documentation library, 1995—2004. Altera Corporation.
3. **Сюрин В. Н., Иванов Н. Н., Альхимович В. В.** Реализация вычислений в конечных полях // Зарубежная электроника. 1990. № 5. С. 59—68.
4. **Orlando G., Paar C.** A High-Perfomance Reconfigurable Elliptie Curre Processor for $GF(2^n)$ // Cryptographie Hardware and Embedded Systems. Springer-Verland. Lecture Notes in Computer Science 1965. 2000. P. 41—56.
5. **Paar C., Fleischemann P., Soria-Rodrigues P.** Fast arithmetic for public-key algorithms in Galois fields with composite exponents // IEEE Trans. on Computers. Oct. 1999. V. 48. № 10. P. 1025—1034.
6. **Захаров В. М., Нурутдинов Ш. Р., Шалагин С. В.** Аппаратная реализация умножения элементов поля Галуа на программируемых микросхемах архитектуры FPGA // Вестник Казан. гос. техн. ун-та им. А. Н. Туполева. 2001. № 1. С. 36—41.
7. **Нурутдинов Ш. Р.** Основы теории полиномиальных моделей автоматных преобразований над полем Галуа. Казань: Изд-во КГУ, 2005. 156 с.
8. **Норенков И. П.** ECAD: автоматизация проектирования в электронике // Информационные технологии. Приложение. 2001. № 8. С. 4.
9. **Шалагин С. В.** Оценка сложности для операций умножения элементов поля Галуа // Материалы Всерос. научной конф. молодых ученых. В 7-ми частях. Новосибирск: Изд-во НГТУ. Ч. 1. 2006. С. 76—78.
10. **Пономарев В. И., Шабалин Л. А.** Проектирование реконфигурируемых устройств обработки цифровых потоков данных // Информационные технологии. 1996. № 5. С. 24—28.
11. **Лидл Р., Нидеррайтер Г.** Конечные поля. В 2 т. Т. 1: Пер. с англ. М.: Мир, 1988. 430 с.

УДК 519.673

А. С. Горобцов, д-р техн. наук, зав. каф.,
В. В. Гетманский, магистрант,
М. В. Резников, магистрант,
Волгоградский государственный технический университет

Параллельное решение систем дифференциально-алгебраических уравнений большой размерности

Рассматриваются алгоритмы параллельных вычислений в методах решения задач динамики многотельных систем. Обосновывается применимость распараллеливания для решения больших многотельных моделей. Приведен пример расчетной схемы большой размерности, решаемой с помощью разработанного метода.

Ключевые слова: *задачи большой размерности, многотельные системы, динамика систем твердых тел, CAE, дифференциально-алгебраические уравнения, разделение модели, синхронные параллельные вычисления, MIMD.*

Системы дифференциально-алгебраических уравнений широко используются в программах моделирования динамики систем многих тел. Такие уравнения имеют вид

$$\begin{cases} \mathbf{M}\ddot{\boldsymbol{x}} - \mathbf{D}^{\mathrm{T}}\boldsymbol{p} = \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, t); \\ \mathbf{D}\ddot{\boldsymbol{x}} = \boldsymbol{h}(\dot{\boldsymbol{x}}, \boldsymbol{x}). \end{cases} \quad (1)$$

Здесь $\boldsymbol{x}$ — вектор переменных системы размерностью n; $\mathbf{M}$ — постоянная матрица коэффициентов при вторых производных; $\boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, t)$ —вектор-функция правых частей дифференциальных уравнений; $\mathbf{D}$ — матрица переменных коэффициентов уравнений связи размерностью $k \times n$ (k — число связей); $\boldsymbol{h}(\dot{\boldsymbol{x}}, \boldsymbol{x})$ — вектор правых частей уравнений связи; $\boldsymbol{p}$ — вектор множителей Лагранжа.

Отметим, что в форме (1) могут записываться также уравнения метода конечных элементов в случае наличия физической и геометрической нелинейностей.

При интегрировании (1) можно выделить две основные составляющие вычислительных затрат. Во-первых, это затраты, связанные с вычислени-

ем правых частей уравнений и коэффициентов матрицы **D**. Во-вторых, это затраты на триангуляцию матрицы

$$\begin{pmatrix} \mathbf{M} & -\mathbf{D}^{\mathrm{T}} \\ \mathbf{D} & 0 \end{pmatrix}. \qquad (2)$$

Триангуляцию матрицы (2) необходимо проводить на каждом шаге интегрирования. В случае, если матрица **D** является блочно диагональной, то триангуляцию можно проводить независимо для каждого блока. Методы приведения матрицы **D** к блочно-диагональному виду представлены в [1].

В настоящей статье рассматриваются методы разбиения (1) на слабосвязанные подсистемы, допускающие раздельное численное интегрирование с минимальным в некотором смысле обменом данных между процедурами интегрирования частей. В предположении, что матрица **D** имеет блочно диагональный вид, связи между подсистемами включены в первое уравнение системы (1), поэтому далее будем рассматривать только уравнение

$$\mathbf{M}\ddot{\boldsymbol{x}} = \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, t). \qquad (3)$$

В случае диагональной матрицы **M** связь между уравнениями (3) определяется только правой частью. Требование диагональности матрицы **M** не сильно снижает общность уравнений, так как, например, в динамике систем многих тел всегда существует система координат с диагональными инерционными коэффициентами.

Представим систему (3) в виде набора из l матричных уравнений:

$$\begin{aligned} &\mathbf{M}_1\ddot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\dot{\boldsymbol{x}}, \boldsymbol{x}_1, t); \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\mathbf{M}_l\ddot{\boldsymbol{x}}_l = \boldsymbol{f}_l(\dot{\boldsymbol{x}}_l, \boldsymbol{x}_l, t). \end{aligned} \qquad (4)$$

В системе (4) в общем случае невозможно выбрать такое разбиение, чтобы $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l)$, поскольку в этом случае система (4) является полностью независимой. Для связанных уравнений целесообразно сформулировать следующую задачу — найти такое разбиение вектора $\boldsymbol{x} = \boldsymbol{x}_1 \cup \ldots \cup \boldsymbol{x}_l$, из которого можно определить точные значения правых частей (1).

Для решения задачи представим каждый подвектор $\boldsymbol{x}_i$ виде двух частей $\boldsymbol{x}_i = (\boldsymbol{x}_{if}, \boldsymbol{x}_{ib})$, где $\boldsymbol{x}_{if}$ — вектор некоторых внутренних переменных; $\boldsymbol{x}_{ib}$ — вектор граничных переменных. Внутренние переменные $\boldsymbol{x}_{if}$ можно определить простым разделением полного вектора на непересекающиеся части $\boldsymbol{x}_{if} \cap \boldsymbol{x}_{jf} = 0$ при $i! = j$. Граничные переменные должны обеспечивать точное вычисление правых частей на каждом шаге интегрирования и иметь минимальную размерность. Для нахождения граничных переменных следует использовать матрицу инциденций правых частей (3). В такой матрице по вертикали откладываются номера переменных вектора $\boldsymbol{x}$, по горизонтали — номера переменных, входящих в правую часть соответствующего уравнения. Рассмотрим нахождение граничных переменных для матрицы инциденций ленточного типа в случае разбиения системы (1) на две части (рис. 1).

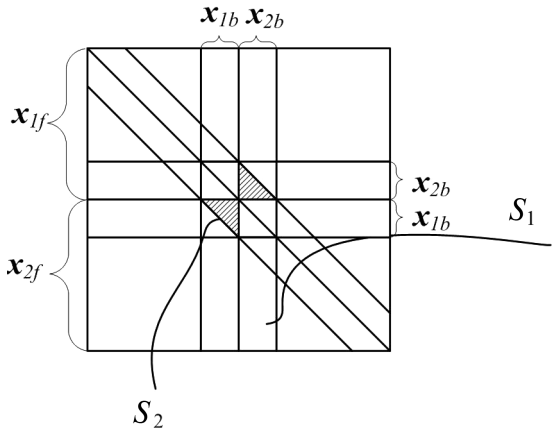


Рис. 1. Структура матрицы инциденции в случае разбиения системы на две части

В функции правых частей первой части входит область S_1, которая определяется переменными $\boldsymbol{x}_{1b}$ и которые необходимы для точного вычисления правых частей подсистемы (1). Аналогично в правые части второй подсистемы входит область S_2, которая в свою очередь определяется переменными $\boldsymbol{x}_{2b}$. Таким образом, если принять $\boldsymbol{x}_1 = (\boldsymbol{x}_{1f}, \boldsymbol{x}_{1b})$, а $\boldsymbol{x}_2 = (\boldsymbol{x}_{2f}, \boldsymbol{x}_{2b})$, то система (4) в этом случае будет точно определять значения старших производных, соответствующих вектору внутренних переменных каждой подсистемы. Однако для независимого численного интегрирования каждой подсистемы необходимо знать точные значения старших производных граничных переменных. Нетрудно заметить, что граничные переменные первой подсистемы являются внутренними переменными второй, и наоборот. Следовательно, при решении после вычисления старших производных каждого из матричных уравнений (4) нужно выполнять следующие замены:

$$\boldsymbol{x}_{1b} = \boldsymbol{x}_{2fz}; \quad \boldsymbol{x}_{2b} = \boldsymbol{x}_{1fz}, \qquad (5)$$

где $\boldsymbol{x}_{2fz}$ — подмножество внутренних переменных второго уравнения (4), соответствующих граничным переменным первого; аналогично $\boldsymbol{x}_{1fz}$.

Таким образом, интегрируя систему (4) с дополнительным обменом данных (5), можно разбить программу на два процесса, которые будут связаны только передачей данных по условию (5).

Для независимого интегрирования системы (1) в форме (4) можно привести ее к виду

$$\begin{cases} \mathbf{M}\ddot{\boldsymbol{x}} - \mathbf{D}_f^{\mathrm{T}}\boldsymbol{p}_f + \mathbf{D}_b^{\mathrm{T}}\alpha_{cb}\mathbf{D}_b\boldsymbol{x} = 0; \\ \mathbf{D}_f\ddot{\boldsymbol{x}} = \boldsymbol{h}_f(\dot{\boldsymbol{x}}, \boldsymbol{x}), \end{cases} \qquad (6)$$

где $\mathbf{D}_f = \begin{pmatrix} 0 & 0 & \mathbf{D}_2 \\ 0 & 0 & 0 \\ \mathbf{D}_1 & 0 & 0 \end{pmatrix}$ — блочно-диагональная матрица связей; $\boldsymbol{p}_f$ — вектор множителей Лагранжа, соответствующих связям с матрицей $\mathbf{D}_f$; $\boldsymbol{h}_f(\dot{\boldsymbol{x}}, \boldsymbol{x})$ — вектор правых частей уравнений связей разделенной системы; $\mathbf{D}_b$ — матрица граничных коэффициентов уравнений связи; α_{cb} — матрица консервативных коэффициентов для граничных связей. Подматрицы $\mathbf{D}_1$ и $\mathbf{D}_2$ выбираются согласно условиям, рассмотренным в [1], и, кроме того, коэффициенты этих матриц должны соответствовать внутренним переменным подсистем (4), которые не входят ни в какие граничные переменные других подсистем.

Для примера рассмотрим задачу расчета динамики цепной системы большой размерности. Цепь представлена набором тел, последовательно соединенных упругими элементами. К некоторым звеньям приложены нагрузки. Для рассматриваемой системы матрица инциденций будет иметь ленточный вид, т. е. допустимо деление системы на линейные участки с сохранением точности расчетов, как у полной модели. Для иллюстрации рассуждений рассмотрим цепь из четырех тел с одной степенью свободы каждого тела, представленную на рис. 2.

Для упрощения вида формул рассмотрим случай с единичными массами тел и коэффициентами жесткости пружин. Внешние нагрузки отсутствуют, тогда система уравнений в матричном виде для этой модели

$$\mathbf{M}\ddot{\boldsymbol{x}} + \mathbf{K}\boldsymbol{x} = 0, \qquad (7)$$

где

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ — матрица масс;}$$

$$\mathbf{K} = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} \text{ — матрица жесткости.}$$

Каждое отдельное уравнение для подсистемы будет иметь аналогичные матрицы жесткости, но с размерностью 3. На матрице

$$\mathbf{K} = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}. \qquad (8)$$

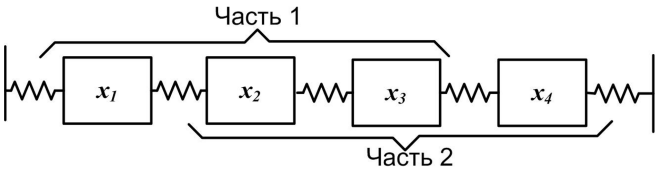


Рис. 2. Схема многозвенного представления упругой конструкции

выделена подматрица для первой части модели. Третья строка соответствует уравнению для нахождения $\ddot{\boldsymbol{x}}_3$, при разбиении теряется последний коэффициент, поэтому координаты третьего тела нельзя считать корректными. Для матрицы системы второй подмодели аналогично теряется первый коэффициент во второй строке, соответствующей $\ddot{\boldsymbol{x}}_2$. При этом уравнения для расчета координат третьего тела полностью корректны для второй подсистемы, а уравнение расчета координат второго тела — для первой подсистемы.

Для сохранения адекватности расчета на каждой итерации пересылаются неизвестные, найденные для тел с полным набором ограничений в одной подмодели, в соответствующие неизвестные в другой подмодели, которые найдены без учета ограничений (рис. 3).

Представленный алгоритм был реализован в программном комплексе моделирования ФРУНД [4]. Алгоритм основан на подходе синхронизации параллельных процессов с использованием управляющего монитора. Программная реализация представляет собой подсистему с клиент-серверной архитектурой (рис. 4).

Алгоритм использует две точки барьерной синхронизации:

- пока все данные не получены сервером, их пересылка клиентам блокируется;
- перед расчетом все данные должны быть получены клиентами от сервера.

Для точного решения уравнений необходимо пересылки проводить на каждой итерации. Для типовой задачи требуется расчет большого числа итераций ($> 10^4$). При увеличении числа тел в модели размерность матрицы увеличивается, т. е. растет вычислительная сложность, которая для данной задачи линейна ($O(n)$). Необходимость в увеличении числа пересылок возникает, когда

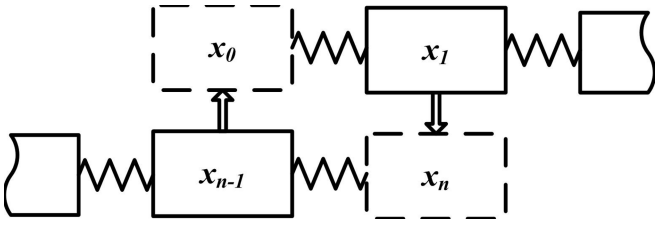


Рис. 3. Схема передачи данных для разделенной модели ленты

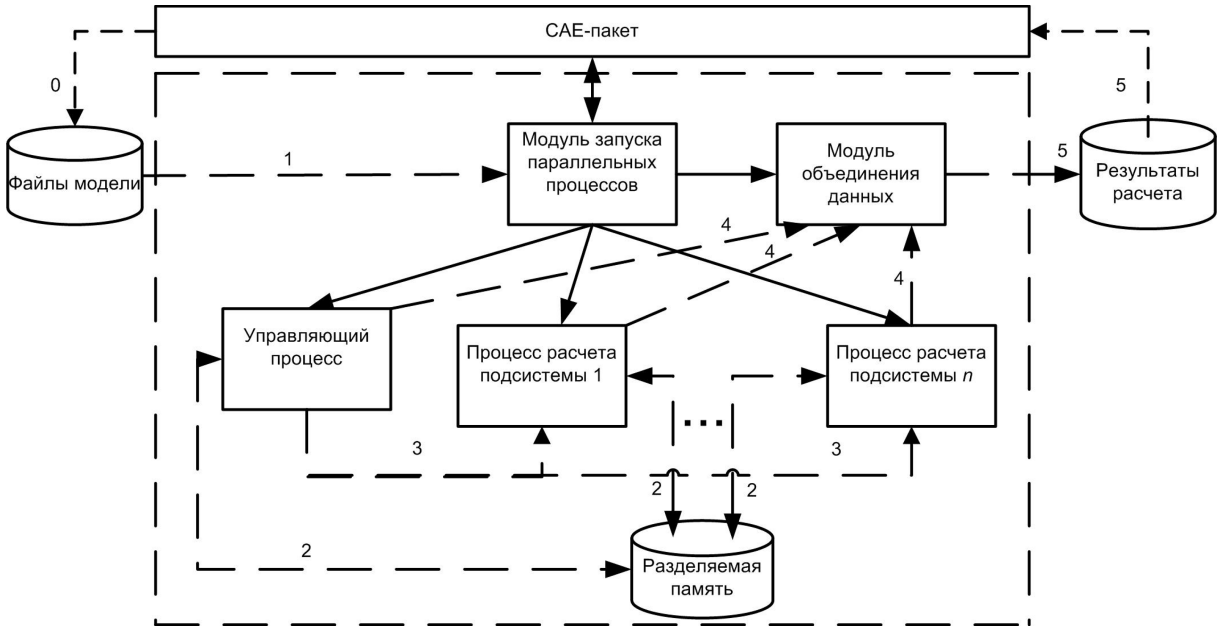


Рис. 4. Архитектура системы синхронизации параллельных расчетов:
0 — параметры и геометрическое описание полной модели; 1 — данные адресации результатов для модели; 2 — пересылаемые между процессами данные; 3 — таблица синхронизации; 4 — результаты расчета подмоделей; 5 — собранные результаты расчета полной модели

размерность подмодели достигает максимально возможного значения $n <= N$, т. е. время пересылок по сравнению со временем расчета при повышении размерности растет несущественно, поэтому полученное замедление с ростом размерности будет падать. Для проверки адекватности метода был проведен ряд экспериментов на модели разной размерности от 10 до 150 элементов, при этом замедление снизилось от 30 до 5 раз (рис. 5).

Серия расчетов проводилась при фиксированном числе итераций на однопроцессорной машине. При использовании многопроцессорных компьютеров можно добиться ускорения вычислений за счет деления времени параллельной расчетной части на несколько независимых процессов. Коэффициент ускорения, показывающий, во сколько раз быстрее программа выполняется на параллельной машине, чем на последовательной, определяется формулой

$$S_p = \frac{(T_s + T_p)}{(T_s + T_p/N)} = \frac{1}{(S + P/N)}, \qquad (9)$$

где T_s — время выполнения последовательной части алгоритма; T_p — время выполнения параллельной части алгоритма; $S = T_s/(T_s + T_p)$ — относительная доля последовательной части; $P = T_p/(T_s + T_p)$ — относительная доля параллельной части.

С учетом того, что $S + P = 1$, можно записать:

$$S_p = \left(S + \frac{1 - S}{N}\right)^{-1} \xrightarrow[N \to \infty]{} \frac{1}{S}. \qquad (10)$$

Формула (10) выражает закона Амдала [2, 3] и показывает, что ускорение параллельной программы обратно зависит от доли последовательного кода в ней. Она справедлива и при программировании в модели общей памяти, и в модели передачи сообщений. Оценить эту величину из анализа текста программы практически невозможно. Такую оценку могут дать только реальные просчеты на различном числе процессоров.

Задачи большой размерности в CAE-системах хорошо масштабируемы, так как каждая подсистема считается на каждой итерации независимо от других. Доля параллельного кода прямо зависит

Рис. 5. Зависимость замедления от размерности

Рис. 6. Фрагмент кинограммы движения цепной системы из 900 тел

от сложности подсистем. Основную часть в последовательном коде занимает работа сервера и обмен данными клиентов с сервером. По закону Амдала доля последовательного кода должна быть минимальной. В нашем случае это контролируется размерностью и сложностью задачи. Чем больше вычислений требуется на каждой итерации, тем меньше доля последовательного кода, которая остается практически постоянной при разбиении разных моделей на одинаковое число частей.

Таким образом, рассмотренный подход можно эффективно применять для решения задач большой размерности на системах тел с большим числом степеней свободы и с большим количеством вычислений для каждой из частей. В идеальном случае время расчета одной итерации для каждой части не должно превышать времени пересылки данных после каждой итерации. Такое возможно при сбалансированном разбиении модели на равные по сложности части и при минимальном числе граничных блоков в матрице инциденции.

В системе моделирования ФРУНД были решены несколько тестовых задач большой размерности, которые подтвердили принципиальную работоспособность алгоритма. Для примера на рис. 6 показан фрагмент анимации расчетного движения цепной системы из 900 тел под действием импульсного нагружения.

Заключение

Для рассмотренного метода распараллеливания можно указать две возможные области эффективного применения: во-первых, это задачи управления в робототехнике [6] и, во-вторых, задачи механики распределенных систем большой размерности [5].

Используемый в теории автоматического управления метод обратной задачи требует параллельного решения нескольких математических моделей достаточно большой размерности, решение которых в рамках одного вычислительного потока приводит к созданию громоздкой программы, модули которой функционируют как конечные автоматы. Использование формы уравнений (4) позволяет организовать параллельное решение каждой подзадачи с использованием сетевых или многопроцессорных технологий. При разбиении каждой отдельной подзадачи в форме (5) можно обеспечить выход на алгоритмы управления реального времени в рамках многопроцессорной технологии.

В задачах механики распределенных систем предлагаемый метод распараллеливания снимает ограничения на размерность рассматриваемой системы (вопросы быстродействия требуют отдельного рассмотрения) и позволяет значительно расширить общность постановки задачи, например, получать формулировки задач динамики систем твердых и упругих тел, в которых упругие свойства тел учитывают физическую и геометрическую нелинейности. Такие формулировки могут значительно повысить содержательность математических моделей, которые используются, например, при проектировании машин. Кроме этого, снятие ограничений с размерности моделей может позволить получать такие модели из существующих систем геометрического моделирования с помощью специальных интерфейсов, использующих общепринятые форматы передачи геометрической информации.

Список литературы

1. **Банах Л. Я.** Условия разбиения системы дифференциально-алгебраических уравнений на слабосвязанные подсистемы / Л. Я. Банах, А. С. Горобцов, О. К. Чесноков // Журнал вычислительной математики и математической физики. — 2006. — Т. 46, № 12. — С. 2223—2227.
2. **Бочаров Н. В.** Технология и техника параллельного программирования [Электронный ресурс] / Н. В. Бочаров. — [2007]. — Режим доступа: http://dks.invitation.ru.
3. **Букатов А. А.** Программирование многопроцессорных вычислительных систем / А. А. Букатов, В. Н. Дацюк, А. И. Жегуло. — Ростов-на-Дону: ЦВВР, 2003. — 208 с.
4. **Гетманский В. В.** Решение задач большой размерности в системах моделирования многотельной динамики с использованием параллельных вычислений / В. В. Гетманский, А. С. Горобцов //. — Волгоград 2007, Известия ВолгГТУ. Сер. "Актуальные проблемы управления, вычислительной техники, информатики и вычислительных систем". Вып. 3. № 9 (35). С. 9—12.
5. **Горобцов А. С.** Алгоритмы численного интегрирования уравнений движения систем тел с множителями Лагранжа / А. С. Горобцов, С. В. Солоденков // Машиностроение и инженерное образование. — 2005. — № 3. — С. 20.
6. **[Электронный ресурс]** /Режим доступа: http://www.frund.vstu.ru.
7. **Горобцов А. С.** Синтез параметров управляемого движения многозвенных механических систем произвольной структуры методом обратной задачи / А. С. Горобцов // Мехатроника, автоматизация, управление. — 2004. — № 6. — С. 43—50.

УДК 004.5

Э. А. Трахтенгерц, д-р техн. наук, проф.,
Институт проблем управления
им. В. А. Трапезникова РАН
E-mail: tracht@ipu.rssi.ru

Компьютерная технология реализации динамики информационного управления в конфликтных ситуациях. Часть I. Информационные оперативные воздействия

Рассматриваются компьютерные технологии реализации информационных оперативных воздействий в процессе информационного управления в конфликтных ситуациях.

Ключевые слова: *компьютерная технология, информационное управление, информационные оперативные воздействия, конфликтная ситуация.*

Введение

Информационные воздействия — распространение определенной информации как средства или одного из средств достижения цели — получили очень широкое распространение на самых различных уровнях: межгосударственных, транснациональных корпораций, политических партий, финансовых, политических групп, коммерческих организаций, спортивных клубов и даже личных отношений. Проведение таких воздействий получили в литературе название информационного управления в конфликтных ситуациях, а также информационных, психологических, информационно-психологических и т. п. войн [2, 4—7, 11, 14]. На их реализацию затрачиваются очень большие средства, а при их проведении обрабатываются и генерируются огромные объемы информации зачастую в сжатые сроки. В связи с этим возникла острая необходимость создания и использования компьютерных информационных технологий для реализации и управления этими процессами.

Динамика информационного управления в конфликтных ситуациях требует не только реализации оперативных и стратегических решений информационного управления, но и оценки их эффективности, а также модификации стратегий и целей в случае их неадекватности сложившейся обстановке.

Существует несколько возможных источников информации о назревающей необходимости таких модификаций: тенденции изменения в освещении деятельности организации, ее оценки в СМИ, тенденции изменения экономических показателей и т. д. Их компьютерный анализ позволяет предвидеть необходимые изменения и своевременно подготовить соответствующие решения. Поэтому не дожидаясь получения полной информации о происходящих изменениях, руководству организации следует определить, какие шаги могут быть предприняты при различных вариантах развития событий для парирования возникающих информационных угроз и использования появляющихся возможностей [16].

Сформировать оперативные воздействия в процессе управления обычно бывает проще, чем модифицировать или менять стратегию, а тем более цель, поскольку они формируются на основе

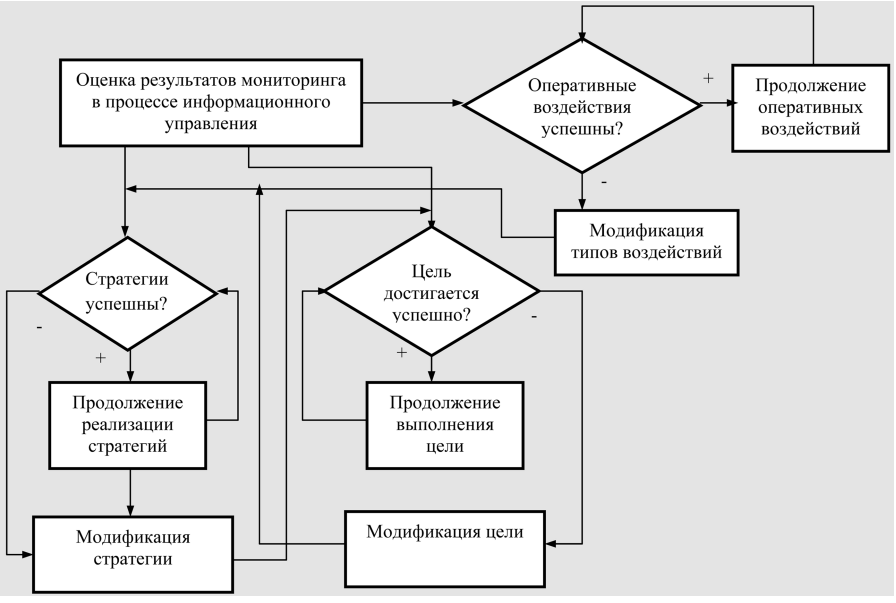


Рис. 1

заранее принятых правил. Оперативные информационные воздействия могут быть как регулярными, например, ежедневные публикации в газете, так и нерегулярными, вызванными какими-нибудь информационными поводами. Модификация стратегии проводится только в тех случаях, когда выполнить задачи стратегии только оперативными воздействиями не удается. Поэтому она проводится нерегулярно и не определяется числом ранее проведенных оперативных воздействий.

Цель модифицируют, если по результатам мониторинга обнаружена ее неадекватность создавшейся обстановке. После того как цель модифицирована, система поддержки принятия решений (СППР) может изменить одну или несколько стратегий ее реализации, поскольку цель может придать уникальность и оригинальность выбору стратегии ее реализации.

Связь между оперативным информационным воздействием, реализацией стратегий и целью информационного управления показана на рис. 1.

На рис. 1 "+"означает успешность, а "−" — неуспешность; действия при достижении цели не показаны.

Анализ динамики начнем с определения информационных оперативных воздействий, во второй части статьи перейдем к реализации стратегий и цели.

1. Предварительное определение возможных значений параметров информационных оперативных воздействий экспертным путем

Формирование оперативных информационных воздействий осуществляется на основе обработки и анализа данных, полученных в результате мониторинга и в соответствии с целью и стратегией информационного управления. При прямом регулировании технологических процессов в АСУ, осуществляемом по адекватным математическим моделям, значение оперативного воздействия достаточно точно определяет изменение регулируемых параметров. Правильно предсказать значение изменения регулируемых параметров информационных, экономических, политических или социальных процессов в зависимости от значений информационных оперативных воздействий возможно далеко не всегда. Это объясняется тем, что информационное воздействие должно оказать влияние на информационные, экономические, политические и социальные процессы, в которых часто нарушается стабильность, и точность математических моделей невелика.

Сначала определим параметры, характеризующие информационное оперативное воздействие V. Пусть это будут: m — тип информационного оперативного воздействия, выбранный из заранее составленного списка, хранящегося в базе данных; l — объем публикуемой информации; S — тип элемента СМИ (наименование телеканала, радиоканала, газеты и т. п.).

Конечно, это не все параметры, которые надо учитывать. Например, важную роль играет время и день передачи информации, номер страницы газеты, на которой она опубликована, в какие передачи или статьи они встроены и т. д. Но для простоты рассуждений ограничимся параметрами m, l и S и будем оценивать эффективность информационных воздействий в зависимости от этих параметров по традиционной лексической шкале (возможно, конечно, по любой другой).

Будем считать, что список типов информационного оперативного воздействия определяется руководителем или экспертом на содержательном уровне и является данными, вводимыми в систему. Такой список приведен, например, в работе [4]. Тогда выбор параметров l и S СППР можно определить из соотношения

$$\sum_{m \in M'} \sum_{l \in L'} \sum_{s=1}^{S} \gamma_m^{l,S} V_m^{l,S} \to \max \qquad (1)$$

при ограничениях

$$\left.\begin{gathered} \gamma_m^{l,S} = \{0, 1\} \\ \sum_{l=1}^{L} \gamma_m^{l,S} \leqslant 1,\ m = \overline{1, M'},\ s = \overline{1, S} \end{gathered}\right\}, \qquad (2)$$

$$\sum_{l=1}^{L} \sum_{s=1}^{S} \gamma_m^{l,S} c_m^{l,S} \leqslant \bar{c}_m,\ m = \overline{1, M'},$$

где M' — подмножество типов информационных оперативных воздействий, используемых в выбранной стратегии; L — объемы публикуемой информации m-го типа воздействий; $c_m^{l,S}$ — цена публикаций m-го типа информационного оперативного воздействия l-го объема в s-й газете, радио или телеканале; $\bar{c}_m$ — финансовые ограничения на информационное оперативное воздействие m-го типа.

Для вычисления функции (1) необходимо знать значения $V_m^{l,S}$, $m = \overline{1, M}$, $l = \overline{1, L}$, $s = \overline{1, S}$. Эти значения могут быть предварительно определены экспертным путем и внесены в соответствующую таблицу либо вычислены по какому-либо эвристическому алгоритму. Определение этих значений − процесс достаточно трудоемкий, но без этой трудоемкости трудно правильно определить тип информационных воздействий, набор используемых СМИ, объемы передаваемой информации и другие параметры. Для их определения проводится комплексный сбор, регистрация и анализ результатов реализации предыдущих этапов оперативных информационных воздействий.

Методами исследований эффективности оперативных воздействий могут быть: кабинетные исследования, интервью, анализ фокус-групп, экспертные оценки, анкетирование и т. д. [14].

Основной смысл кабинетных исследований состоит в сборе и анализе вторичных данных. В кабинетных исследованиях данные всегда бывают нецелевыми, поскольку создаются не в ходе исследования, а берутся из других источников уже готовыми. Источниками данных являются публикации и статьи, издания электронных и печатных СМИ, специальные издания и отчеты, Интернет и т. д. Проведение кабинетных исследований, как правило, требует меньше времени, чем их непосредственное изучение. Стоимость кабинетных исследований невелика. Преимущество состоит также в возможности получения информации о проблемах, недоступных прямому изучению.

Интервью может проводиться в различных формах. Личные интервью могут быть как формализованные, так и неформализованные. При формализованном интервью используется конкретная схема проведения опроса. Обычно это опросный лист, содержащий заранее подготовленные четкие формулировки вопросов. Формализованное интервью эффективно при определении социальных и демографических характеристик. Наибольшее применение формализованные опросы получили при проведении количественных исследований. Основными недостатками данного метода являются высокая стоимость и незначительный географический охват.

Неформализованные интервью — это специфический метод сбора информации, при котором определены только тема и цель. Конкретной схемы проведения опроса нет. Это дает возможность понять респондента, изучить как рациональные, так и иррациональные мотивы и причины поведения. Неформализованные интервью могут быть индивидуальными и групповыми.

При анализе фокус-групп респондентов просят не только оценить что-либо по принципу "нравится — не нравится", но и объяснить свою точку зрения. Последующий анализ полученных результатов позволяет понять мотивы и механизмы формирования позиции участника фокус-группы. Другие методы анализа комментариев не требуют.

На основании полученных исследований можно получить экспертные оценки значений $V_m^{l,S}$, $m = \overline{1, M}$, $l = \overline{1, L}$, $s = \overline{1, S}$, а затем по формуле (1) с учетом ограничений (2) определить рациональные характеристики информационных оперативных воздействий.

В зависимости от результатов исследований и мониторинга информационные оперативные воздействия могут проводиться: более агрессивно (расходы на них увеличатся); без изменений; менее агрессивно (расходы на них могут быть сокращены) или продолжение информационных воздействий будет признано нецелесообразным.

Может оказаться полезным ввести индекс, определяющий степень согласованности [13] значений и знаков наблюдений (измерений) с предсказываемыми данными относительно среднего значения параметров полученных наблюдений:

$$d = 1 - \frac{\sum_{i=1}^{N} (P_i - S_i)^2}{\sum_{i=1}^{N} (|P_i - E| + |S_i - E|)^2},$$

где N — число измерений (наблюдений); P_i — значение i-го предсказания; S_i — значения i-го наблюдения; E — среднее значение измеренных данных. Этот индекс может помочь адекватности прогнозирования.

Проведение информационных оперативных воздействий для уменьшения напряженности в обществе бывает важно и при осуществлении социальных и экономических реформ, которые далеко не всегда воспринимаются одобрительно. Так, принятие закона об изменении порядка найма и увольнения молодых специалистов во Франции в 2006 г. вызвало бурю протестов не только студентов, но и профсоюзов, а попытка правительства несколько смягчить закон в пользу молодых специалистов (но не отменять его вообще) вывела на улицы до 3 000 000 демонстрантов, требующих его полной отмены. Введение в 2005 г. закона о монетизации льгот в России вследствие несогласованности действий федеральных и местных властей также вызвало акции протеста. Такие примеры можно умножить.

2. Реализация оперативных воздействий в динамике информационного управления и анализ их эффективности

В этом разделе рассмотрим на конкретных данных формирование СППР оперативных информационных воздействий с использованием метода компьютерной игры. Заметим, что при формировании информационного воздействия этим методом вмешательство эксперта или руководителя не требуется.

В табл. 1 показаны данные мониторинга о поставках на рынок продукции атакуемой фирмы за период τ^k и τ^{k+1} и реакции на них рынка по трем позициям: объем проданной продукции; среднее изменение цен на эту продукцию; объемы новых заказов. Будем считать, что оперативные воздействия представляют собой следующую комбинацию типов воздействий: «приоритет негативной

Таблица 1

Периоды τ	Индекс вида продукции j	Объем поставок произведенной продукции фирмы R_j^k, тыс. руб.	Цена единицы продукции P_j^k, руб.	Реакция рынка на информационные оперативные воздействия атакующей системы		
				Объем проданной продукции W_j^k, тыс. руб.	Среднее изменение цен на единицу продукции A_j^k, руб.	Объем заказов G_j^k, тыс. руб.
1	2	3	4	5	6	7
k	1 ... 4 ... 8 9	210 ... 460 ... 480 90	150 ... 210 ... 170 80	210 ... 430 ... 470 50	+10 ... 0 ... 0 −15	240 ... 450 ... 420 30
$k+1$	1 ... 4 ... 8 9	330 ... 450 ... 420 20	230 ... 205 ... 166 60	246 ... 460 ... 450 5	+80 ... −5 ... −4 —	280 ... 450 ... 420 0

информации», «прямая ложь» и «наклеивание ярлыков». Назовем такое сочетание оперативным информационным воздействием типа 12, а его эффективность обозначим $V_{12}^{l,S}$.

Цели рынка и фирмы-производителя в нашем случае как будто совпадают: произвести и продать как можно больше товаров и услуг по максимально возможной цене. Но поведение рынка определяется многими факторами. Для простоты рассуждений будем считать, что к моменту τ^k основным возмущением стали информационные воздействия на фирму-производителя, значения которых определены по формулам (1) и (2). Рынок ответил на эти воздействия сокращением сбыта продукции атакованной фирмы типа $j = 4$ и $j = 8$, прекращением выпуска продукции типа $j = 9$, но при этом спрос на продукцию типа $j = 1$ не только не сократился, но даже возрос. (Наименование типа продукции в данном случае не имеет значения).

Анализ данных проведем методом компьютерной игры в терминах работ [9, 10]. Два шага такой игры представлены на рис. 2. Рис. 2 показывает как информационные воздействия $V_{12,j}^k$, осуществленные в период τ^k, сказались на спросе и производстве продукции фирмы, подвергшейся информационным воздействиям в период τ^{k+1}.

Примечание. Матрица R^k помещена вне рисунка, чтобы его не загромождать. Данные о продукции типа $j = 9$ за период τ^{k+1} не показаны, так как производство продукции прекращено.

На рис. 2 объем поставок продукции j-го типа, произведенной фирмой и полученной рынком за период τ^k, обозначен R_j^k; объем проданной продукции — W_j^k; объем продукции, заказанной рынком фирме — G_j^k; изменение цен на продукцию — $A_j^k = P_j^{k-i} - P_j^k$, где P_j^k — цена единицы продукции (столбец 4 табл. 1). Значение параметров l и S информационного воздействия $V_{12,j}^k$ = 1, 4, 8, 9 показаны типом СМИ и объемом публикации или продолжительностью передачи.

Рассмотренный ниже алгоритм аналогичен процессу определения равновесной цены, иногда называемой "нащупыванием" [1]. Он заключается в следующем. Если известны функции совокупного спроса $\Phi(p)$ и совокупного предложения $\Psi(p)$ безотносительно от возможного изменения их причин, то последовательность $\{p_s\}$ строится следующим образом. Пусть при некоторой цене p_s спрос выше предложения. Тогда повышаем цену до значения p_{s+1}, чтобы выполнялось условие $\Phi(p_{s+1}) = \psi(p_s)$ (или $\psi(p_{s+1}) = \Phi(p_s)$). Если же спрос ниже предложения, то, наоборот, цену понижаем. Таким образом, каждый член p_s последовательности $\{p_s\}$ строится как решение уравнений относительно p:

$$\Phi(p_s) - \psi(p_{s-1}) = 0 \text{ или } \Phi(p_{s-1}) - \psi(p_s) = 0. \quad (3)$$

Корень этого уравнения и является следующим членом последовательности. В качестве начального приближения p_0 можно взять любое значение $p > 0$. Поскольку эти функции нам неизвестны, будем варьировать значения P_s, "нащупывая" нужные значения оперативных воздействий.

Результаты оперативных воздействий целесообразно использовать как для анализа реакций атакуемой фирмы, так и для определения эффективности оперативных информационных воздействий. Метод компьютерной игры и метод "нащупывания" позволяют реализовать эти задачи. В графе рис. 2 использованы данные о ходе про-

даж и информационных оперативных воздействиях за два периода τ^k и τ^{k+1}.

Период τ^k. Производство фирмы в период τ^k, показанное матрицей R^k (объемы произведенной продукции — столбец 3 табл. 1), является производственным ответом фирмы на информационные оперативные воздействия атакующей системы в период τ^{k-1}.

Реакция рынка отображена тремя параметрами: объемом проданной продукции W_j^k (столбец 5 табл. 1), изменением цен на продаваемый товар A_j^k (столбец 6 табл. 1) и коммерческим воздействием рынка на фирму, который косвенным образом также подвергается информационному воздействию атакующей системы через объем заказываемой продукции G_j^k (столбец 7 табл. 1). Информационные воздействия отображены параметрами $V_{12,j}^k$, которые условно считаются одним из важнейших факторов изменения параметров рынка.

Период τ^{k+1} отображен аналогично, но в нем не показаны информационные воздействия $V_{12,j}^{k+1}$, так как их результат надо отображать в период τ^{k+2}.

На основании данных об уровне продаж и заказов в последовательные периоды τ^k и τ^{k+1} СППР определяет значение изменения расходов по проведению оперативных информационных воздействий m-го типа на производство продукции j-го вида фирмы, подвергшейся информационным воздействиям в период τ^{k+1}, по формуле, связывающей расходы на информационное оперативное воздействие с изменением объемов продаж продукции фирмы-изготовителя и заказов на нее рынка:

$$\Delta C_{m,j}^{k+1} = \eta(\alpha_j^k | W_j^k - G_j^{k+1}|),$$
$$\text{если } W_j^k > G_j^{k+1},\ j = \overline{1,J};$$
$$\Delta C_{m,j}^{k+1} = \mu(\beta_j^k | W_j^k - G_j^{k+1}|),$$
$$\text{если } G_j^{k+1} > W_j^k,\ j = \overline{1,J},$$

тогда $C_{m,j}^{k+1} = C_{m,j}^k + \Delta C_{m,j}^{k+1}$.

Возвращаясь к алгоритму "нащупывания" значения расходов в уравнениях (3), еще раз подчеркнем, что, варьируя коэффициенты α_j^k и β_j^k, СППР "нащупывает" нужные значения $C_{m,j}^{k+1}$ так, чтобы в результате оперативных воздействий, вычисляемых по формулам (1) и (2), максимировать объемы непроданной продукции и неоказанных услуг фирмы-производителя.

Заметим, что в нашем примере информационные оперативные воздействия произвели нужный эффект на продукцию типа $i = 4, 8, 9$ фирмы, подвергшейся информационным воздействиям, и совершенно не повлияли на сбыт продукции типа $j = 1$. Можно даже сказать, что по продукции типа $j = 1$ они дали обратный результат. Поэтому целесообразно положить $C_{12,9}^{k+1} = 0$, $C_{12,4}^{k+1} = C_{12,4}^k$, $C_{12,8}^{k+1} = C_{12,8}^k$, т. е. прекратить компрометацию продукции типа $j = 9$ (ее выпуск прекращен), а расходы на информационные воздействия, связанные с продукцией типа

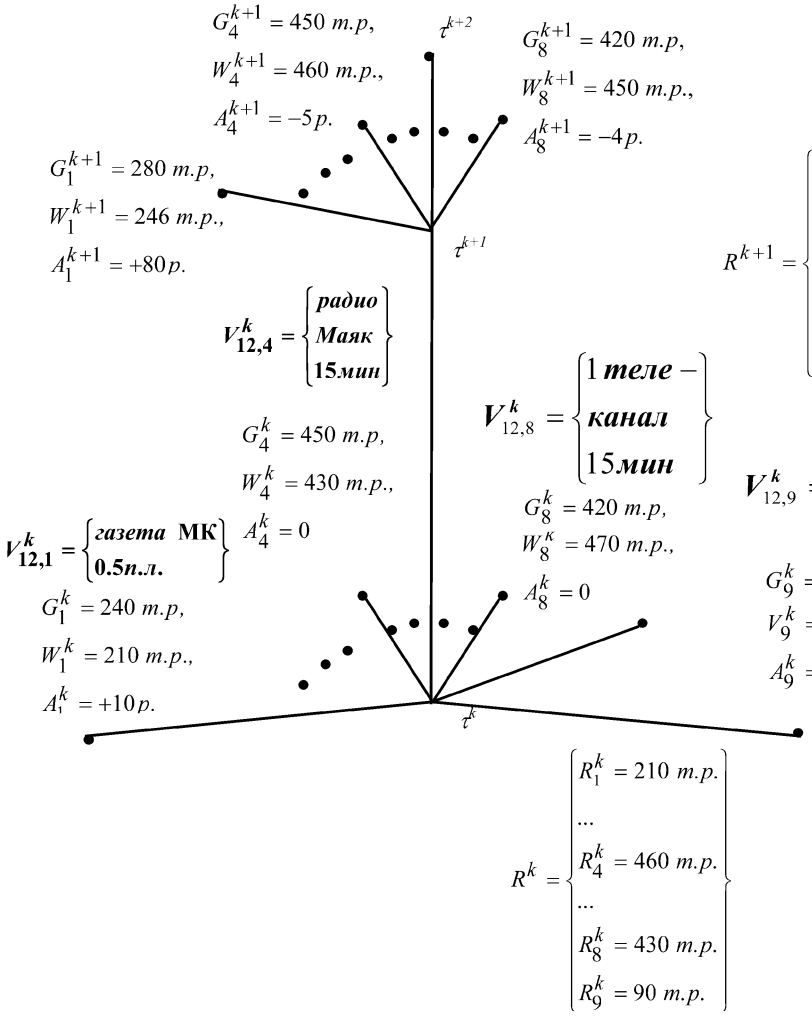


Рис. 2

$j = 4$ и 8, оставить на прежнем уровне. Эти расчеты можно было бы провести и на более формальном уровне за счет варьирования коэффициентов α_j^k и β_j^k при анализе результатов в периоды τ^{k-1}, $l = 1, 2, \ldots$ [1].

Что касается информационных воздействий по компрометации продукции типа $j = 1$, то тут требуются дополнительные исследования экспертов для определения возможных стратегий информационного воздействия: следует ли увеличивать значение $C_{12,1}^{k+1}$ или прекратить информационные воздействия на этот тип продукции, положив $C_{12,1}^{k+1} = 0$.

Иногда в процессе анализа эффективности информационных воздействий требуется определить только момент изменения тенденции развития ситуации, требующий модификации параметров информационного оперативного воздействия.

Приведем пример простого метода, определяющего такой момент [10]. Пусть в базе данных хранятся данные $x(t)$ за интервал времени τ, фиксированные в N моментов за время t. Определим скользящее среднее

$$M_\tau(x(t)) = \frac{1}{\tau} \sum_{i=t}^{t+\tau-1} x(i-\tau),\ t = \tau + l, \ldots, N, \quad (4)$$

т. е. среднее значение x за последние τ моментов, считая для простоты, что интервалы между фиксируемыми моментами времени равны. Легко показать, что если значения $x(t)$ возрастают, то $M_\tau(x(t)) < x(t)$, а если убывают — $M_\tau(x(t)) > x(t)$. Очевидно, что, как всякое статистическое среднее, $M_\tau(x(t))$ зависит от числа учитываемых точек.

Функция (4) может быть использована для определения по какому-нибудь одному критерию момента времени подачи сигнала о необходимости изменения характера информационных воздействий. Пересечение кривой, характеризующей, например, цену акций $x(t)$, с кривой, описывающей $M_\tau(x(t))$, может определять следующие решения:

- если пересечение происходит в области локального минимума кривой $x(t)$, то не надо менять характер оперативных информационных воздействий, так как их эффективность будет увеличиваться;
- если пересечение происходит в области максимума кривой $x(t)$, то характер оперативных информационных воздействий надо менять (например, менять используемые СМИ, так как эффективность информационных воздействий будет падать).

Ниже приводится пример более сложной оценки эффективности тренда информационных оперативных воздействий путем построения эталонных векторов эффективности. Пусть эксперты с помощью СППР согласовали следующие критерии оценки эффективности информационных воздействий:

- падение курса акций атакуемой фирмы;
- падение рейтинга руководителя атакуемой фирмы;
- сокращение продаж фирмы;
- соответствие фазе информационных воздействий;
- соответствие оперативных воздействий требованиям стратегий.

Для оценки эффективности оперативных воздействий используем метод распознавания образов [3]. Построим векторы оценок эффективности информационных воздействий и представим их в табл. 2.

Заметим, что оценки в табл. 2 в определенном смысле инверсны, т. е. например, оценка "отлично" "падения курса акций" означает, что акции упали очень сильно, что и требовалось при информационной атаке фирмы. Для простоты будем считать, что все критерии имеют один "вес". Естественно, что в результате информационных воздействий значение критериев оценки в общем случае не совпадают ни с одним вектором оценки эффективности. Поэтому будем искать оценку эффективности в определенном смысле наиболее близкую к результатам воздействия.

Для этого введем меру близости между i-м вектором эффективности табл. 2 и оценкой результатов l информационных воздействий:

Таблица 2

Значения критериев					Оценка эффективности информационных оперативных воздействий	Индекс вектора
Падение курса акций	Падение рейтинга руководителя	Сокращение продаж	Оценка соответствия фазе информационных воздействий	Оценка соответствия оперативных воздействий стратегии		
1	2	3	4	5	6	7
Отлично Хорошо Удовл. Плохо	Хорошо Хорошо Удовл. Плохо	Хорошо Удовл. Удовл. Удовл.	Отлично Удовл. Удовл. Плохо	Отлично Хорошо Удовл. Плохо	Отлично Хорошо Удовл. Плохо	1 2 3 4

Таблица 3

Оценка тренда эффективности информационных оперативных воздействий	Возможные изменения характера информационных оперативных воздействий			
	Оценка эффективности последнего воздействия			
	Отлично	Хорошо	Удовлетворительно	Плохо
Отлично	Не менять	Слегка увеличить степень остроты	—	—
Хорошо	Не менять	Слегка увеличить объемы информации и степень остроты	Увеличить объемы информации в СМИ, пользующиеся наибольшим влиянием, и степень остроты	—
Удовлетворительно	—	Значительно увеличить объемы информации и степень остроты	Модифицировать характер информационных операционных воздействий, список используемых СМИ и рассмотреть необходимость модификации стратегий	
Плохо	Модифицировать реализуемые информационные стратегии			

$$d(r_i, g_l) = \sqrt[\wedge]{\sum_{j=1}^{J} k_j (r_i^j - g_l^j)^2},$$

где k_j — "вес" j-го критерия; r_i^j — значение j-го критерия i-го вектора оценки эффективности табл. 2; g_l^j — значение j-го критерия оценки результатов l информационных воздействий.

Для распознавания системой близости оценки l информационных воздействий к i-й оценке эффективности введем функцию

$$\mu(r_i, g_l) = \min_{i \in I} d(r_i, g_l), \qquad (5)$$

где I — множество векторов оценки эффективности (см. табл. 2), т. е. система сравнивает вектор g_l со всеми векторами r_i. Индекс i, для которого функция $\mu(r_i, g_l)$ достигает минимума и определяет оценку эффективности проведенных l информационных воздействий.

В зависимости от оценки эффективности, полученной по соотношению (5) и табл. 2, система может дать рекомендации по модификации параметров информационного оперативного воздействия, например, в соответствии с табл. 3, заранее составленной экспертами.

Рекомендации, аналогичные данным в табл. 3, позволяют экспертам сосредоточиться на содержании генерируемой информации, автоматически формируя рекомендации по объемам информации, привлекаемым типам СМИ и т. п. Они позволяют также связать успешность выполнения информационных оперативных воздействий с необходимостью модификации стратегий.

Заключение

Используя множество возможных значений параметров информационных оперативных воздействий, предварительно определенное экспертным путем, СППР может на основе анализа сложившейся обстановки определить типы воздействий, конкретные СМИ, объемы публикуемой информации, оценить их эффективность и, в случае необходимости, формировать предложения по изменению характера информационных оперативных воздействий и модификаций стратегий.

Список литературы

1. **Альсевич В. В.** Введение в математическую экономику. М.: УРСС. 2004.
2. **Бухарин С. Н., Цыганов В. В.** Методы и технология информационных войн. М.: Академический проект, 2007.
3. **Горелик А. Л., Скрипкин В. А.** Методы распознавания. М.: Высшая школа, 2004.
4. **Донской М.** Методы ведения информационной войны // Прогнозис. 2006. № 2. http://www.itblogs.ru/blogs/donskoy/arhive/2006/10/20/8132/aspx.
5. **Кульба В. В., Малюгин В. Д., Шубин А. Н.** и др. Введение в информационное управление. СПб.: Изд. С.-Петербургского университета. 1999.
6. **Манойло А. В.** Информационно-психологическая война как средство достижения политических целей // E-mail: amanoilo@finserv.ru.
7. **Погециев Г. Г.** Информационные войны. М.: Изд-во МГУ, 2004. library.htm.
8. **Расторгуев С. П.** Информационная война. М.: Радио и связь, 1999.
9. **Трахтенгерц Э. А.** Компьютерная поддержка принятия решений. М.: СИНТЕГ, 1998.
10. **Трахтенгерц Э. А., Иванилов Е. Л., Юркевич Е. В.** Современные компьютерные технологии управления информационно-аналитической деятельностью. М.: СИНТЕГ, 2007.
11. **Ausloos M., Ivanova K.** Mechanistic approach to generalized technical analysis of share prices and stock market indices // The European Physical Journal B. 2002. V. 27. P. 177—187.
12. **Chomsky W., Barsamien D.** Propoganda and the public mind. Cambridge, MA: South End Press, 2000.
13. **Elbir T.** Comparison of model predictions with the data of an urban air quality monitoring network in Izmir Turkey // Atmospheric Environment. 2003. 37. P. 2149—2157.
14. **http://www.informanaliz.ru/cat/182.**
15. **Mazzoco D. W.** Networks of power: corporate TV's threat to democracy. Boston: MA: South End Press, 1994.
16. **Sigh S. K., Watson H. J., Watson R. T.** EIS support for the strategic management process // Decision Support Systems. 2002. V. 33. P. 71—85.

УДК 004.3

В. Г. Шахов, канд. техн. наук, проф.,
С. В. Нопин, аспирант,
Омский государственный университет
путей сообщения

Анализ защищенности абонентских систем IP-телефонии от несанкционированного доступа

Проводится анализ защищенности абонентских систем IP-телефонии от несанкционированного доступа. Получены данные об особенностях реализации защиты от несанкционированного доступа систем IP-телефонии, функционирующих в среде операционной системы Windows.

***Ключевые слова**: модель нарушителя, атака, инсталляция, перехват, пароль.*

Процесс передачи речи через IP-сети в защищенном режиме является результатом взаимодействия:

- злоумышленника (нарушителя), изучающего защиту системы, и злоумышленника, реализующего несанкционированный доступ (НСД);
- природных явлений, реализующих помехи;
- пакетов с данными, циркулирующими в системе;
- блоков обработки и передачи речевых пакетов, подлежащих защите.

При рассмотрении модели передачи речи по IP-сетям в защищенном режиме необходимо ввести понятие модели нарушителя. Кратко сформулируем определение модели нарушителя в соответствии с руководящими документами Гостехкомиссии России [1, 2].

Под *моделью нарушителя* будем понимать абстрактное (формализованное или неформализованное) описание нарушителя правил разграничения доступа.

Модель нарушителя определяет:

- типы нарушителей, которые могут воздействовать на объект;
- цели, которые могут преследовать нарушители каждого типа, возможный количественный состав, используемые инструменты, оснащение и проч.;
- типовые сценарии возможных действий нарушителей (атаки), описывающие последовательность (алгоритм) действий групп и отдельных нарушителей, способы их действий на каждом этапе.

Применительно к системе защищенной IP-телефонии в качестве классификационного признака может использоваться уровень квалификации нарушителя, его физический уровень доступа (внутренний или внешний нарушитель), используемые методы и средства (пассивные или активные). Кроме того, руководящие документы Гостехкомиссии России [1] предлагают классификацию нарушителей, основанную на возможностях, предоставляемых нарушителю штатными средствами информационной системы, т. е. на начальном уровне доступа к системе. Такой подход является оправданным при рассмотрении некоторых атак, поскольку уровень возможностей непосредственно определяет, какие атаки могут быть осуществлены нарушителем, а какие нет.

Рассматривая возможные стратегии действий нарушителя и атаки на систему передачи речи по IP-сетям, работающую в защищенном режиме и реализующую сценарий "компьютер — компьютер", мы будем предполагать, что нарушитель является специалистом высшей квалификации, знает все об операционных системах, сетевых протоколах, системе передачи речи по IP-сетям и ее средствах защиты, и в его распоряжении имеются максимально доступные на его уровне ресурсы [5, 10].

На примере системы IP-телефонии, работающей в защищенном режиме и реализующей сценарий "компьютер — компьютер", проведем анализ защищенности этой системы на основе ОС Windows и аппаратных возможностей современных ПЭВМ [7, 8, 9].

Под математической моделью атаки будем понимать ее формализованное описание, построенное с точки зрения принятой модели защищенности. При этом будем говорить о вероятностной и теоретико-игровой моделях атак. В рамках *вероятностной модели* значимыми будут вероятность предотвращения атаки системой защиты, вероятности ее обнаружения и локализации, другими словами, вероятность успешного завершения атаки. Эта вероятность в общем случае будет зависеть от времени; характер этой зависимости и будет составлять суть модели атаки. Перечисленным атакам соответствуют две математические модели: модель перебора и модель проверки.

Атаки, связанные с проверкой некоторого числа вариантов, можно описать моделью перебора. Типичным примером таких атак является харак-

терная для большинства современных систем, в том числе и для системы IP-телефонии, атака подбором пароля или ключа.

Атаки, основанные на ошибках (недостатках) в системе безопасности, и им подобные можно описать с помощью модели проверки. Такие атаки используют уязвимость системы защиты, проверяя единственный вариант — наличие или отсутствие данной уязвимости. Вероятность успешного завершения в этом случае не зависит от времени и целиком определяется наличием или отсутствием используемой уязвимости.

Такая математическая модель отражает большинство существующих в настоящее время атак, поскольку под понятие уязвимости подпадают как ошибки (недостатки) в системе безопасности, так и ошибки администрирования.

С точки зрения *теоретико-игровой модели* значение имеет ущерб, причиняемый применением той или иной атаки, и вероятность предотвращения этой атаки различными методами защиты. Ущерб будет основной характеристикой при описании атак с позиции этой модели. Проблема заключается в том, что оценить возможный ущерб бывает достаточно сложно, а эти оценки, как правило, оказываются частными и субъективными, поэтому распространить их на множество мест применения систем связи, защищенных от несанкционированного доступа, оказывается проблематично.

При анализе уязвимостей рассмотрим атаки, основанные не на ошибках в программном обеспечении, которые разработчик системы может исправить за считанные дни, а на концептуальных свойствах функционирования системы IP-телефонии, работающей в защищенном режиме.

Для удобства каждой атаке присвоим уникальный индекс, состоящий из префикса "А" и номера.

А1. Доступ к речевой информации в обход системы IP-телефонии, в обход ОС Windows и в обход ПЭВМ

Целью данной атаки является доступ к речевой информации, когда она еще не достигла ПЭВМ. На данном этапе нарушитель может перехватить голосовое сообщение как до аналого-цифрового преобразования (АЦП), так и после цифроаналогового преобразования. Перехват речи в аналоговой форме возможен с помощью различных подслушивающих устройств, установленных в непосредственной близости от ПЭВМ, либо с помощью подслушивающего устройства, имеющего непосредственное электрическое соединение с микрофоном, осуществляющим ввод речи [6]. Кроме НСД данный этап преобразования речи может быть подвержен воздействию помех. Сами помехи можно рассматривать как НСД к полезной информации, который имеет естественное происхождение. Природа этих помех имеет аналоговый и цифровой вид. Цифровые шумы возникают вследствие: интегральной и дифференциальной нелинейности АЦП звуковой карты; дрейфа частоты резонатора, определяющего выборки сигнала, шумов квантования. Аналоговая составляющая помех определяется как внутренними шумами устройства ввода/вывода звука, так и шумами, наведенными внешними сигналами. Таким образом, вероятность воздействия помех всегда равна 1, однако применительно к реально функционирующей системе передачи речи воздействие помех на этапе ввода/вывода речи важно для нас потому, что определяет ее качество. Усредненная экспертная оценка (УЭО) качества речи выводится из оценок отдельных экспертов по пятибалльной шкале: 5 — отлично, 4 — хорошо, 3 — удовлетворительно, 2 — неудовлетворительно, 1 — плохо. Будем считать, что при оценке меньше 3 баллов вероятность успеха атаки НСД к полезной информации равна 1, иначе равна 0.

Вероятность успеха атаки НСД определяется условиями, в которых функционирует система IP-телефонии. На практике обнаружить или предотвратить эту атаку возможно лишь при одновременном использовании технических и организационных и инженерно-технических методов.

Из технических методов наиболее эффективным действием против прослушивания является постоянная проверка помещения на наличие несанкционированных подслушивающих устройств с помощью специализированных детекторов-радиопеленгаторов. Для борьбы с естественными внешними шумами может использоваться звуковая изоляция помещения (рабочего места), для борьбы с естественными внутренними шумами аппаратуры — более качественное программно-аппаратное обеспечение системы передачи речи.

Предотвратить атаку прослушивания могут организационные и инженерно-технические методы, направленные на обеспечение надежной защиты помещения против физического проникновения в помещение случайных лиц, способных принести записывающие или радиопередающие устройства.

Обнаружить первую и вторую часть рассмотренной атаки возможно. В случае пассивной записи можно физически обнаружить записывающие устройства; в случае, если запись речи и передача ее осуществляются через радиоканал, можно выявить подозрительный радиотрафик с помощью прослушивания радиочастотного спектра, также можно локализовать радиопередатчик или другое полупроводниковое устройство с помощью детекторов радиоизлучения [6]. Обнаружить низкое качество речи можно на слух с помощью усредненных экспертных оценок качества речи для конкретных реализаций систем IP-телефонии, которые создаются специально обученными дикторами и операторами.

А2. Доступ к речевой информации в обход системы IP-телефонии

Целью атаки является доступ к речевой информации, когда она достигла ПЭВМ и преобразована в отсчеты (оцифрована). На данном этапе нарушитель может перехватить голосовое сообщение как после аналого-цифрового преобразования, так и до цифроаналогового преобразования. Перехват речи в цифровой форме возможен в случае использования специализированной звуковой карты и/или специализированного программного обеспечения, осуществляющего на низком уровне управление звуковой картой.

Возможны следующие варианты передачи перехваченной информации:

- данные (оцифрованные отсчеты речи) с помощью специализированного программного обеспечения вводятся через звуковую карту, преобразуются и передаются нарушителю через открытую вычислительную сеть в удобном виде;
- данные сохраняются локально на жестком диске ПЭВМ и в дальнейшем нарушитель их забирает;
- данные преобразуются и передаются нарушителю по локальному интерфейсу ПЭВМ, например через USB-порт.

Атака А2 похожа на первую часть А1, за исключением того, что в роли подслушивающего устройства выступает ПЭВМ, а вместо радиоканала используется вычислительная сеть.

Обнаружить рассмотренную атаку возможно, поскольку она является активной. Если происходит запись речи и передача ее через сеть, можно выявить подозрительные пакеты с помощью анализатора трафика. Предотвратить атаку можно запретом генерации трафика в сети несанкционированными программами, что реализуется использованием брандмауэров.

А3. Получение речевой информации внедрением закладок в модули ввода/вывода звука DirectSound ОС Windows

Целью атаки является доступ к пакетам речевой информации во время ввода/вывода звука средствами DirectSound. Для реализации этой атаки нарушитель должен подменить одну из библиотек DirectSound, выполняющую ввод/вывод звука во время функционирования системы IP-телефонии. Функции DirectSound реализуют библиотеки dsound.dll и dsound3d.dll, которые представляют собой исполняемые файлы и находятся в системной папке \Windows\system32.

Возможны следующие варианты передачи перехваченной информации:

- данные во время ввода/вывода звука при функционировании системы IP-телефонии из модулей DirectSound с закладкой преобразуются и передаются нарушителю через открытую вычислительную сеть в удобном виде;
- данные сохраняются локально на жестком диске ПЭВМ (в дальнейшем нарушитель их забирает);
- данные преобразуются и передаются нарушителю по локальному интерфейсу ПЭВМ, например через USB-порт.

На практике обнаружить или предотвратить эту атаку возможно лишь при одновременном использовании тех же технических, организационных и инженерно-технических методов, что и для атаки А2.

А4. Получение речевой информации внедрением закладок в модули компрессии/декомпрессии ОС Windows

Целью данной атаки является доступ к пакетам речевой информации, когда они достигли этапа компрессии/декомпрессии звука (ACM ОС Windows). Для реализации этой атаки нарушитель должен подменить одну из библиотек, выполняющую компрессию/декомпрессию звука. Аудиокодеки представляют собой исполняемые файлы с расширением *.acm и находятся в системной папке \Windows\system32. Кроме того, аудиокодеки с троянскими вставками могут устанавливаться штатным образом для увеличения возможностей менеджера сжатия звука (ACM).

Возможны следующие варианты передачи перехваченной информации:

- данные во время работы системы IP-телефонии и вызова аудиокодека с закладкой преобразуются и передаются нарушителю через открытую вычислительную сеть в удобном виде;
- данные сохраняются локально на жестком диске ПЭВМ и в дальнейшем нарушитель их забирает;
- данные преобразуются и передаются нарушителю по локальному интерфейсу ПЭВМ, например, через USB-порт.

На практике обнаружить или предотвратить эту атаку возможно лишь при одновременном использовании тех же технических, организационных и инженерно-технических методов, что и для атаки А2.

А5. Получение ключа и/или речевой информации внедрением закладок в программные модули защиты от несанкционированного доступа ОС Windows

Целью атаки является доступ к пакетам речевой информации, когда они достигли этапа преобразований данных для защиты от несанкционированного доступа ОС Windows (атака А5.1). Для

реализации этой атаки нарушитель должен подменить одну из библиотек, выполняющую защиту данных от несанкционированного доступа. Библиотеки для защиты от несанкционированного доступа ОС Windows представляют собой исполняемые файлы с расширением *.dll и находятся в системной папке \Windows\system32. В ОС Windows для загрузки библиотеки с функциями защиты от несанкционированного доступа она должна быть подписана цифровой подписью компании Microsoft. Проверка подписи осуществляется библиотекой advapi32.dll. Для проведения атаки необходимо заменить advapi32.dll на такую же библиотеку из набора разработчика библиотек защиты от НСД CSPDK (Criptographic Service Provider Development Kit), который можно найти на сайте компании Microsoft [11]. В этом случае тестовую цифровую подпись для библиотеки с закладкой можно сгенерировать утилитой CspSign.exe, которая также входит в состав CSPDK. Замену advapi32.dll можно выполнить несколькими способами. Можно загрузиться в другой операционной системе и заменить advapi32.dll, либо загрузиться в безопасном режиме и заменить advapi32.dll (в обычном режиме подсистема восстановления файлов ОС Windows 2000/XP/2003 после перезагрузки восстановит замененный файл), для ОС Windows 98 в обычном режиме достаточно просто заменить advapi32.dll. Реализация атаки заметно упрощается, если в компании Microsoft удастся получить штатную цифровую подпись для библиотеки с закладкой. В этом случае advapi32.dll заменять не нужно.

В случае установки модуля для защиты от НСД с закладкой нарушитель кроме речевых данных может получить информацию о ключе и режиме шифрования (назовем эту атаку А5.2). Во время работы системы IP-телефонии и вызова модуля с закладкой полученная информация может передаваться нарушителю через открытую вычислительную сеть в удобном виде либо сохраняться локально на жестком диске ПЭВМ, и в дальнейшем нарушитель ее забирает по локальному интерфейсу ПЭВМ, например через USB-порт.

На практике обнаружить или предотвратить эту атаку возможно лишь при одновременном использовании тех же технических, организационных и инженерно-технических методов, что и для атаки А2. Кроме того, при использовании ОС Windows 2000/XP/2003 необходимо ограничить загрузку альтернативных операционных систем, чтобы предотвратить замену библиотеки advapi32.dll, осуществляющей проверку цифровой подписи.

А6. Подбор пароля программой типа "Троянский конь"

Целью атаки является получение паролей пользователей системы IP-телефонии во время их ввода на локальном компьютере.

Для перехвата паролей пользователей можно использовать постоянно работающие программы (процессы), фиксирующие нажатия клавиш на клавиатуре. Атака сводится к установке соответствующего программного обеспечения.

В случае успешной установки в операционную систему такого программного обеспечения нарушитель сможет получить протокол всех нажатий клавиш и на основе этой информации с минимальными вычислительными затратами определить пароли пользователей системы IP-телефонии. Во время работы системы IP-телефонии полученная информация может передаваться нарушителю через открытую вычислительную сеть в удобном виде либо сохраняться локально на жестком диске ПЭВМ, и в дальнейшем она передается нарушителю по локальному интерфейсу ПЭВМ, например через USB-порт.

На практике обнаружить или предотвратить эту атаку возможно лишь при одновременном использовании тех же технических, организационных и инженерно-технических методов, что и для атаки А2.

А7. Перехват пакетов с данными

Целью этой атаки является определение пароля легального пользователя системы IP-телефонии для получения доступа к информации от его имени. В отличие от предыдущих локальных атак данная атака является сетевой.

В рассматриваемой системе IP-телефонии пароли пользователей не хранятся в системе. Кроме того, для увеличения стойкости пароля генерация ключа на основе пароля реализуется согласно рекомендациям исследовательской лаборатории RSA Laboratories [12]. Текстовый пароль относительно небольшой произвольной длины с помощью хеш-функции (SHA-1) отображается в вектор фиксированной длины. Для улучшения качества этой схемы введен модификатор ключа (случайное число длиной 12 байт), который замешивается в функцию хеш-преобразования вместе с паролем. Это позволяет получить разный ключ при одинаковых паролях и осложняет атаку перебором паролей. Кроме того, введен счетчик итераций J — число раз, которое должна повториться функция преобразования, участвующая в генерации ключа. Например, 10 000 итераций почти не скажутся на скорости вычисления ключа обычной ПЭВМ для легальных пользователей,

однако для нарушителя при переборе паролей потребуются огромные вычислительные мощности.

Для шифрования/дешифрования потока речевых пакетов пользователи на передающей/приемной стороне вводят одинаковый пароль для одинаковых режимов и алгоритмов шифрования. Зашифрованные пакеты попадают в открытую вычислительную сеть и могут быть перехвачены нарушителем. Атака представляет собой проверку некоторого, как правило, значительного числа возможных вариантов пароля. Наиболее общий алгоритм атаки подразумевает полный последовательный перебор всех возможных вариантов пароля.

Время T, необходимое для опробования одного пароля, складывается из времени, необходимого для создания ключа, и времени, необходимого для его опробования:

$$T = 1/S_{\text{к}} + J/S_J \qquad (1)$$

где $S_{\text{к}}$ — скорость опробования одного ключа (единиц/с); S_J — скорость генерации одного ключа на основе пароля (единиц/с); J — число раз, которое должна повториться функция преобразования, участвующая в генерации ключа на основе пароля.

Тогда в модели полного последовательного перебора паролей вероятность успеха атаки

$$p_{A7.1}(t) = \begin{cases} \dfrac{t}{N_{\text{п}}(1/S_{\text{к}} + J/S_J)}, \\ \text{если } \dfrac{t}{N_{\text{п}}(1/S_{\text{к}} + J/S_J)} <= 1; \\ 1, \text{ если } \dfrac{t}{N_{\text{п}}(1/S_{\text{к}} + J/S_J)} > 1, \end{cases} \qquad (2)$$

где $N_{\text{п}}$ — число вариантов пароля (емкость множества паролей).

Существует модификация данной атаки, когда ищется не пароль, а непосредственно ключ. Вероятность успеха модифицированной атаки на шифроданные в системе IP-телефонии также зависит от времени, скорости опробования одного ключа и от числа вариантов ключа (модель полного последовательного перебора ключей):

$$p_{A7.2}(t) = \begin{cases} \dfrac{t}{N_{\text{к}}/S_{\text{к}}}, \text{ если } \dfrac{t}{N_{\text{к}}/S_{\text{к}}} <= 1; \\ 1, \text{ если } \dfrac{t}{N_{\text{к}}/S_{\text{к}}} > 1, \end{cases} \qquad (3)$$

где $N_{\text{к}}$ — число вариантов ключа (емкость множества ключей); $S_{\text{к}}$ — скорость опробования одного ключа (единиц/с).

Очевидно, что второй вариант атаки является взломом алгоритма, используемого для шифрования/дешифрования данных с помощью метода "грубая сила", и ее трудность определяется длиной ключа шифрующего алгоритма. Определим параметры модели полного последовательного перебора паролей, когда трудоемкость атаки по ней сопоставима или больше трудоемкости атаки в модели полного последовательного перебора ключей, т. е. когда $p_{A7.1}(t) <= p_{\text{A7.2}}(t)$:

$$\frac{t}{N_{\text{п}}(1/S_{\text{к}} + J/S_J)} <= \frac{t}{N_{\text{к}}/S_{\text{к}}},$$

$$\text{тогда } N_{\text{п}} >= N_{\text{к}}/\left(1 + \frac{S_{\text{к}}J}{S_J}\right). \qquad (4)$$

Примем следующее обозначение: $S_{\text{к}} = MS_J$. Взяв логарифм по основанию 2 от обеих частей неравенства и учитывая, что основание логарифма больше 1, получим:

$$\log_2 N_{\text{п}} >= \log_2 N_{\text{к}} - \log_2(1 + JM). \qquad (5)$$

Обозначим $\log_2 N_{\text{п}} = П$ — длина пароля в битах; $\log_2 N_{\text{к}} = K$ — длина ключа в битах, тогда

$$П >= K - \log_2(1 + JM). \qquad (6)$$

На рисунке показан расчетный график эффективности генерации ключей из паролей, т. е. $(K - П)/K \; \forall \; K \in [56, 128, 256]$ при $M = 1$. Из рисунка следует, что наибольшая эффективность от применения алгоритма, получающего ключи из паролей, может быть получена для алгоритмов шифрования с относительно небольшой длиной ключа, например 56 бит.

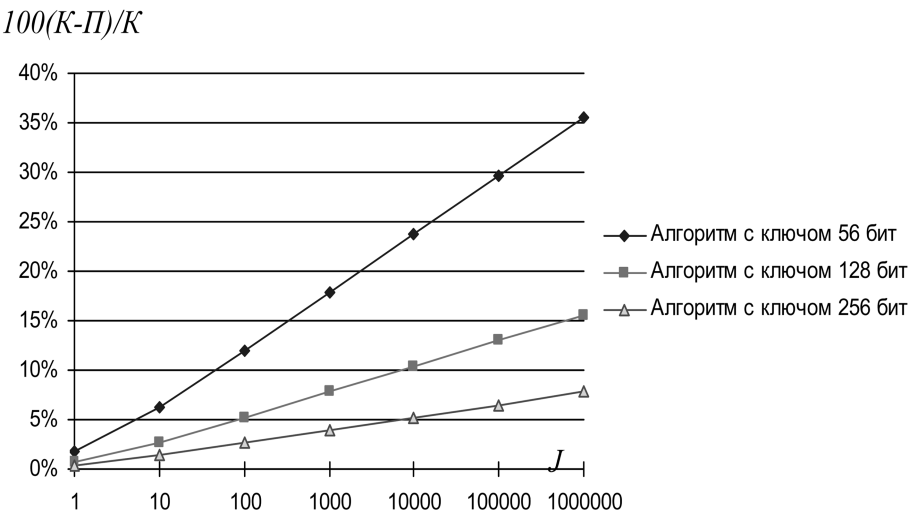
На практике обнаружить или предотвратить атаку подбором (перебором) пароля (ключа) бывает чрезвычайно тяжело, поскольку она, как правило, проводится на компьютере (компьютерах) нарушителя. Для снижения риска ее успешной реализации необходимо обеспечить выбор надежного алгоритма шифрования, длинного и качественного пароля, а также ограничить срок действия пароля.

A8. Навязывание пользователю ложного сообщения

Целью этой атаки является имитация речи легального пользователя системы IP-телефонии. Так же, как и предыдущая атака, она является сетевой.

При успешной реализации предыдущей атаки нарушитель получает пароль (ключ), поэтому у него появляется возможность навязать пользователю системы зашифрованные пакеты с ложной информацией, при этом пользователь может об этом даже не догадываться (атака A8.1). Однако существует другой, более простой для нарушителя, способ осуществлять дезинформацию.

Зашифрованные пакеты попадают в открытую вычислительную сеть и могут быть перехвачены



Эффективность генерации ключей из паролей в зависимости от *J* при *M* = 1

нарушителем. В некоторых режимах шифрования, например, в шифре простой замены каждый одинаковый блок исходного текста приводит к появлению одинакового блока шифрованного текста. Исходным текстом можно легко манипулировать путем удаления, повторения или перестановки блоков. Таким образом, если нарушитель находится между абонентами, использующими данный режим шифрования, то у него появляется возможность имитации речи каждого из абонентов, даже если он не обладает ключом для шифрования/дешифрования.

Атака А8.2 представляет собой вставку в поток зашифрованных пакетов зашифрованных блоков в порядке и виде, нужным нарушителю.

На практике предотвратить атаку навязывания ложного сообщения можно, следуя рекомендациям по противодействию предыдущей атаки, а также используя режимы шифрования, в которых невозможна манипуляция блоками шифроданных или путем добавления к каждому пакету с открытой информацией некоторых данных, характеризующих уникальность пакета, например, время и/или порядковый номер пакета. Если не используются указанные выше рекомендации, обнаружить указанную атаку очень сложно, так как пакеты с шифроданными являются либо подлинными, либо зашифрованными с помощью настоящего ключа.

Частным случаем данной атаки являются естественные помехи. Без злого умысла (нарушитель — природа) шифрованный текст может быть удален, повторен, изменен или переставлен. При этом даже минимальная защита структуры данных с помощью некоторой избыточности позволит с большой вероятностью (зависит от способа внесения избыточности) обнаружить и (возможно) исправить искажения, произошедшие в шифротексте.

А9. Отказ в обслуживании

Целью атаки является сбой работы системы IP-телефонии по сети и создание ситуации "отказ в обслуживании", когда система не сможет обмениваться информацией с внешним миром. Так же, как и предыдущая атака, она является сетевой.

При работе системы IP-телефонии для приема и передачи данных используются порты. Например, для приема основного потока данных может использоваться порт № 2001, а для приема служебной информации порт № 2002. Атака реализуется несколькими способами: создание направленного шторма запросов на указанные выше порты, реализация ошибок сетевых протоколов путем передачи в адрес атакуемой системы некорректных пакетов. В первом случае возможна ситуация, когда перегруженный входящими пакетами удаленный компьютер не сможет осуществлять нормальный прием/передачу данных, во втором случае возможны переполнения в механизмах обработки запросов.

На практике предотвратить и блокировать атаку "отказ в обслуживании" сложно, а обнаружить легко. Существует множество шагов, которые можно предпринять для смягчения эффекта от таких атак. Если атака уже началась, то нужно определить, откуда приходит злонамеренный трафик, и связаться с источником (с провайдерами или другими лицами, причастными к его генерации). Если цель — единичная машина, простая смена IP-адреса прекратит атаку. Новый адрес можно дать только наиболее важным внешним пользователям. Есть шанс, что могут помочь некоторые методики фильтрации. Тщательное исследование перехваченных пакетов иногда дает характерные следы (сигнатуры), на основе которых можно создать ACL (access control lists — списки контроля доступа) маршрутизатора или правила брандмауэра. И последней возможностью является установка дополнительного оборудования и увеличение пропускной способности во время атаки и ожидания ее прекращения.

Наибольшая сложность при защите от атак "отказ в обслуживании" состоит в поддельных IP-адресах атакующих машин. Эта проблема может быть решена с использованием методики, называемой "исходящая фильтрация". Если сетевые администраторы установят такую фильтрацию, поддельные пакеты станут почти невозможными, что многократно сократит процесс идентификации в исследовании этих атак. К со-

жалению, во многих сетях эти фильтры отключены, и поддельные пакеты беспрепятственно проходят.

A10. Подмена программного обеспечения IP-телефонии

Подмена программного обеспечения IP-телефонии позволяет нарушителю реализовать атаки, аналогичные атакам А3—А6, А8, А9 в связи с тем, что управляющая программа имеет доступ ко всем подмодулям системы (в частности, могут быть получены отсчеты "сырого" звука, отсчеты сжатого и зашифрованного звуков, пароль и ключ шифрования, также может быть осуществлены атаки "навязывания ложного сообщения" и "отказ в обслуживании"). Назовем получение нарушителем пакетов со звуком — атакой А10.1, получение ключа или пароля — атакой А10.2, атаку типа "отказ в обслуживании" — А10.3.

На практике обнаружить или предотвратить эту атаку возможно лишь при одновременном использовании тех же технических, организационных и инженерно-технических методов что и для атаки А2.

Рассмотренные отдельные атаки не всегда могут привести нарушителя к положительному для него конечному результату, поэтому активность нарушителя в общем случае может складываться из некоторых последовательностей атак, зависящих как от целей нарушителя, так и от его возможностей.

Возможные варианты последовательностей применения атак составляют стратегии действий нарушителя. В каждом конкретном атакующем воздействии может быть реализована только одна из возможных стратегий.

Стратегии действий нарушителя, направленные на НСД к информации или на нарушение работы системы передачи речи.

Информация может быть получена в обход ПЭВМ (А1), в обход системы IP-телефонии (А2). Особенностью этих видов атак является то, что они могут быть осуществлены вообще без инсталляции и без функционирования системы IP-телефонии.

Суть атак А3, А4 и А5 заключается в изменении работы стандартных компонентов операционной системы Windows, используемых в предложенной системе IP-телефонии (модулей DirectSound, аудиокодеков и модулей для защиты данных от несанкционированного доступа). При атаке А3 нарушитель может получить доступ к отсчетам оцифрованной речи. При атаке А4 нарушитель может получить доступ к отсчетам оцифрованной речи как до, так и после ее компрессии или декомпрессии. При атаке А5.1 нарушитель может получить доступ к отсчетам оцифрованной речи после ее компрессии или декомпрессии, в результате реализации модификации этой атаки А5.2 нарушитель может получить информацию об используемом для защиты от НСД ключе.

В случае получения нарушителем информации о ключе возможно два варианта действий нарушителя: пассивный перехват данных, проходящих через открытые сети передачи IP-пакетов, их дешифрование и прослушивание либо в реальном времени, либо в записи. Второй вариант предполагает активное навязывание ложных сообщений путем имитации речи одного или нескольких абонентов (дезинформация) — первый вариант атаки А8.

Атаки А6 и А7 преследуют цель получения информации о пароле и о ключе для криптопреобразований. В случае успешной реализации атаки А6 нарушитель получает пароль, что при открытом алгоритме генерации ключа из пароля означает легкое получение ключа. Результатом успешной реализации атаки А7 также является информация о ключе. Реализация указанных атак приводит отношения нарушитель — система защищенной IP-телефонии в незащищенное состояние.

Второй вариант атаки А8 заключается в манипуляции зашифрованными пакетами данных без информации об используемом ключе в некоторых режимах шифрования данных.

Атака А9 — "отказ в обслуживании" предназначена для нарушения нормального функционирования системы IP-телефонии путем активного воздействия на ключевые элементы маршрута передачи пакетов с данными: порты приема данных на приемной или передающей стороне, локальные коммутаторы, граничные и магистральные маршрутизаторы.

Последняя атака А10 "подмена программного обеспечения IP-телефонии" позволяет нарушителю реализовать атаки, аналогичные атакам А3—А6, А8, А9 в связи с тем, что управляющая программа имеет доступ ко всем подмодулям системы.

На основе анализа потенциальных угроз авторами предложены методические рекомендации, позволяющие снизить риски несанкционированного доступа к защищаемой речевой информации и нарушения нормального функционирования системы IP-телефонии, работающей в защищенном режиме.

Список литературы

1. **Гостехкомиссия** России. Руководящий документ. Концепция защиты средств вычислительной техники от несанкционированного доступа к информации. М., 1992.
2. **Гостехкомиссия** России. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации. М., 1992.
3. **Домарев В. В.** Безопасность информационных технологий. Методология создания систем защиты. Киев: ДиаСофт, 2001. 688 с.
4. **Лукацкий А. В.** Обнаружение атак. СПб.: BHV-Санкт-Петербург, 2003. 596 с.
5. **Лукацкий А. В.** IP-телефония: угрозы, атаки и способы их отражения. // Мир связи. Connect! 2002. № 8. С. 35—51.

6. **Максимов Ю. Н.** Технические методы и средства защиты информации / Ю. Н. Максимов, В. Г. Сонников, В. Г. Петров и др. СПб.: ООО "Издательство Полигон", 2000. 320 с.

7. **Нопин С. В.** Свидетельство об официальной регистрации программ для ЭВМ. Система защищенной IP-телефонной связи. № 2006612351 — Программы для ЭВМ... (офиц. бюл.). 2006. № 4 (57). С. 20.

8. **Нопин С. В., Шахов В. Г.** Возможности использования встроенных звуковых кодеков операционной системы (ОС) Windows в системах IP-телефонии // Омский научный вестник. 2006. № 1 (34). С. 155—157.

9. **Нопин С. В., Шахов В. Г.** Разработка защищенных от несанкционированного доступа систем IP-телефонии на основе операционной системы Windows // Омский научный вестник. 2006. № 9 (46). С. 137—142.

10. **Прохоров П. В.** Разработка методов оценивания качества и эффективности защиты операционной системы Windows NT. Дис. ... канд. техн. наук. Омск: 2002. 270 с.

11. **Microsoft.** [Electronic resource]. Redmond, 2007. Режим доступа: http://www.microsoft.com, свободный. — Загл. с экрана.

12. **RFC 2898.** PKCS #5: Password-Based Cryptography Standard, Version 2.0 [Electronic resource] / B. Kaliski — Sterling, 2000.

УДК 004.738.5

А. А. Печников, канд. физ.-мат. наук,
ст. науч. сотрудник,
Институт прикладных
математических исследований
Карельского научного центра РАН,
г. Петрозаводск

Вебометрические исследования Web-сайтов университетов России

Проведено исследование сайтов классических университетов Российской Федерации, основанное на измерениях с помощью поисковых машин таких индикаторов сайта как число индексируемых страниц и число ссылок на него. Предложен критерий ранжирования сайтов по их узнаваемости в Интернете и на его основе проведено ранжирование официальных университетских сайтов.

Ключевые слова: *Web, Интернет, вебометрика, индикаторы, поисковые машины, сайт, ранжирование.*

Введение

В работе [1] вебометрика (*webometrics*) определяется как научное направление, посвященное исследованиям количественных аспектов конструирования и использования информационных ресурсов, структур и технологий применительно к Web.

Проект "*Webometrics Ranking of World Universities*" испанской исследовательской группы *Cybermetrics Lab* [2] посвящен вебометрическим исследованиям сайтов вузов и научно-исследовательских институтов мира. Рейтинг публикуется на сайте проекта [3] с обновлением один раз в полгода. По результатам этих исследований сайты российских вузов занимают достаточно скромные позиции в мировом рейтинге. Например, официальный сайт Московского государственного университета находится на 150-м месте, а большинство сайтов российских вузов занимают места где-то в последней тысяче (если вообще попали в Top 4000 Universities).

Проведенное авторами самостоятельное исследование сайтов классических университетов Российской Федерации, использующее подходы *Cybermetrics Lab*, позволило выявить ряд интересных моментов, касающихся как способов измерения основных индикаторов Интернет-ресурсов, так и подходов к ранжированию сайтов. Предложен собственный подход к ранжированию сайтов и с его использованием определен рейтинг классических университетов России.

1. Подходы и методики *Cybermetrics Lab*

Для построения схемы ранжирования сайтов *Cybermetrics Lab* [3] предлагает использовать следующие четыре индикатора:

- число уникальных гипертекстовых ссылок с других ресурсов (*V — Visibility*, цитируемость);
- общее число страниц сайта (*S — Size*, размер);
- количество полнотекстовых файлов, под которыми понимаются файлы с расширениями *.pdf, *.ps, *.doc, *.xls, *.ppt и *.rtf (*R — Rich files*);
- число ссылок на научные статьи, размещенные на сайте (*Sc — Scholar*, "научность сайта").

Индикатор *S* измеряется с использованием поисковых машин *Google*, *Yahoo*, *Live Search and Exalead*, индикатор *V — Yahoo Search, Live Search* и *Exalead*, индикатор *R — Google*, а индикатор *Sc — Google Scholar*. Итоговые значения для *S* определяются как суммы результатов четырех замеров с отбрасыванием минимальных и максимальных значений, а для *V* — просто как суммы трех замеров.

По каждому индикатору выполняется ранжирование сайтов по убыванию их значений. Для обозначения ранга по заданному индикатору используются обозначения *RankV*, *RankS*, *RankR* и *RankSc* соответственно (наивысший ранг равен 1). Интегральный показатель, называемый "вебометрическим рангом" (*WR — Webometrics Rank*), в [3] вычисляется как $WR(position) = 4 \times RankV + 2 \times RankS + RankR + RankSc$.

Значения коэффициентов позволяют сделать вывод о том, что наибольшая значимость придается числу размещенных на сайте полнотекстовых файлов и числу ссылок на научные статьи, найденные *Google Scholar*. Следующим по значимости является число страниц на сайте, а затем — число ссылок на сайт с других сайтов.

Cybermetrics Lab отмечает серьезные проблемы с точным определением того, что понимать под "единицей анализа" (наличие у одной организации нескольких доменных областей, наличие сайтов подразделений, имеющих адреса, не ассоциируемые с адресом основного сайта и т. д.). Однако в качестве "единиц анализа" в работе [3] выбраны адреса официальных сайтов университетов и институтов, а не множества адресов всех Интернет-ре-

сурсов данного учреждения, что, очевидно, сужает постановку общей задачи.

2. Задачи и объект исследования

Основными задачами нашего исследования являются:

- анализ подходов и методик *Cybermetrics Lab*;
- доработка (и переработка) подходов и методик к измерениям и ранжированию с учетом проведенного анализа;
- сравнительный анализ и ранжирование официальных Web-сайтов классических университетов России на основе собственного подхода.

Интернет-ресурсы университета в целом представляют собой сложный информационный комплекс, в котором, как правило, имеется официальный сайт вуза, а также сайты факультетов, институтов, кафедр и административных подразделений, библиотечный сайт, страницы преподавателей, и т. д. Некоторые подразделения вуза (в особенности это свойственно подразделениям, профессионально связанным с информационными технологиями) имеют Интернет-ресурсы, зарегистрированные под именами, не содержащими доменного имени основного сайта вуза. В этом случае только содержательный анализ ресурса может дать ответ на вопрос, является ли этот ресурс частью Интернет-ресурсов университета, что само по себе является непростой задачей. Поэтому, заведомо сужая область исследований, в качестве так называемых "единиц анализа" для нашего исследования выбраны доменные имена главных страниц официальных сайтов университетов, не являющихся директориями некоторого домена.

Последнее условие связано с тем, что не все поисковые машины могут проводить поиск в директориях и поддиректориях некоторого домена. По этой причине из списка сайтов классических университетов был вынужденно удален сайт Башкирского государственного университета, имеющий адрес *www.bashedu.ru/firstbgu_r.htm* (как раздел портала образовательных организаций Башкортостана *www.bashedu.ru*).

В качестве источников информации взяты официальные сайты Федерального агентства по образованию (*http://www.ed.gov.ru*) и Федерального портала "Российское образование" (*http://www.edu.ru*). Если возникали сомнения относительно основного адреса официального сайта университета, то использовался адрес, указанный в официальных источниках.

По данным портала "Российское образование" в мониторинге университетов за 2006 год указан 81 классический государственный университет. Исключив из этого списка Башкирский университет (по указанным выше причинам) и Карачаево-Черкесский университет (по причине отсутствия сайта), в качестве объекта исследований получаем множество из 79 официальных сайтов классических государственных университетов России.

Ограничиваясь только официальными сайтами университетов, мы, на самом деле, уходим от постановки задачи, сформулированной в [3], — вместо "Вебометрического ранга университетов" имеем "Вебометрический ранг официальных сайтов университетов", что, по-видимому, ближе к действительности и для исследований *Cybermetrics Lab*.

3. Об индикаторах сайтов и применимости поисковых машин в качестве "измерительных приборов"

Для более точного понимания того, что характеризуют индикаторы *S*, *V*, *R* и *Sc*, разобьем все множество пользователей Интернета на два подмножества по отношению к фиксированному сайту. К первому подмножеству отнесем всех пользователей, которые заходят на сайт по прямому набору его адреса в строке браузера (или через меню Избранное, что, в конце концов, одно и то же). Можно сказать, что участники этого подмножества "знают" данный сайт (пока возьмем слово в кавычки). Второе подмножество — это все остальные пользователи. Некоторые из них попадают на сайт через ссылки, найденные различными поисковыми машинами по задаваемой строке поиска, а другие — через гиперссылки, выставленные на данный сайт на других сайтах. Участники второго подмножества в силу того, что они не относятся к первому подмножеству, не знают о сайте, но имеют потенциальную возможность узнать о нем благодаря описанным способам.

Теперь вернемся к индикаторам. На самом деле *S* — это не общее число страниц сайта, как об этом сказано в [3], а число страниц сайта, проиндексированных конкретной поисковой машиной. Для примера: Яндекс практически "не видит" страниц на сайтах финских университетов, а бывшая *Teoma* (нынешний *Ask*) — обнаруживает лишь 60 страниц на сайте МГУ. Однако при проведении замеров обнаруживается, что и разные российские поисковые системы индексируют весьма различное число страниц на одном и том же сайте. Механизмы индексации страниц являются секретной особенностью поисковых машин и, по-видимому, существенно влияют на контекстный поиск по поисковым фразам. Единственное, что можно осторожно предположить, так это то, что чем больше страниц сайта проиндексировано, тем вероятнее возможность попадания на данный сайт через эту поисковую машину.

Более просто пояснить суть индикатора *V*. Поскольку это число ссылок на данный сайт, найденное поисковой машиной на других сайтах, то оно, скорее всего, зависит от числа всех страниц Интернета, охватываемых поисковой машиной (у Яндекса на 26 декабря 2007 года — "Поиск по 3 029 570 128 веб-страницам"). Следовательно, чем больше ссылок на сайт обнаруживается в Интернете, тем вероятнее возможность попадания на него с других сайтов.

Индикатор *R* (по аналогии с *S*) — это число полнотекстовых страниц сайта, проиндексированных конкретной поисковой машиной. По-существу, это частный случай индикатора *S*. Понятно, что чем больше *R*, тем вероятнее возможность попадания на сайт через поисковую машину с использованием опций расширенного поиска.

Суть индикатора *Sc* можно пояснить следующим образом: чем больше ссылок на полнотекстовые страницы сайта, содержащие научные публикации, находит *Google Scholar*, тем вероятнее возможность попадания на сайт. Скажем сразу же, что *Google Scholar* является бета-версией и в настоящее время мало пригоден для обнаружения ссылок на российские сайты. В России поисковой системой, аналогичной *Google Scholar*, является поисковая машина проекта *Scholar.Ru* (также бета-версия, индексирующая сегодня весьма небольшое число публикаций в Интернете).

Сделаем некоторые выводы из сказанного. Множество пользователей Интернета по отношению к данному сайту можно разделить на два подмножества. Первое — это пользователи, знающие данный сайт и обращающиеся к нему по адресу, а второе — это все остальные пользователи, которые имеют потенциальную возможность узнать о сайте различными способами. Индикаторы характеризуют возможность попадания на сайт через один из четырех таких способов: *S* — обнаружение с помощью поисковой машины и переход на сайт, *V* — переход с другого сайта по ссылке, *R* — обнаружение с помощью поисковой машины полнотекстового файла данного сайта и переход на сайт и *Sc* — переход через ссылку, найденную поисковой машиной для научных публикаций, причем чем больше значение индикатора, тем вероятнее такая возможность. Следовательно, критерий, основанный на *S*, *V*, *R* и *Sc*, характеризует возможность попадания пользователей из второго подмножества на заданный сайт. Можно сказать, что этот критерий характеризует узнаваемость сайта в Интернете, поскольку, если по отношению к пользователям первого подмножества применим термин "знание сайта", то по отношению к пользователям второго подмножества более приемлемым кажется термин "узнавание".

Теперь можно дать еще более точную формулировку для нашего случая задачи ранжирования сайтов по критериям, использующим индикаторы *S*, *V*, *R* и *Sc*: "Вебометрический ранг узнаваемости официальных сайтов университетов".

Необходимость использования *R* и *Sc* в [3] обосновывается принципами открытого доступа, когда сеть Интернет рассматривается в первую очередь как средство функционального объединения глобальной базы научных знаний [4]. Однако вряд ли официальный сайт университета может рассматриваться в качестве такого средства. Скорее это относится к сайтам научных подразделений и электронных библиотек вуза. Поэтому сомнительно включать в оценки рейтинга официального сайта вуза такой индикатор, как *Sc* — "научность сайта". Конечно, он имел бы право на существование в случае, если под единицей анализа понималась бы вся совокупность интернет-ресурсов университета, но из методологии *Cybermetrics Lab* этого не следует. То же самое можно сказать и по поводу индикатора *R*. Как уже отмечалось, он является частным случаем *S* и фактически измеряется при измерении *S*.

4. Методика измерений

В качестве средств измерения в нашем случае были взяты три наиболее распространенные в России поисковые машины — Яндекс, Google.ru и Рамблер. По данным многочисленных источников, в сумме их применяют примерно 90 % пользователей Рунета. Все замеры проводятся по доменному имени официального сайта университета — DNOS (без указания на протокол *http://*, например, *new.vpti.vladimir.ru, www.bsu.edu.ru, www.msu.ru*).

Измерения в Яндексе:

S — измеряется на странице *http://webmaster.yandex.ru/check.xml*, в строке поиска вводится DNOS. Результат поиска определяется как число найденных страниц (во фразе Яндекса: "... страниц не менее nnnn", где nnnn — целое число, принимаемое в качестве значения индикатора).

V — измеряется на странице *http:/www.yandex.ru/,* в строке поиска вводится "DNOS" (в кавычках). Результат поиска определяется как число найденных сайтов ("... сайтов не менее nnnn").

Измерения в Рамблере:

S — измеряется на странице расширенного поиска *http://www.rambler.ru/cgi-bin/advanced.cgi?set = www* с установкой следующих основных опций: слова запроса — отсутствуют, язык — любой, формат — любой, искать документы только на следующих сайтах: DNOS. Результат поиска определяется как число найденных документов ("найдено документов: nnnn").

V — измеряется на странице расширенного поиска с указанием в строке поиска DNOS установкой следующих основных опций: поиск по тексту — гиперссылок, язык — любой, формат — любой, искать документы только на следующих сайтах — нет. Результат поиска определяется как число найденных документов ("Вы искали: DNOS, найдено ... документов: nnnn").

Измерения в Google.ru:

S — измеряется на главной странице *http://www.google.ru*, в строке поиска вводится site:DNOS.

Результат поиска определяется как число найденных страниц ("... приблизительно nnnn с DNOS").

V — измеряется на главной странице, в строке поиска вводится link:DNOS. Результат поиска определяется как число найденных страниц ("... приблизительно nnnn ссылающихся на DNOS").

Измерения тематического индекса цитирования Яндекса:

Для сайтов университетов был измерен так называемый тематический индекс цитирования Яндекса (тИЦ), о котором говорится, что "...тИЦ определяет «авторитетность» интернет-ресурсов с учетом качественной характеристики ссылок на них с других сайтов" [5].

Значения тИЦ измеряются на странице каталога "Высшее образование" *http://yaca.yandex.ru/yca/cat/Science/Higher_Education/*, в строке поиска вводится DNOS. Результат поиска определяется в стоке "...Цитируемость: nnnn".

5. Измерения, результаты и анализ

Измерения индикаторов *S*, *V* и тИЦ проводились в период с 5 ноября по 30 декабря 2007 года ежемесячно. В проводимом исследовании не ставилась задача постоянного мониторинга индикаторов и тИЦ, поэтому по каждому сайту были проведены по три замера и в качестве результирующего значения принято среднее арифметическое.

Задача мониторинга индикаторов сама по себе представляет достаточно большой интерес хотя бы с целью определения того, на каком этапе жизненного цикла находится сайт. Конечно, удержаться от мониторинга индикаторов полностью было невозможно, и поэтому вкратце некоторые результаты можно передать одной фразой: индикаторы сайтов крупных университетов достаточно стабильны во времени. Вместе с тем, следует четко понимать, что измеренное значение индикатора — это значение, полученное с использованием заданной поисковой машины в конкретный момент времени при четко определенном виде поискового запроса.

Ввиду экономии места приведем выборку из таблицы измерений (табл. 1), содержащую информацию по сайтам, вошедшим в лидирующую группу, и другим сайтам,

Таблица 1

Выборка из полной таблицы измерений

№	Название университета	DNOS	*Sy*	*Vy*	*Sr*	*Vr*	*Sg*	*Vg*	*CY*
1	Адыгейский ГУ	www.adygnet.ru	28 760	238	1 651	396	1 360	90	1 100
2	Алтайский ГУ	www.asu.ru	101 892	430	8 689	4 149	7 890	250	3 400
3	Амурский ГУ (Благовещенск)	www.amursu.ru	6 136	272	3 180	519	3 640	54	650
5	Белгородский ГУ	www.bsu.edu.ru	57 611	431	63 399	1 701	14 000	309	2 000
14	Дагестанский ГУ	www.dgu.ru	81 937	369	1 790	1 313	1 050	84	2 100
15	Дальневосточный ГУ (Владивосток)	www.dvgu.ru	73 013	848	10 890	1 676	37 800	855	4 800
16	Елецкий ГУ	www.elsu.ru	1 936	130	749	188	222	22	650
20	Кабардино-Балкарский ГУ	www.kbsu.ru	4 114	374	4	857	734	31	1 800
21	Казанский ГУ	www.ksu.ru	19	611	44 871	4 143	31 600	1 770	2 500
22	Калмыцкий ГУ (Элиста)	www.kalmsu.ru	202	114	220	144	148	8	700
35	Нижегородский ГУ	www.unn.ru	37 256	649	30 103	1 875	14 100	455	5 500
36	Новгородский ГУ	www.novsu.ru	4 266	128	28 146	1 096	80 700	996	2 200
37	Новосибирский ГУ	www.nsu.ru	176 788	1 407	53 728	5 452	70 400	2 510	9 800
38	Омский ГУ	www.omsu.ru	7 619	155	2 904	821	1 830	138	1 100
48	Санкт-Петербургский ГУ	www.spbu.ru	81 722	891	2 053	3 169	2 000	1 790	4 500
49	Саратовский ГУ	www.sgu.ru	95 642	940	17 743	2 063	27 700	845	3 400
50	Сахалинский ГУ	www.sakhgu.ru	461	57	248	232	321	22	80
51	Северо-Восточный ГУ (Магадан)	www.svgu.ru	738	6	385	5	818	1	10
61	Тольяттинский ГУ	www.tltsu.ru	9 665	158	0	328	980	11	350
62	Томский ГУ	www.tsu.ru	101 341	765	5 456	4 039	6 030	932	7 300
63	Тульский ГУ	tsu.tula.ru	11 754	676	14 417	546	10 100	47	650
76	Южно-Уральский ГУ (Челябинск)	susu.ac.ru	129 426	1 564	67 924	1 244	69 000	303	1 100
77	Южный федеральный университет (Ростов-на-Дону)	www.sfedu.ru	6 388	1 567	70	91	83	7	2 700

Обозначения: *Sy, Sr и Sg* — значения *S*, измеренные Яндексом, Рамблером и Google.ru; *Vy, Vr* и *Vg* — значения *V,* измеренные Яндексом, Рамблером и Google.ru; *CY* — значения тематического индекса цитирования.

входящим в их окружение с сохранением основной порядковой нумерации (см. следующий раздел "Критерии и результаты ранжирования"). Полные данные об измерениях и рейтингах приведены на ресурсе [6].

Проверка зависимостей между *Sy*, *Sr* и *Sg*, а также *Vy*, *Vr*, *Vg* и *CY* (естественно, по значениям, взятым из полной таблицы измерений) дала следующие значения коэффициентов корреляции ρ:

$$\rho(Sy, Sr) = 0{,}342,\ \rho(Sy,Sg) = 0{,}354,\ \rho(Sr,Sg) = 0{,}631;$$
$$\rho(Vy, Vr) = 0{,}641,$$
$$\rho(Vy, Vg) = 0{,}612,\ \rho(Vr, Vg) = 0{,}899.$$

Полученные результаты в целом подтверждают рассуждения из раздела 3 о том, что *S* зависит от поисковой машины и не может характеризовать реальный объем сайта, для которого проводятся измерения. И наоборот, *V* зависит от реального числа ссылок на сайт вне зависимости от поисковой машины, с помощью которой проводятся измерения.

(Отвлекаясь от последовательного изложения результатов исследования, скажем о том, что замеры *V* с помощью американской поисковой машины *Yahoo*! также показывают сильную корреляцию:

$$\rho(Vy, Vyah) = 0{,}574,\ \rho(Vg, Vyah) = 0{,}860,$$
$$\rho(Vr, Vyah) = 0{,}873.)$$

Проверка зависимостей между *Vy*, *Vr*, *Vg*, *Vyah* и *CY* показывает сильную зависимость тИЦ и числа внешних ссылок на сайт, фиксируемых любой поисковой машиной:

$$\rho(Vy, CY) = 0{,}727,\ \rho(Vg, CY) = 0{,}910,$$
$$\rho(Vr, CY) = 0{,}887\ \rho(Vyah, CY) = 0{,}871.$$

Поскольку значения индикаторов *S*, измеренных различными поисковыми машинами, взаимонезависимы и характеризуют возможность обнаружения сайта с помощью поисковой машины и переход на него, в качестве интегрального индикатора можно принять некоторое средневзвешенное значение по всем трем замерам. В качестве итоговых значений индикаторов *V* практически можно принять замеры любой поисковой машиной или тИЦ вследствие их сильной взаимной корреляции.

6. Критерий и результаты ранжирования

Опишем предлагаемый подход к определению вебометрического ранга узнаваемости официальных сайтов университетов в виде следующей последовательности шагов:

1. Для каждого сайта из множества сайтов университетов вычисляется *Sint* — интегральный показатель числа страниц как средневзвешенное значение трех значений индикаторов по формуле:

$$Sint_i = Py_i \times Sy_i + Pr_i \times Sr_i + Pg_i \times Sg_i,$$

где *i* — порядковый номер сайта, а *Py*, *Pr* и *Pg* — показатели популярности поисковых систем.

2. Проводится нормирование всех *Sint*:

$$Sint\ norm_i = \frac{Sint_i}{\sum\limits_i Sint_i}.$$

3. Проводится нормирование всех *CY*:

$$CYnorm_i = \frac{CY_i}{\sum\limits_i CY_i}.$$

4. Для каждого сайта вычисляется *WRR* — вебометрический ранг узнаваемости (*Webometrics Rank of Recognition*) по формуле

$$WRR_i = Ps \times Sint\ norm_i + Pv \times CYnorm_i,$$

где *Ps* и *Pv* — вероятности реализации того или иного способа попадания на сайт (через обнаружение с помощью поисковой машины и переход на сайт, либо через переход на него с другого сайта по ссылке).

Таблица 2

Top 20 официальных сайтов университетов России

№	Название университета	*WRR*	Rank
1	Новосибирский государственный университет	0,084	1
2	Московский государственный университет	0,077	2
3	Южно-Уральский государственный университет (Челябинск)	0,071	3
4	Новгородский государственный университет	0,046	4
5	Дальневосточный государственный университет (Владивосток)	0,037	5—6
6	Саратовский государственный университет	0,037	5—6
7	Белгородский государственный университет	0,031	7
8	Томский государственный университет	0,028	8
9	Алтайский государственный университет	0,027	9
10	Казанский государственный университет	0,025	10
11	Нижегородский государственный университет	0,023	11
12	Оренбургский государственный университет	0,021	12
13	Санкт-Петербургский государственный университет	0,020	13
14	Дагестанский государственный университет	0,018	14
15	Пермский государственный университет	0,017	15—16
16	Пензенский государственный университет	0,017	15—16
17	Воронежский государственный университет	0,016	17
18	Петрозаводский государственный университет	0,015	18—19
19	Поморский государственный университет (Архангельск)	0,015	18—19
20	Горно-Алтайский государственный университет	0,014	20

5. Проводится ранжирование сайтов по убыванию значений *WRR*.

Для реализации процедуры ранжирования сайтов университетов России были использованы данные проекта LiveInternet по статистике сайта "Сайты Рунета" за три месяца на декабрь 2007 года [7].

В качестве значений *Py*, *Pr* и *Pg* были приняты данные по статистике переходов на сайт "Сайты Рунета" (0,451, 0,144 и 0,273 соответственно).

Для вычисления значений *Ps* и *Pv* были взяты следующие значения:

- переходы с поисковых систем $SE = 23\ 518\ 342$,
- переходы с каталогов и рейтингов $Cat = 1\ 613\ 767$,
- число переходов по ссылкам $L = 778\ 711$.

Ps и *Pv* вычислялись по формулам

$$Ps = \frac{SE}{SE + Cat + L}; \quad Pv = \frac{Cat + L}{SE + Cat + L}.$$

Первые двадцать результатов ранжирования приведены в табл. 2.

Заключение

В статье предложен один из возможных подходов к ранжированию официальных сайтов университетов, основанный на их узнаваемости в Интернете.

Сайты, составляющие Интернет-ресурсы университета, создаются для достижения разных целей с различным акцентированием на тех или иных функциях. Очевидно, что официальный сайт университета создается для достижения иных целей, нежели сайт электронной библиотеки. Возможно основной трудностью при разработке моделей ранжирования сайтов как раз и является отсутствие четко сформулированных целей их создания. Вместе с тем, мы должны уметь формулировать эти цели, если хотим оценивать эффективность наших затрат.

Адекватность предложенного критерия реальному положению дел — это достаточно серьезная тема для продолжения проведенного исследования. Еще одним направлением его развития может быть построение моделей эффективности функционирования сайтов с точки зрения их узнаваемости, — для этого надо уметь оценивать реальные объемы сайтов и затраты на их поддержку.

Разработка математических моделей, в свою очередь, может послужить основой для более точного формулирования целей создания сайтов посредством количественного сравнительного анализа некоторых характеристик уже созданных ресурсов. А значит, позволит определить и направления их дальнейшего развития.

Список литературы

1. **Thelwall M., Vaughan L., Björneborn L.** Webometrics // Annual Review of Information Science and Technology. 2005. Vol. 39. P. 81—135.
2. **Portal** de estudios cuantitativos en Internet. [Электронный ресурс]. 2007. — Режим доступа: http://internetlab.cindoc.csic.es.
3. **Webometrics** Ranking of World Universities. [Электронный ресурс] — 2007. — Режим доступа: http://www.webometrics.info.
4. **Berlin** Declaration on Open Access to Knowledge in the Sciences and Humanities. Conference on Open Access to Knowledge in the Sciences and Humanities, October 20—22, 2003, Berlin. [Электронный ресурс] —http://www.zim.mpg.de/openaccess-berlin/berlindeclaration.html.
5. **Индекс** цитирования. [Электронный ресурс] — 2007. — Режим доступа: http://help.yandex.ru/catalogue/?id = 873431.
6. **Ранжирование** официальных сайтов университетов России. [Электронный ресурс]. 2007. Режим доступа: http://www.krc.karelia.ru/HP/pechnikov/.
7. **Статистика** сайта "Сайты Рунета". [Электронный ресурс] — 2007. —Режим доступа: http://www.liveinternet.ru/stat/ru/.

CONTENTS

Адрес редакции:

107076, Москва, Стромынский пер., 4/1

Телефон редакции журнала **(495) 269-5510**
E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Художник *В.Н. Погорелов*.
Технический редактор *О. А. Ефремова*. Корректор *Т. В. Пчелкина*

Сдано в набор 08.09.2008. Подписано в печать 14.10.2008. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 9,8. Уч.-изд. л. 11,08. Заказ 1123. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.
Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"
142110, Московская обл., г. Подольск, ул. Кирова, 15