ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

4(152) 2009

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Информация о журнале доступна по сети Internet 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 004.272.3

Ф. М. Путря, аспирант, МИЭТ,
e-mail: fm-aka-killer@yandex.ru

Архитектурные особенности процессоров с большим числом вычислительных ядер

Проанализированы причины, обусловившие смену методов наращивания вычислительной мощности микропроцессоров и переход к разработке и производству многоядерных процессоров. Раскрыта проблема ограничения производительности многоядерных процессоров, обусловленная особенностями внутрикристальной коммутации. Проанализированы особенности ряда промышленно выпускаемых многоядерных процессоров и нескольких исследовательских проектов многоядерных систем. Проведен сравнительный анализ подходов к построению коммутационной логики в многоядерных процессорах. Выявлены архитектурные особенности перспективных многоядерных систем, содержащих большое число вычислительных ядер (несколько десятков ядер и более).

Ключевые слова: *многоядерные процессоры, коммутационная логика, производительность, масштабируемость, многоядерные архитектуры, системы с асимметричным доступом к памяти (NUMA), сети на кристалле, массовый параллелизм.*

Долгое время рост производительности каждого следующего поколения процессоров обеспечивался за счет увеличения рабочей частоты вычислительного ядра. Впоследствии этот путь стал менее привлекательным, поскольку потребляемая мощность процессоров, спроектированных для работы на высокой частоте, оказывалась неприемлемо высокой, что приводило к проблемам отвода теплоты от кристалла, снижению надежности такой системы, а также исключало возможность использования процессоров в мобильных системах. Поэтому при проектировании процессора особое внимание стало уделяться достижению оптимального соотношения производительности и потребляемой мощности. Помимо роста потребляемой мощности увеличение длины вычислительного конвейера, которое необходимо для поднятия рабочей частоты ядра процессора, приводит к ряду проблем, связанных с очисткой конвейера при неправильном предсказании ветвления и блокировками, вызванными зависимостями между инструкциями, что снижает эффективность работы процессора. Особенно ярко это было выражено в микроархитектуре NetBurst, предложенной Intel. Процессоры на основе данной микроархитектуры, работая на более высокой частоте, практически не давали прироста производительности по сравнению с предыдущим поколением процессоров. Поэтому дальнейшие усилия разработчиков были направлены на увеличение числа инструкций, выполняемых вычислительным ядром процессора за такт, поскольку это дает прирост производительности даже без увеличения рабочей частоты ядра.

Большинство выпускаемых в настоящее время процессоров являются суперскалярными. Суперскалярные процессоры за счет механизма внеочередного исполнения команд могут на параллельных конвейерах исполнять за один такт несколько инструкций последовательного кода. Для организации внеочередного исполнения команд в таких процессорах необходим дополнительный набор управляющей логики, которая не выполняет непосредственно вычислений. Кроме того, увеличение числа операционных устройств не приводит к пропорциональному увеличению производительности, так как в суперскалярных процессорах, которые не используют явный параллелизм на уровне инструкций, далеко не всегда есть возможность загрузить сразу все исполнительные устройства. Эмпирическим путем показано, что для суперскалярных процессоров при линейном росте аппаратных затрат (числа транзисторов) наблюдается рост производительности, пропорциональный корню квадратному из числа транзисторов [1]. Так, сравнение двух поколений процессоров Alpha показало, что суперскалярный процессор Alpha 21264 (EV6), занимающий площадь в 5 раз большую, чем процессор с последовательным исполнением команд Alpha 21164 (EV5), имеет производительность лишь в 2 раза выше [2]. Стало очевидно, что дальнейшее усложнение одного вычислительного ядра процессора при значительных аппаратных затратах дает лишь небольшой рост производительности.

Переход к многоядерным процессорам

Достигнутая в современных кристаллах (произведенных по технологическим нормам 45—90 нм) степень интеграции, составляющая порядка миллиарда транзисторов, позволяет разместить на одном кристалле несколько относительно сложных

вычислительных ядер. Это дало импульс развитию многоядерных процессоров, которые оказались намного более выигрышными с точки зрения потребляемой мощности и эффективности использования транзисторов.

Изготовление многопроцессорной системы, или многокристальной сборки, в одном корпусе дешевле производства кристалла, содержащего аналогичное число вычислительных ядер, что обусловлено более высоким процентом выхода годных для небольших схем. Это обстоятельство явилось одной из причин того, что первые четырехъядерные процессоры, выпускаемые корпорацией Intel на основе микроархитектуры Core2, представляли собой не один кристалл, а многокристальную сборку из двухъядерных процессоров. Однако у многоядерного решения, реализованного в одном кристалле, есть явное преимущество над многопроцессорной системой, которое и определило выбор направления развития вычислительной техники именно в сторону многоядерных процессоров. Одним из основных факторов, определяющих производительность системы с множеством вычислительных узлов, является скорость обмена данными между этими узлами. Однако максимальное число внешних выводов отдельного кристалла составляет число порядка 1000, при этом только часть выводов используется для коммутации в многопроцессорной системе. Для логического блока внутри кристалла это ограничение составляет несколько десятков тысяч линий связи, что вместе с возможностью передачи данных на частоте ядра делает пропускную способность каналов связи между элементами в многоядерной системе на порядки большей по сравнению с многопроцессорным аналогом [3]. В случае же успешного внедрения технологии внутрикристальных оптических приемопередатчиков пропускная способность линий связи на кристалле возрастет многократно, и перевес в сторону многоядерных чипов станет еще более ощутимым. Серьезные разработки области оптической коммутации в данный момент ведутся фирмами IBM и SUN.

Первые многоядерные процессоры. Симметричная архитектура

Наиболее популярные многоядерные процессоры общего назначения (например, как процессоры с микроархитектурой Core2 от Intel или процессоры Phenom от AMD) построены по принципу симметричной системы с общей памятью. При этом соединение ядер с общей внутрикристальной кэш-памятью в первых многоядерных процессорах, как правило, осуществлялось посредством общей шины, что позволяло при минимальных аппаратных затратах обеспечить целостность кэш-памяти за счет механизма "подглядывания" за общей шиной (snooping). Так, корпорацией Intel в процессе разработки микроархитектуры Core были проанализированы три варианта двухъядерного процессора. В первом варианте два ядра без общей кэш-памяти соединялись внешней системной шиной. Другие два варианта были с общей внутрикристальной кэш-памятью второго уровня (один с шинной организацией, другой — с коммутатором) с использованием механизма поддержания когерентности кэш-памятей на основе общего каталога. В последнем случае каждая строка общей кэш-памяти расширяется дополнительными битами, хранящими информацию о том, какими именно ядрами разделяется данная строка. Эти биты и образуют каталог. Эффективность последних двух вариантов была примерно одинаковой, причем заметно выше, чем у варианта без общей кэш-памяти. Для микроархитектуры Core компанией Intel была выбрана шинная организация по причине более простой аппаратной реализации и меньшей потребляемой мощности, что определило ее успех в секторе мобильных процессоров.

Работы над созданием многоядерных процессоров уже давно ведутся множеством фирм, и имеется широкий спектр проектов многоядерных вычислительных систем, построенных преимущественно на ядрах с архитектурой с сокращенным набором команд (RISC). Еще во второй половине 90-х годов прошлого века в Стендфордском университете был разработан проект четырехядерного процессора Hydra, позднее легший в основу многоядерных процессоров фирмы SUN, а фирмой DEC был предложен проект восьмиядерного процессора Piranha. Первый чип, состоящий из двух VLIW (Very Large Instruction Word — архитектура с широким командным словом) ядер с общей кэш-памятью данных MAJC-5200 [4], был выпущен компанией SUN.

Некоторое время спустя компанией IBM выпускается двухъядерный процессор Power4 [5], связь между ядрами в нем также осуществляется посредством общей кэш-памяти, но уже второго уровня. Позднее линейку двухъядерных процессоров IBM продолжили Power5 [6] и Power6 [7]. В Power6 объединение ядер происходит с помощью внешней кэш-памяти третьего уровня, контроллер которой расположен на кристалле и разделяется между двумя ядрами.

В многоядерных процессорах, как и в многопроцессорных системах, полностью загрузить все вычислительные устройства практически невозможно. В результате реальная производительность кристалла значительно ниже его пиковой производительности, что несколько снижает эффективность процессора. Разработка более совер-

шенных компиляторов, ориентированных на параллельные вычисления, делает данную проблему менее существенной. Однако с увеличением числа вычислительных ядер на первый план выходит уже проблема, связанная с организацией взаимосвязей между элементами на кристалле. Эффективность коммутации на основе общей шины, да и эффективность симметричных систем в целом, является высокой только для малого числа ядер. С ростом числа вычислительных ядер эффективность таких систем резко падает. Это особенно хорошо видно из оценки эффективности многоядерной системы, которую можно провести по аналогии с многопроцессорной системой, используя коэффициент масштабирования K, определяемый как отношение общей производительности системы P к числу ядер N и производительности одноядерного процессора $P1$:

$$K = \mathrm{P}/(P1 \cdot N).$$

Для симметричной архитектуры с шинной организацией типичная зависимость коэффициента масштабирования от числа ядер приведена на рис. 1, на котором видно, что применение систем с шинной архитектурой, состоящих более чем из четырех ядер, становится малоэффективным.

По этой причине в новых процессорах, содержащих четырех и более вычислительных ядра, для соединения с общей памятью предпочтительней использовать коммутатор, а не общую шину, что позволит ядрам одновременно обращаться к разным участкам общей памяти. Именно такой подход использовала компания Intel при создании новой микроархитектуры Nehalem, в которой, как и в процессоре Phenom от AMD, предусмотрена общая кэш-память третьего уровня. Для поддержания когерентности кэш-памятей первого и второго уровней в ней используется система на основе общего каталога, которая усложняет аппаратную часть подсистемы памяти, однако делает

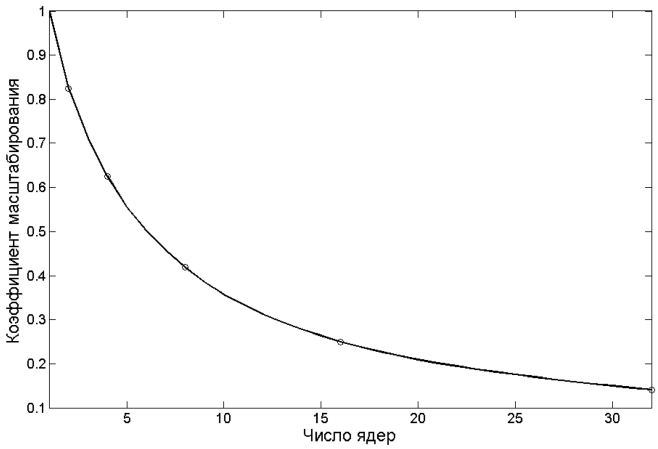


Рис. 1. Зависимость коэффициента масштабирования симметричной многоядерной системы с шинной архитектурой от числа ядер

ее более эффективной. Кроме того, в микроархитектуре Nehalem используется включающая (inclusive) кэш-память. С одной стороны, это несколько снижает эффективный объем внутрикристальной кэш-памяти, а с другой — значительно уменьшает объем трафика, необходимого для поддержания когерентности кэш-памятей. Кэш-память третьего уровня имеет большое время доступа, и в целях повышения эффективности работы с памятью в Nahalem применена многопоточная организация ядра.

На отечественном рынке можно отметить готовящийся к выходу многоядерный процессор, разрабатываемый ГУП НПЦ ЭЛВИС. Процессор состоит из управляющего RISC-ядра и кластера из четырех специализированных ядер. Кластер построен как система с асимметричным доступом к памяти, а ядра в кластере связываются коммутатором типа точка—точка.

Дальнейшее увеличение числа ядер

Сосредоточив усилия на разработке серверных процессоров, фирма SUN выпускает восьмиядерный процессор Niagara [8], а позднее его модернизацию — процессор Niagarall. Niagara состоит из восьми мультитредовых ядер (каждым ядром поддерживалось четыре потока). Процессор представляет собой симметричную систему с общей кэш-памятью второго уровня. Отличительной особенностью процессора является то, что ядра соединены с кэш-памятью второго уровня через неблокируемый глобальный коммутатор типа точка—точка (Crossbar). Общая пропускная способность коммутатора составляет 200 Гбайт/с. Для поддержания когерентности кэш-памяти первого уровня используется механизм сквозной записи в кэш-память второго уровня (write-through). Для ряда приложений такой механизм поддержания целостности данных может привести к существенной потере производительности вследствие регулярных обращений к памяти более высокого уровня, однако для коммутации типа точка—точка он реализуется с наименьшими аппаратными затратами. Малый объем кэш-памяти первого уровня разработчики процессора попытались скомпенсировать большим числом потоков, способных одновременно выполняться на одном ядре. Таким образом, при кэш-промахе вычислительное ядро не простаивает, ожидая подкачки данных в локальную кэш-память, а переключается на параллельный поток вычислений.

В исследовательском проекте Nahalal [9] для многоядерных процессоров предлагается повысить эффективность кэш-памяти за счет введения небольшого объема общей кэш-памяти. В этом случае вычислительные ядра располагаются во-

круг участка общей памяти, а локальная память каждого ядра располагается на периферии кристалла (рис. 2, см. четвертую сторону обложки). Разделяемые несколькими ядрами данные динамически в процессе работы процессора перемещаются в участок общей памяти, и за счет малого объема этого участка памяти и особого расположения ядер имеется возможность быстрого доступа ядер как к общим данным, так и к локальным данным.

Однако на практике симметричная организация описанных выше многоядерных процессоров оказывается непригодна для дальнейшего масштабирования. С увеличением числа ядер на кристалле для симметричной системы вследствие проблем трассировки значительно усложняется организация коммутации, увеличивается длина линий связи и, соответственно, растут задержки на них, что, в конечном счете, приводит к увеличению времени доступа к памяти и падению эффективности симметричной системы. Поэтому для перехода от вычислительных систем с малым числом ядер (2—4, а в отдельных случаях 8 вычислительных ядер) к системам, содержащим 16 и более ядер, необходимо коренное изменение принципов проектирования внутрикристальной коммутации. Системы с асимметричным доступом к памяти (NUMA) позволяют организовать быстрый доступ вычислительных ядер к локальным участкам памяти с сохранением общего для всех ядер адресного пространства. Такая архитектура процессора в случае большого числа ядер является более эффективной по сравнению с симметричными системами, а также легче масштабируется. Среди отрицательных черт архитектуры следует отметить необходимость оптимизации программ под конкретную архитектуру — это та плата, которую приходится отдавать за возможность построения систем с большим числом ядер. Для таких систем невозможной становится также и полная коммутация всех со всеми, что приводит к необходимости создания сетей на кристалле с поддержкой механизма маршрутизации. На данный момент промышленно уже выпускается несколько процессоров, основанных на таком подходе.

Разработанный совместно IBM, Sony и Toshiba процессор CELL [10, 11] состоит из управляющего ядра Power, обладающего кэш-памятью первого и второго уровней и восьми вспомогательных SIMD (Single Instruction Multiple Data — архитектура мультипроцессора с одним потоком команд и множественным потоком данных) процессорных элементов SPE (Synergistic Processing Element). Все восемь SPE-элементов связаны четырехканальной кольцевой шиной (по два канала на каждое направление), по которой осуществляются DMA (Direct Memory Access)-обмены (рис. 3, см. четвертую сторону обложки). В каждом SPE располагается буферный элемент, позволяющий передавать проходящие через него транзитные запросы по кольцевой шине. Особенностью процессора CELL является отсутствие механизма кэширования для всех восьми специализированных вычислительных ядер, что заметно упрощает всю подсистему памяти. Каждый SPE-элемент содержит небольшой объем локальной памяти (Local Store — LS), а доступ к остальной памяти организуется посредством DMA-обменов. Компенсация большого времени доступа к памяти достигается введением регистрового файла большого объема (на 128 слов, по 128 бит каждое), многопоточной организацией ядра, в которой один из потоков отвечает за подкачку данных и способен самостоятельно инициировать DMA-обмены. Слабым местом кольцевой организации CELL является резкое увеличение диаметра сети с увеличением числа ядер. Диаметр сети определяет расстояние, т. е. число "скачков" между промежуточными узлами сети, которые необходимо выполнить пересылаемому пакету при обмене данными, между самыми удаленными друг от друга узлами сети.

От многоядерных процессоров к системам с "массовым параллелизмом"

Первые попытки создания системы с большим числом ядер (16 и более) были предприняты разработчиками 16-ядерного процессора RAW [12]. Процессор представлял собой сеть на кристалле, с топологией "двумерная решетка". Создатели процессора назвали такую организацию плиточной архитектурой (tile architecture), поскольку топология процессора напоминает плиточную кладку (рис. 4, см. четвертую сторону обложки). 16 ядер этого процессора располагались на кристалле в виде симметричной матрицы 4 × 4. При этом каждый элемент системы имел только ортогональные соединения с непосредственными соседями. Для осуществления коммутации со всеми элементами системы в каждом из 16 вычислительных модулей помимо непосредственно ядра с MIPS-архитектурой (Microprocessor without Interlocked Pipeline Stages — семейство RISC-микропроцессоров) имеется маршрутизатор, перенаправляющий транзитные пакеты, проходящие через конкретный элемент. Процессор RAW был только исследовательским проектом, однако в данный момент уже выпускается коммерческий 64-ядерный процессор TILERA [13], в основу которого легли результаты, полученные при работе над проектом RAW. В каждом узле TILERA, в отличие от RAW, вместо простого MIPS использовалось уже VLIW-ядро. Аналогичные идеи легли в основу 80-ядерного процессора компании Intel, развивающей проект Terascale Computing [14].

Исследовательская группа Стендфордского университета Smart Memories [15] в данный момент занимается разработкой реконфигурируемой многоядерной системы, используя при этом настраиваемые вычислительные ядра xtensa_1x2 компании Tensilica [16]. Предлагаемая система построена по плиточному принципу по аналогии с процессором RAW, однако имеет более сложную иерархию (рис. 5, см. четвертую сторону обложки). Вычислительные кластеры (Quad), состоящие из четырех вычислительных узлов (Tiles), связаны между собой сетью, имеющей топологию "двумерная решетка". Вычислительный кластер является настраиваемым и может работать как четырехъядерная система либо как одно VLIW-ядро (в этом случае инструкция декодируется одновременно четырьмя вычислительными узлами). Четыре вычислительных узла связываются в кластере с помощью кластерного интерфейса (QI — Quad Interface), через который осуществляется обмен данными ядра одного вычислительного узла с памятью другого вычислительного узла того же кластера. В состав одного вычислительного узла (Tile) входят два вычислительных ядра, связанных через коммутатор с набором блоков памяти. Время доступа ядра к внутренней памяти его вычислительного узла (Tile) составляет два такта, время доступа к памяти другого вычислительного узла в этом же кластере (Quad) составляет уже пять тактов, доступ к памяти других кластеров осуществляется через коммутационную сеть с использованием механизма маршрутизации. Память ядра может работать в трех режимах: кэш-память, FIFO и сверхоперативная память, что позволяет настраивать систему под различные классы задач.

Исследовательский проект TRIPS [17] предлагает новый тип многоядерных систем, основанных на адаптивном подходе к вычислениям. В основе TRIPS лежит новая архитектура под названием EDGE (Explicit Data Graph Execution — явное выполнение графа данных), с помощью которой становится возможным эффективно обрабатывать большие массивы информации. Процессор TRIPS состоит из нескольких ядер (2—4 ядра), каждое из которых представляет собой полиморфную структуру, состоящую из большого числа вычислительных узлов, способных вместе работать как один процессор. Отличительной особенностью является то, что в процессе исполнения инструкции на каком-либо вычислительном узле нужные операнды не выбираются из конкретных регистров, а непосредственно перенаправляются к исполняющему инструкцию арифметико-логическому устройству (АЛУ) от узла, в котором были получены результаты, необходимые для исполнения инструкции. Задача построения графа вычислений для оптимальной загрузки узлов ложится на компилятор.

Еще дальше от классической Неймановской архитектуры стоит проект WaveScalar [18], в котором вычислительные узлы интегрированы в структуру кэш-памяти. Результат, полученный на таких вычислительных узлах, перенаправляется непосредственно в узел, в котором исполняется инструкция, использующая эти данные. Блоки, совмещающие АЛУ и кэш-память, сгруппированы в кластеры, размещенные в виде двумерной решетки.

Заключение

Подводя итог, можно отметить, что в технологии создания многоядерных вычислительных систем обнаруживаются следующие общие тенденции:

- будет создаваться все больше процессоров, состоящих из ядер разной архитектуры. Как правило, на ядрах общего назначения в этом случае работает операционная система, а на специализированных ядрах выполняются вычисления. Примерами могут служить процессор CELL и перспективные разработки AMD в области объединения центрального и графического процессора на одном кристалле;
- в перспективе в многоядерных процессорах доминирующей станет организация асимметричного доступа к памяти;
- с точки зрения коммутации для многоядерных систем будет характерно использование сетей на кристалле с регулярной структурой, возможно, с применением иерархической коммутации с регулярным построением на верхнем уровне, например, как в проекте Smart Memories, а также применение децентрализованной системной логики;
- для многоядерных систем с большим числом ядер, в которых уже не используется общая шина, значительно усложняется логика поддержания когерентности кэш-памятей всех ядер. В ряде случаев, например, для специализированных процессоров, ориентированных на потоковые вычисления, более эффективным методом будет замена механизма кэширования на программный механизм подкачки данных с помощью DMA. В случае же применения кэш-памяти, скорее, будут использоваться усовершенствованные протоколы поддержания когерентности на основе распределенного каталога;
- как в случае с кэшированием, так и в случае без кэширования стоит проблема большого времени доступа к памяти, которая физически располагается далеко от вычислительного ядра, выполняющего обмен с памятью. Для компенсации большого времени доступа необходимо применение многопоточной организации ядра, увеличение регистрового файла, а также использование механизмов потоковой подкачки

данных, возможно, даже включенных в систему инструкций, например, как в SIMD расширении SSE4, предложенном Intel;

- производительность перспективных многоядерных систем будет определяться уже не столько производительностью входящих в их состав вычислительных ядер, сколько эффективностью межъядерной коммутации.

Архитектурные особенности будущих вычислительных систем диктуются проблемами, возникающими при их аппаратной реализации. Переход же к массовому использованию подобных систем будет определяться степенью готовности программного обеспечения, ориентированного на параллельные вычисления, и компиляторов, способных эффективно использовать как параллелизм на уровне инструкций, так и параллелизм на уровне потоков, а также учитывать архитектурные особенности систем с асимметричным доступом к памяти.

Список литературы

1. **Hennessy J. L., Jouppi N. P.** Computer technology and architecture: An evolving interaction // Computer. 1991. Vol. 24, Issue 9. P. 18—29.
2. **Kumar R., Farkas K., Jouppi N., Ranganathan P.** A multicore approach to addressing the energy-complexity problem in microprocessors http://citeseer.ist.psu.edu/kumar03multicore.html.
3. **Dally W. J., Towles B.** Route Packets, Not Wires: On-Chip Interconnection Networks // DAC 2001, 2001. P. 684—689.
4. **Tremblay M.** Majc-5200: A vliw convergent MPSOC // In Microprocessor Forum. 1999.
5. **Power4** Design Overview. http://www.research.ibm.com/power4.
6. **IBM.** Power5: Presentation at microprocessor forum. 2003.
7. **Le H. Q., Starke W. J.** IBM POWER6 microarchitecture // IBM J. Res. Dev. 2007. V. 51. N 6.
8. **Kongetira P., Aingaran K., Olukotun K.** Niagara: A 32-way multithreaded spare processor // IEEE MICRO Magazine. 2005. Vol. 25. Issue 2. P. 21—29.
9. **Guz Z., Keidar I., Kolodny A., Weiser U. C.** Nahalal: Cache Organization for Chip Multiprocessors // IEEE Computer Architecture Letters. 2007. Vol. 6. Issue 1.
10. **Kahle J. A., Day M. N., Hofstee H. P., Johns C. R., Maeurer T. R., Shippy D.** Introduction to the cell multiprocessor // IBM Journal of Research and Development. 2005.
11. **Hofstee H. P.** Power efficient processor architecture and the cell processor // 11th International Symposium on High-Performance Computer Architecture (HPCA'05). 2005. P. 258—262.
12. **Waingold E., Taylor M., Srikrishna D., Sarkar V. et al.** Baring it all to software: Raw machines // Computer. 1997. Vol. 30. Issue 9. P. 86—93.
13. **Tile** Processor Architecture. Technology Brief — 2007 // www.tilera.com.
14. **A Tera-scale** Computing Research Overview // www.intel.com.
15. **Smart** Memories Project http://www-vlsi.stanford.edu/smart_memories/
16. www.tensilica.com.
17. **Burger P., Keckler S. W.** Scaling to the end of silicon with EDGE architectures // Computer. 2004. Vol. 37. Issue 7. P. 44—55.
18. **Swanson S., Michelson K., Schwerin A., Oskin M.** WaveScalar // International Symposium on Microarchitecture (MICRO-36 2003). 2003.

УДК 004.382

С. Г. Бобков, канд. техн. наук,
НИИ системных исследований РАН, г. Москва,
e-mail: bobkov@cs.niisi.ras.ru

Высокоэффективный адаптер коммуникационной среды многопроцессорной ЭВМ

Рассмотрены методики повышения эффективности обмена данными для коммуникационной среды SAN промышленного назначения.

Ключевые слова: *коммуникационная среда, технология параллельных компьютеров, коммуникационная среда масштаба вычислительной системы.*

Один из основных способов повышения производительности вычислительной системы — это увеличение числа вычислительных устройств, т. е. построение параллельной ЭВМ. При построении высокопроизводительного параллельного компьютера ключевыми являются два вопроса: создание мощного базового процессорного узла и создание эффективной подсистемы коммуникаций, связывающей базовые узлы друг с другом — коммуникационной среды масштаба вычислительной системы. Эффективность таких систем в наибольшей степени определяется коммуникационной средой. Наиболее мощные суперЭВМ созданы на базе относительно низкочастотных микропроцессоров, но с мощной коммуникационной средой. Существует целый ряд стандартов и спецификаций для создания коммуникационных сред, такие как Virtual Interface [1, 2], InfiniBand [3], SCI [4], Myrinet [5, 6], RapidIO [7] и др. Коммуникационная среда масштаба вычислительной системы (ВС) служит для создания на базе независимых компонентов (вычислительных модулей, устройств ввода/вывода, периферийных устройств и др.) единой распределенной вычислительной системы, параллельной ЭВМ. В англоязычной литературе для таких коммуникационных сред введен термин SAN — *System Area Network*. Этот термин вводится, чтобы отличить такие среды от сред для LAN [8] — *Local Area Net-*

works — локальных вычислительных сетей (ЛВС). Особенность SAN, которая отличает ее от ЛВС — это направленность на организацию эффективных обменов информацией между процессами, выполняющимися одновременно на вычислительных узлах системы, в то время как задача ЛВС — это организация обмена информацией между машинами сети.

Публикации отечественных разработок микросхем по созданию сетей SAN отсутствуют. Для создания высокопроизводительных параллельных ЭВМ общего применения российские разработчики пока вынуждены использовать коммуникационное оборудование зарубежных производителей, главным образом, коммуникационных сред типа Myrinet и SCI на базе устройств и программного обеспечения соответственно компаний Myricom и Dolphinics Interconnect Solutions. Высокопроизводительное оборудование коммуникационных сред, удовлетворяющее требованиям условий индустриального и специального применения, для российского потребителя является коммерчески недоступным.

Классификация коммуникационных сред

Коммуникационные среды для создания параллельных ЭВМ можно выделить в отдельный класс коммуникационных сред по аналогии с разделением на классы глобальных сетей и ЛВС (рис. 1).

В качестве параметра разделения по классам можно указать характерные расстояния, а также назначение сред:

- глобальные сети — объединение независимых ЛВС и удаленных пользователей;
- ЛВС — объединение пользователей в рамках рабочих групп или предприятий;
- среды масштаба ВС — создание параллельной ЭВМ, обеспечение эффективного взаимодействия процессов, идущих параллельно на разных модулях параллельной ЭВМ.

В соответствии с назначением коммуникационных сред масштаба вычислительной системы можно выделить характерные требования, предъявляемые к таким средам [5]:

- малые времена задержек на межпроцессные обмены (1—1000 мкс);
- высокая пропускная способность (1—50 Гбит/с на каждое соединение);
- возможность совмещения по времени передач данных через среду с вычислениями в модулях;
- базирование на стандартах;
- надежные протоколы с управлением потоком и коррекцией ошибок;
- поддержка различных топологий вычислительных систем;
- масштабируемость, конфигурируемость и технологичность исполнения.

Разработчики высокоскоростных коммуникационных сред предлагают специальные коммуникационные интерфейсы, которые, с одной стороны, более полно используют аппаратные возможности коммуникационной среды и, с другой стороны, предоставляют пользователю более быстрый и непосредственный доступ к коммуникационной среде из прикладной программы. Переносимость прикладного программного обеспечения при этом достигается за счет использования стандартов параллельных языков и параллельных расширений языков высокого уровня. Сами стандартные системы параллельного программирования при этом реализуются непосредственно поверх специализированных коммуникационных библиотек. Наибольшая производительность дости-

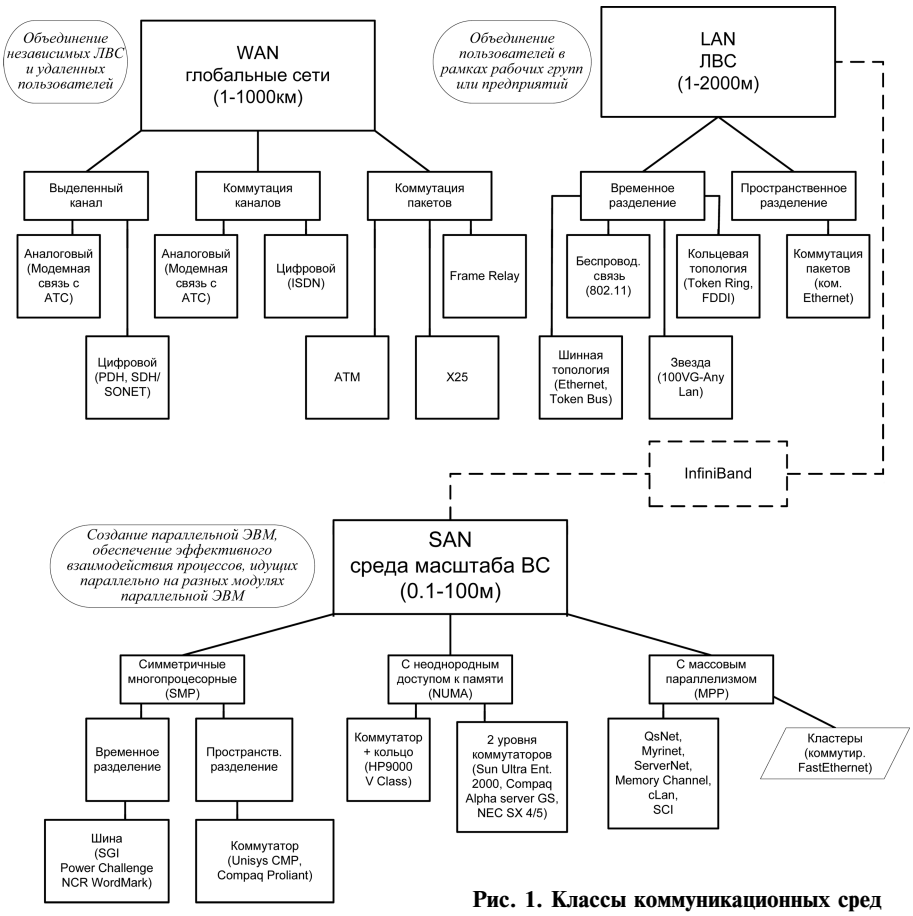


Рис. 1. Классы коммуникационных сред

гается при прямом использовании в прикладных приложениях специализированных коммуникационных библиотек.

В SMP-системах (симметричных мультиплексорных системах) коммуникационный интерфейс обычно реализуется в рамках стандартных средств взаимодействия процессов, предоставляемых операционной системой. Наиболее эффективный механизм при этом — использование разделяемой памяти. Подобный механизм используется, например, интерфейсом MPICH при установке его на SMP-систему. Собственно специальные коммуникационные интерфейсы имеет смысл разрабатывать для MPP-систем (массивно-параллельных систем).

Высокоскоростной адаптер коммуникационной среды

Основная задача интерфейсов для построения параллельных ЭВМ типа MPP — более тесно связать пользовательские приложения и аппаратные возможности коммуникационной среды. Задержки, обусловленные взаимодействием между приложениями, являются одним из основных параметров многопроцессорных систем. В стандартных многоуровневых сетевых протоколах такие задержки в значительной степени связаны с вызовами системных функций операционной системы. Уменьшить задержки можно, если часть функций по взаимодействию между виртуальными адресными пространствами памяти процессов переложить на модуль адаптера коммуникационной среды. При этом адаптер должен иметь в своем составе механизм работы с виртуальными и физическими адресами памяти (рис. 2). Доступ процессов к ресурсам адаптера может осуществляться напрямую в выделенном виртуальном адресном пространстве, без вызова функций ядра операционной системы, адаптер при этом самостоятельно решает вопросы разграничения доступа.

Ввиду того, что программная реализация интерфейса тесно связана с аппаратными ресурсами, разработка коммуникационного интерфейса должна вестись совместно с разработкой коммуникационных устройств. Из используемых коммуникационных интерфейсов для построения систем MPP индустриального применения наиболее распространен интерфейс Myricom GM, он имеет соответствующую реализацию на аппаратуре индустриального исполнения и среде канального уровня Myricom. Недостатками интерфейса GM является отсутствие аппаратной поддержки групповых и широковещательных пересылок, а также отсутствие команд удаленного чтения и модификации данных и операций прямого доступа к удаленной памяти. Аппаратная поддержка групповых и широковещательных пересылок позволяет существенно сократить время на выполнение таких операций в системе и сократить избыточность трафика в среде. Операции прямого доступа к удаленной памяти и удаленного чтения с модификацией позволяют повысить скорость взаимодействия процессов, идущих на удаленных модулях, при выполнении операций синхронизации процессов и доступа к глобальным разделяемым переменным.

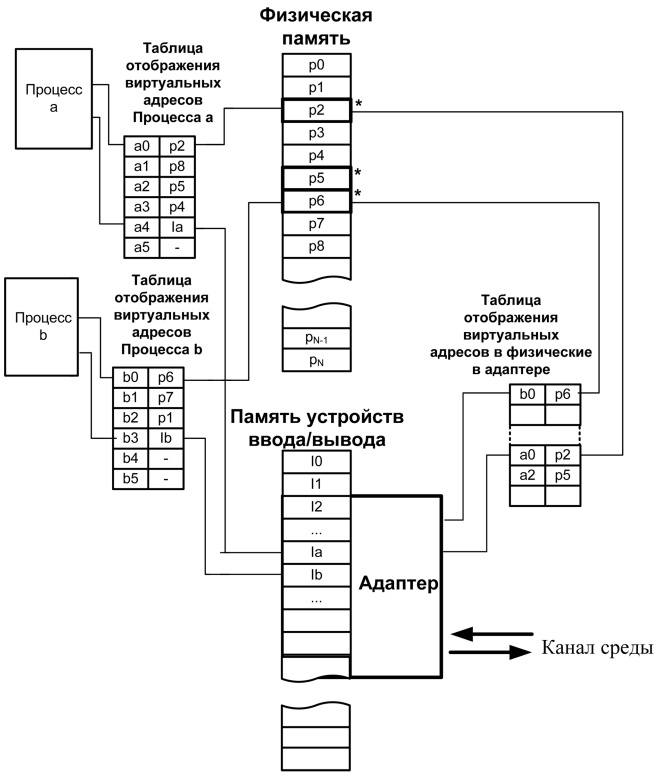


Рис. 2. Принцип функционирования интерфейса адаптера

Разработанный коммуникационный интерфейс (КоИн) [9, 10] является вариантом специализированной среды, реализующей общие принципы, рекомендованные спецификацией Virtual Interface. Все адаптерные функции интерфейса КоИн реализуются в микросхеме интеллектуального коммуникационного контроллера (ИКК), на базе которой разработан модуль адаптера коммуникационной среды. КоИн базируется на таких открытых спецификациях коммуникационных интерфейсов, как Virtual Interface (VI) [1] и InfiniBand [4], и коммуникационной среде канального уровня Myrinet [6]. Также были приняты во внимания решения, предложенные в реализациях коммуникационных интерфейсов GM [11] фирмы Myricom и QNA/QsNet [12] фирмы Quadrics Supercomputer Worlds, а также принципы построения сетевых сред Token Ring [13]и FDDI [14].

В КоИн реализована поддержка аппаратных групповых и широковещательных пересылок за

счет добавления механизмов организации таких пересылок в протокол канального уровня, а также операций удаленного чтения с модификацией, которые отсутствуют в VI и Myricom GM. КоИн поддерживает как схему обмена данными методом послать/принять, по которой работает Myricom GM, так и схему обмена данными методом удаленного прямого доступа к памяти, по которой работает QNA/QsNet.

КоИн имеет четырехуровневую организацию (рис. 3). Уровень прикладной — это библиотека функций, доступная разработчику приложений. Физический уровень подобен Myricom GM, в нем используются приемопередатчики LVDS (низковольтовых дифференциальных сигналов). Транспортный уровень — это реализация механизмов взаимодействия процессов со средой. Канальный уровень — реализация протокола доступа к среде и управление пересылками пакетов.

Транспортный уровень представлен специальными объектами, через обращения к которым процессы прикладного уровня могут взаимодействовать друг с другом. Эти объекты называются портами. Фактический обмен данными в системе осуществляется между портами (рис. 4).

В рамках одной операционной системы каждый порт имеет уникальный адрес. Чтобы обратиться к какому-то порту, необходимо знать адрес операционной системы, в которой он находится (например, адрес SMP-узла), и номер этого порта внутри данной операционной системы. Порты могут обмениваться данными как в режиме с предварительным установлением соединения, так и в режиме без предварительного установления соединения (датаграммном — *datagram*). Каждый порт имеет определенные атрибуты поддерживаемого уровня надежности и качества обслуживания. Обмен данными может осуществляться только между портами, обладающими одинаковыми соответствующими атрибутами.

Канальный и физический уровни могут быть двух видов. В случае если взаимодействующие порты принадлежат одному SMR-узлу, обмен данными может осуществляться с использованием стандартного механизма IPC (Inter-Process Communication) — разделяемой памяти [15]. Обработка таких взаимодействий ложится на центральный процессор. В случае многомодульной машины используется специальная коммуникационная среда, образованная совокупностью адаптеров, коммутаторов и каналов межсоединений. В случае если система состоит из небольшого числа узлов (до 15), коммутаторы могут отсутствовать и среда может иметь кольцевую топологию.

Интерфейс поддерживает две программные модели взаимодействия процессов:

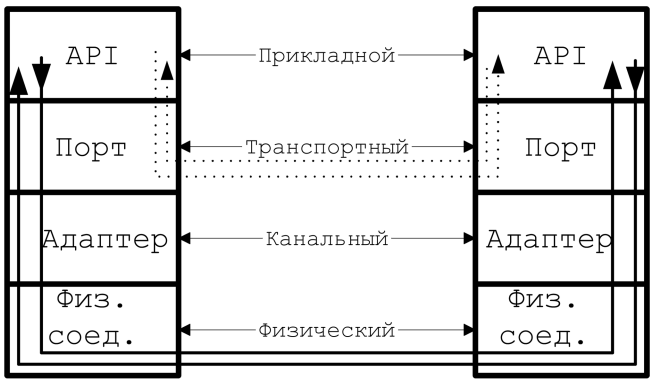


Рис. 3. Четырехуровневая модель КоИн

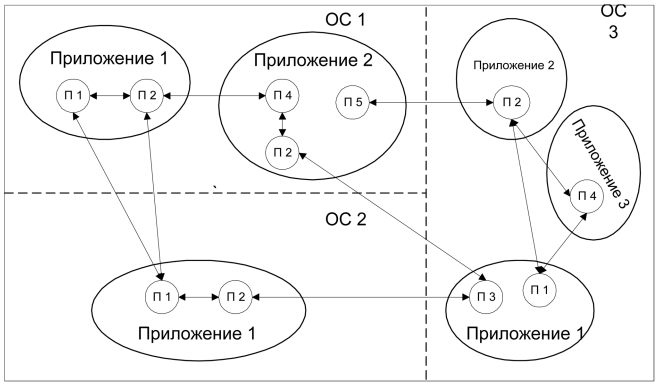


Рис. 4. Порты — взаимодействующие объекты

- модель обмена сообщениями послать/принять (Send/Receive);
- модель прямого доступа к удаленной памяти (RDMA);

  (a) удаленная запись (RDMA Write);

  (b) удаленное чтение (RDMA Read);

  (c) удаленное чтение с модификацией (RDMA Atomic).

Модель обмена сообщениями предполагает наличие некоторого соглашения по передаваемым и принимаемым данным между сторонами-участниками обмена. В этой модели обмены всегда происходят с уведомлением противоположной стороны.

В *модели прямого доступа к удаленной памяти* обмен данными может вестись непосредственно со стороны удаленного процесса без уведомления противоположной стороны. Эта модель в общем случае не требует никаких механизмов синхронизации, и поэтому такие операции могут выполняться значительно быстрее, чем операции первого типа.

Все пересылки описываются специальными управляющими структурами — дескрипторами. Бывают два типа дескрипторов:

- дескрипторы операций Send/RDMA;
- дескрипторы операций Receive/Notify.

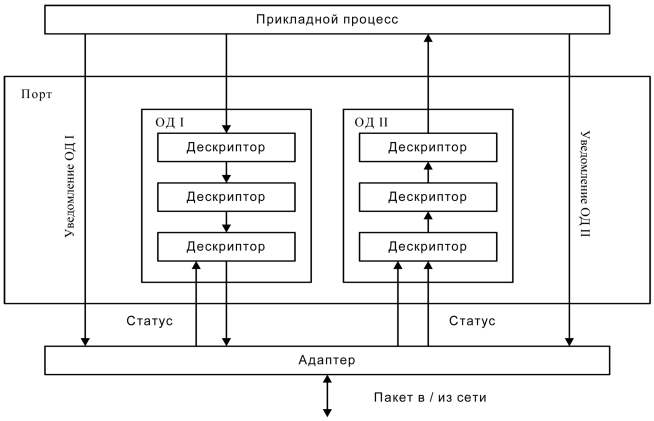


Рис. 5. Очереди дескрипторов

Для каждого из этих типов приложение может создать в рамках порта одну и более очередей ОД (очередь дескрипторов) (рис. 5). Дескрипторы первого типа инициируют работу механизмов по передаче данных. В модели Send/Receive каждому дескриптору Send на передающей стороне должен однозначно соответствовать дескриптор Receive на принимающей стороне. Дескрипторы второго типа обеспечивают механизм надежного и безопасного приема данных.

Дескрипторы типа Notify отсутствуют в спецификации Virtual Interface и введены для уведомления удаленного процесса о том, что с его памятью была проведена операция прямого доступа, что позволяет повысить отказоустойчивость индустриальных систем и обеспечить синхронизацию процессов, идущих на разных модулях системы.

Приложение может создать специальную очередь завершения (ОЗ) операций, объединяющую несколько очередей дескрипторов. При завершении любого дескриптора в ОЗ добавляется соответствующая запись. Приложение при этом периодически опрашивает лишь верхушку очереди ОЗ (рис. 6).

Работа КоИн обеспечивается функциями трех типов:

- реализуемые на уровне прикладного процесса;
- реализуемые на уровне ядра операционной системы;
- реализуемые адаптером — интеллектуальным коммуникационным контроллером (ИКК).

На уровне ядра операционной системы реализуются функции создания портов и управления ими, а также функции выделения и регистрации памяти. Прикладной процесс работает только с портом. В рамках порта процесс может непосредственно организовывать пересылки и прием данных. Приложение должно зарегистрировать в операционной системе ту память, в которой будут храниться очереди дескрипторов ОД, очереди завершения ОЗ и буферы принимаемых и передаваемых данных. Процесс отвечает за создание и удаление дескрипторов. Непосредственную обработку управляющей информации, содержащейся в дескрипторе, а также заполнение ОЗ выполняет адаптер среды. Поэтому адаптер должен иметь доступ к соответствующим областям памяти, для чего они и регистрируются в ОС. После добавления дескриптора в любую из очередей прикладной процесс добавляет короткое оповещение-ссылку на этот дескриптор в специальную очередь оповещений, находящуюся в адаптере. Область очереди оповещений адаптера должна быть отображена в виртуальную память процесса. При приходе очередного оповещения адаптер узнает о появлении дескриптора в ОД, который он должен обработать. Адаптер считывает дескриптор из пространства памяти процесса и выполняет описанную в нем операцию. После завершения выполнения операции адаптер либо выставляет в соответствующем дескрипторе флаг завершения операции, либо добавляет запись в ОЗ. Для некоторых видов дескрипторов адаптер может генерировать аппаратные прерывания.

Для построения индустриальных параллельных ЭВМ с различными показателями отказоустойчивости КоИн имеет три уровня надежности:

- ненадежная доставка;
- надежная доставка;
- надежное получение.

Режим *ненадежной доставки* имеет место, когда после обработки дескриптора первого типа инициатор не ожидает результата удаленной операции. Флаг завершения выставляется сразу же после выполнения соответствующей операции пересылки. В этом режиме поддерживаются лишь операции Send/Receive и операции удаленной записи RDMA Write. Ошибки передачи интерфей-

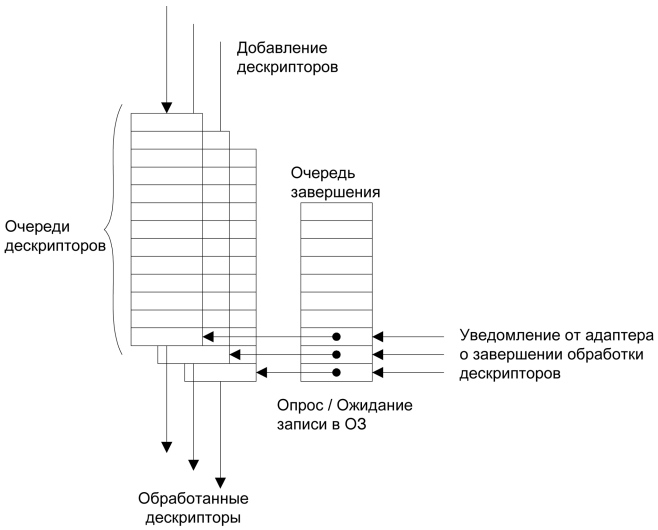


Рис. 6. Очереди завершения операций (ОЗ)

сом не детектируются и может происходить потеря пакетов.

В режиме *надежной доставки* флаг завершения дескриптора выставляется только после получения пакета-отклика об успешной доставке исходного пакета в удаленный целевой адаптер.

В режиме *надежного получения* операция считается успешно завершенной только после размещения соответствующих данных в операционной памяти целевого модуля. Флаг завершения дескриптора выставляется только после получения соответствующего пакета-отклика.

В режимах надежной доставки и надежного получения в случае если такой отклик не приходит в течение заданного периода тайм-аута, операция автоматически повторяется адаптером. Если несколько попыток подряд окажутся неудачными, адаптер генерирует сигнал фатальной ошибки. В этих режимах поддерживаются операции Send/Receive и операции RDMA Write, RDMA Read, RDMA Atomic.

В системах индустриального применения ряд событии должен обрабатываться за предсказуемое и ограниченное сверху время. Для передачи специальных высокоприоритетных команд и сообщений, время обработки которых критично для системы, в интерфейс КоИн введена двухуровневая схема приоритетов сообщений. Для сообщений с высоким приоритетом организуется своя очередь дескрипторов. Более приоритетные сообщения передаются быстрей, чем низкоприоритетные. Низкоприоритетное сообщение передается либо если канал свободен, либо только после того, как оно пропустит некоторое заданное число высокоприоритетных сообщений.

Прежде чем взаимодействовать с коммуникационным интерфейсом, приложение должно создать порт или несколько портов. При создании порта происходит вызов функции системы, которая назначает порту уникальный адрес, выделяет память под порт, заполняет ряд атрибутов порта, а также создает в адаптере структуру, указывающую на данный порт. Также в адаптере открывается новый раздел в таблице трансляции виртуальных адресов в физические памяти для текущего процесса, если этот процесс еще не создавал портов и не регистрировал страниц.

Порт может содержать очереди следующих видов:

- очереди дескрипторов типа I (Send/RDMA): операции обычного приоритета; операции высокого приоритета;
- очереди дескрипторов типа II: дескрипторы Receive; дескрипторы Notify;
- очереди завершения обработки дескрипторов.

К очередям всех типов имеет непосредственный доступ как пользовательский процесс, так и адаптер коммуникационной среды. Поэтому области памяти, в которых хранятся очереди, должны быть зарегистрированы в операционной системе и защищены от перемещения в другие области и свопирования. Таблицы преобразования виртуальных адресов процесса в физические адреса операционной памяти для соответствующих страниц должны быть доступны адаптеру. При добавлении очередной записи в ОД процесс извещает об этом адаптер. Извещением является короткое сообщение, содержащее адрес соответствующего дескриптора. Извещение пишется непосредственно в адаптер в очередь извещений, поэтому соответствующая область адаптера должна быть отображена в виртуальное адресное пространство процесса.

Адаптер среды имеет прямой доступ не к любым областям памяти процесса, а к специально выделенным страницам. Операционная система при организации работ с памятью и обеспечении механизмов поддержки виртуальности может перемещать страницы из одних областей в другие, буферизовать некоторые страницы во вторичной памяти и т. д. При этом информация, хранящаяся в таблицах трансляции адресов процессов, соответствующим образом изменяется. Для возможности работы с виртуальными адресами адаптер должен иметь копии этих таблиц. Чтобы информация в таблицах адаптера была корректной, необходимо вводить специальные механизмы поддержки когерентности таблиц, что явилось бы расширением стандартных функций операционной системы. Однако чтобы не вносить глубинных изменений в ядро операционной системы, можно использовать другой путь.

Страницы памяти процессов, с которыми работает адаптер, должны быть зафиксированы в физической памяти с тем, чтобы информация в таблицах трансляции адресов процессов в адаптере оставалась неизменной. Процесс может использовать эти страницы либо только как специальные буферы для приема и хранения данных, либо в качестве обычной памяти процессов. В первом случае, прежде чем выполнять коммуникационные операции, процесс должен соответствующим образом подготовить эти буферы, например, скопировать в них данные из обычной рабочей памяти. Во втором случае дополнительных копирований не требуется, и те же буферы используются как для коммуникационных операций, так и для долговременного хранения данных в рамках процесса.

Функциональная модель адаптера в системе согласно спецификации VI приведена на рис. 7.

Со стороны центрального процессора (процессоров) узла адаптер виден как некая область памяти устройств ввода/вывода. Регистры адаптера и области очередей уведомлений о поступлении дескрипторов отображаются в виртуальное адрес-

ное пространство памяти процессов. Процессор может взаимодействовать с адаптером в одностороннем порядке только через эти две области.

Адаптер имеет прямой доступ к операционной памяти системы и может обрабатывать в операционной памяти следующие объекты:

- очереди дескрипторов ОД;
- очереди завершения выполнения дескрипторов ОЗ;
- страницы разделяемой памяти;
- глобальную очередь прерываний.

К областям, в которых хранятся перечисленные объекты, адаптер имеет прямой доступ, поэтому эти области должны быть зарегистрированы в операционной системе и соответствующие таблицы трансляции адресов должны быть занесены в адаптер.

Адаптер может генерировать аппаратные прерывания центральному процессору вычислительного модуля. Эти прерывания должны обрабатываться соответствующим агентом ядра операционной системы.

В состав самого адаптера входят следующие функциональные единицы:

- интерфейс с вычислительным модулем (хост-системой);
- блок общего управления;
- блок хранения и обработки контекстов портов;
- блоки приема/передачи данных на каналы коммуникационной среды.

Взаимодействие всех этих единиц позволяет эффективно реализовывать различные коммуникационные функции автономно на адаптере при минимальных затратах вычислительных ресурсов центрального процессора (процессоров) системы. Кроме того, адаптер может работать в режиме централизованного управления, когда всеми операциями управляет центральный процессор через специальные регистры адаптера, к которым он имеет доступ.

Модуль адаптера включает в себя микросхему интеллектуального коммуникационного контроллера (ИКК) и микросхемы статической и динамической памяти. Структурная схема ИКК показана на рис. 8.

Динамическая память используется для буферизации пакетов и их временного хранения. Статическая память предназначена для временного хранения различной управляющей информации, структур и для использования в качестве рабочей памяти RISC-ядра ИКК.

Интеллектуальный коммуникационный контроллер поддерживает три режима функционирования:

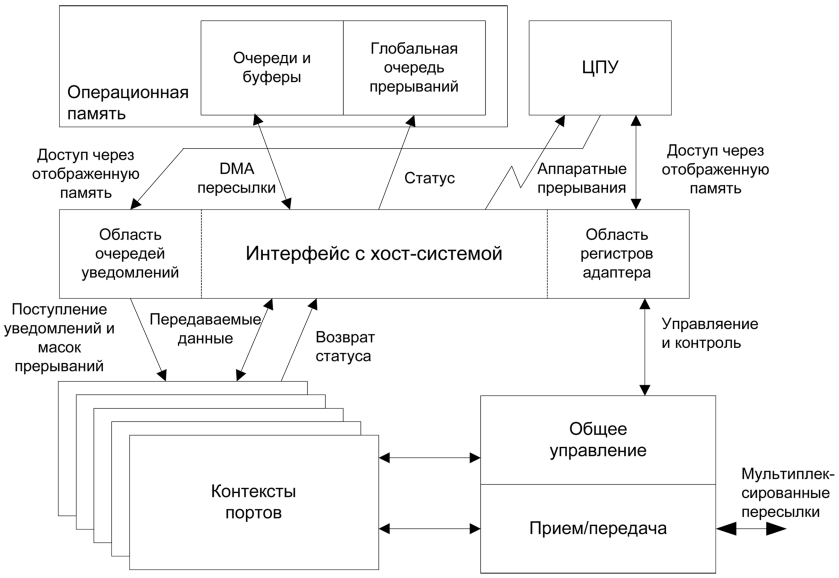


Рис. 7. Обобщенная модель адаптера коммуникационной среды в системе

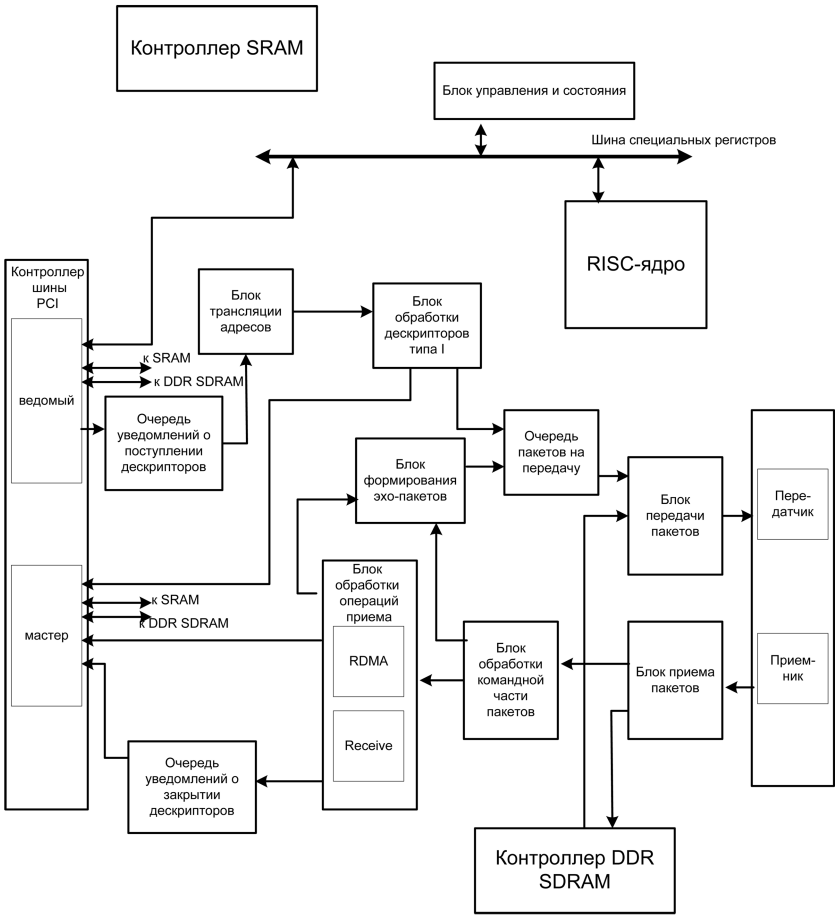


Рис. 8. Структура интеллектуального коммуникационного контроллера

- полное управление потоком данных с центрального процессора вычислительного модуля (хост-системы);
- независимое обслуживание коммуникационных функций на программном уровне встроенного RISC-ядра ИКК;
- аппаратное обслуживание коммуникационных операций на уровне ИКК.

В состав ИКК входит набор специальных регистров, через которые может осуществляться контроль работы ИКК со стороны центрального процессора. Этот режим является самым медленным и требует относительно больших вычислительных затрат центрального процессора.

Второй режим работы позволяет частично разгрузить центральный процессор от обработки интерфейсных функций и возложить их выполнение на RISC-ядро ИКК. Этот режим является более быстрым и позволяет реализовывать на ИКК различные варианты коммуникационных интерфейсов.

Третий режим работы является наиболее производительным. В этом режиме часть коммуникационных функций, наиболее критичных по времени, выполняется аппаратно. Тот или иной характер выполнения операций может выбираться на этапе конфигурирования. Для расширения возможностей по реализации более широкого спектра функций рекомендуется использовать этот режим совместно со вторым режимом. При этом базовые коммуникационные функции будут реализовываться на аппаратном уровне, а функции расширения — на микропрограммном уровне RISC-ядра ИКК.

Предложенная архитектура позволяет повысить производительность адаптера за счет аппаратной реализации наиболее критичных по времени коммуникационных функций, возможности независимой параллельной работы различных блоков потоковой машины и RISC-ядра, разделения памяти на память данных и память управляющей информации.

Аппаратная реализация наиболее критичных по времени коммуникационных функций позволяет увеличить скорость выполнения этих функций по сравнению со скоростью их программного выполнения на RISC-ядре при той же частоте. Моделирование показало, что с использованием PCI при тактовой частоте 100 МГц и 32-разрядной 33-мегагерцовой шине минимальная задержка на передачу сообщения длиной 64 байт из пространства памяти процесса на одном модуле в пространства памяти процесса на соседнем модуле составляет не более 5 мкс.

Возможность независимой параллельной обработки коммуникационных операций в потоковой машине и на RISC-ядре позволяет увеличить число одновременно обрабатываемых коммуникационных операций до 10 (что в 2,5 раза больше, чем на процессоре LANai фирмы Myricom):

1) обмен данными между системной памятью модуля и локальной памятью DDR SDRAM;

2) передача данных из DDR SDRAM в среду;

3) прием данных из среды с буферизацией в DDR SDRAM;

4) трансляция виртуальных адресов в физические;

5) обработка дескрипторов типа I;

6) формирование маршрута пакета;

7) обработка командной части входных пакетов и операций Receive/RDMA;

8) формирование эхо-пакетов;

9) формирование уведомлений о закрытии дескрипторов;

10) выполнение последовательного кода микропрограммы на RISC-ядре.

Разделение памяти на память для данных и память для управления позволяет повысить скорость выполнения коммуникационных операций за счет повышения скорости доступа к памяти, которая достигается в результате уменьшения числа блоков, конкурирующих за доступ к разделяемому ресурсу.

Заключение

Проведенные исследования показали следующее.

Основная задача специализированного высокопроизводительного интерфейса — более тесно связать пользовательские приложения и аппаратные возможности коммуникационной среды. Задержки, обусловленные взаимодействием между приложениями, уменьшаются за счет того, что такие взаимодействия могут происходить вне вызовов системных функций операционной системы, в отличие от реализаций стандартных многоуровневых сетевых протоколов. Достигается это в результате того, что часть функций по взаимодействию между виртуальными адресными пространствами памяти процессов переносится на адаптер коммуникационной среды. При этом адаптер имеет в своем составе механизмы работы с виртуальными и физическими адресами памяти. Доступ процесса к ресурсам адаптера может осуществляться напрямую, без вызова функций ядра операционной системы, адаптер при этом самостоятельно решает вопросы разграничения доступа.

Разработан коммуникационный интерфейс, позволяющий создавать отечественные параллельные ЭВМ индустриального применения с массовым параллелизмом со скоростями обмена данными по каждой линии более 1 Гбит/с и временами задержек на межпроцессорные взаимодействия от 5 мкс, что по характеристикам производительности соответствует зарубежным ком-

мерческим аналогам. В интерфейсе реализована поддержка аппаратных групповых и широковещательных пересылок за счет механизмов организации таких пересылок в протокол канального уровня, а также добавлены операции удаленного чтения с модификацией. Интерфейс поддерживает как схему обмена данными методом послать/принять, так и схему обмена данными методом удаленного прямого доступа к памяти.

Адаптер позволяет одновременно обрабатывать до 10 коммуникационных операций, что в 2,5 раза превосходит зарубежный аналог LANai фирмы Myricom. Выигрыш достигается в результате того, что обработка базовых коммуникационных операций реализуется не микропрограммно на процессорном ядре адаптера, а аппаратно в конвейере на специально разработанной потоковой машине адаптера.

Список литературы

1. **Virtual** Interface Architecture Specification. Version 1.0. Compaq Computer Corp., Intel Corporation, Microsoft Corporation. December 16, 1997.
2. **Virtual** Interface Architecture for SANs. Technology Brief. Compaq Computer Corporation, ECQ Technology Communications. June 1998.
3. **InfiniBand** Architecture Specification. Vol 1, 2. Release 1.0. October 24, 2000. Final.
4. **Корнеев В. В.** Параллельные вычислительные системы. М.: Нолидж, 1999. 320 с.
5. **Myrinet-on-VME** Protocol Specification Draft Standart. VITA 26-199x. Draft 1.1, 31 August 1998.
6. **Boden N., Cohen D., Federman R., Kulawik A., Seizovic J., Myrinet W. Su.** A Gigabit-per-Second Local Area Network // IEEE Micro. Feb. 1995.
7. **RapidIO** Interconnect Specification, www.rapidio.org/specs/current.
8. **Новиков Ю. В., Карпенко Д. Г.** Аппаратура локальных сетей: функции, выбор, разработка / Под общей редакцией Ю. В. Новикова. М.: ЭКОМ. 1998. 288 с.
9. **Бобков С. Г., Сидоров Е. А.** Коммуникационный интерфейс КоИн // Средства разработки высокопроизводительных вычислительных систем. Коммуникационные технологии. Системы памяти. Сб. статей / Под ред. чл.-корр. РАН В. Б. Бетелина. М.: НИИСИ РАН. 2001. С. 88—132.
10. **Sidorov E., Bobkov S., Aryashev S.** Communication Interface CoIn // Parallel Computing Technologies. Lecture Notes in Computer Science, Springer Berlin/Heidelberg. 2001. Vol. 2127. P. 344—349.
11. **The GM** Message Passing System. Myricom, Inc. 1999 (http://www.myri.com/scs/GM/doc/refman.pdf).
12. **QSW** SuperCluster Architecture Overview. Quadrics Supercomputer Worlds Ltd. (http://www.quadrigs.com).
13. **Token** ring access method and Physical Layer specifications // ANSI/IEEE Std 802.5-1998E.
14. **Information** Systems — Fiber Distributed Data Interface (FDDI) — Token Ring Media Access Control (MAC) (formerly ANSI X3.139-1987 (R1997)).
15. **Гордеев А. В., Молчанов А. Ю.** Системное программное обеспечение. СПб.: Питер, 2002. 736 с.

УДК 004.72

В. Ю. Аристархов, аспирант,
Нижегородский государственный технический университет, г. Нижний Новгород,
e-mail: vasily.aristarkhov@intel.com

Алгоритм приема сигнала в целом для высокоскоростных беспроводных сетей

Описан способ организации передачи данных на основе независимых субпоследовательностей для беспроводных высокоскоростных сетей с использованием детектора, реализующего критерий обобщенного максимального правдоподобия. Особенностью разработанного метода приемопередачи является сравнительно низкая вычислительная сложность алгоритма детектирования и устойчивая работа в каналах связи с частотно-селективными замираниями. Для приведенного метода получены кривые помехоустойчивости на основе результатов моделирования.

Ключевые слова: *высокоскоростная беспроводная сеть, критерий максимального правдоподобия, физический уровень, вычислительная сложность, алгоритм детектирования, помехоустойчивость.*

Введение

В настоящий момент одной из актуальных научных проблем телекоммуникации и развития систем связи является разработка методов организации высокоскоростного приемопередающего тракта в персональных беспроводных сетях с возможностью масштабирования частотного диапазона и устойчивой работы в условиях замираний. Именно передача высокоскоростного трафика представляет особый интерес, так как эта технология позволяет заместить имеющиеся проводные каналы (соединения между компьютерами, соединения системного блока с монитором и т. д.) беспроводными, существенно упростив высокоскоростной доступ в сеть. В феврале 2002 года Федеральная Комиссия по связи США (FCC — Federal Communications Commission) предложила

использовать частотный диапазон 3,1...10,6 ГГц для коммерческих приложений [1], что послужило отправной точкой для создания беспроводных сетей передачи данных на базе сверхширокополосных (СШП) сигналов. Первые опыты по применению СШП сигналов для систем связи относятся к 70-м годам прошлого века [2, 3]. Существуют два принципиальных подхода к формированию сигнальных конструкций при использовании СШП сигналов.

Первый заключается в использовании многополосной передачи, при которой выделенный частотный диапазон разбивается на поддиапазоны. Для более эффективного использования частотного ресурса и борьбы с межсимвольной интерференцией (МСИ) применяется уплотнение сигнала с ортогональным частотным разделением (MB OFDM — MultiBand Orthogonal Frequency Division Multiplexing). Суть уплотнения заключается в разбиении последовательности символов данных на параллельный поток с увеличением длительности каждого символа. Для обеспечения ортогональности каждая поднесущая должна содержать целое число колебаний на период символа [4].

Второй подход заключается в передаче маломощных кодированных импульсов в очень широкой полосе без несущей частоты. В эфир излучается не гармоническое колебание, а сверхкороткий импульс, или моноимпульс, длительность которого может колебаться в пределах 0,2...2 нс, а период импульсной последовательности составляет от 10 до 1000 нс.

Ввиду ряда существенных недостатков данных способов передачи информации описанные выше предложения, не приобрели статуса международного стандарта. Основными проблемами стали технологическая сложность устройств, высокая стоимость, невозможность обеспечить высокую пропускную способность.

Методы передачи информации в каналах с памятью

Основной сложностью при разработке беспроводных систем связи является влияние нежелательных помех, связанных с интерференцией прямых и отраженных волн, которая приводит к изменениям амплитуды принимаемых сигналов — замираниям, их динамический диапазон может достигать 40...45 дБ. Канал обнаруживает частотно-селективные свойства в случае, когда принятый многолучевой компонент символа выходит за пределы передачи символа, другим названием этой категории замираний в канале является вводимая в канал МСИ. Канал обнаруживает частотно-неселективные свойства, когда полученные многолучевые компоненты символа поступают в течение длительности передачи символа.

Среди схем приема в каналах с частотно-селективными замираниями можно выделить методы компенсации МСИ на основе использования линейного фильтра с регулируемыми коэффициентами и обратной связи по решению [4, 5]. Данный тип компенсации не всегда качественно работает в каналах с селективными замираниями, характеризируемыми наличием точек на оси частот с нулевыми значениями модуля передаточной характеристики. Корректор, пытаясь дополнить частотную характеристику до "идеальной", дает на этих частотах сомножитель "бесконечность", что ведет к характерным всплескам шума [5]. Другим подходом является использование метода, основанного на критерии максимального правдоподобия принятой последовательности (МППП). Алгоритм детектирования символов в данном случае можно представить следующим образом.

Пусть детектор принимает символ $b_0^{(i)}$, при этом задержка принятия решения равна длительности МСИ: $\tau_d = LT$, где L — число интервалов МСИ. Тогда принимаемый сигнал $r(t)$ принимает вид

$$r(t) = s_0^{(i)}(t) + s_{b_k}(t) + s_{b_l}(t) + z(t),$$

где $s_0^{(i)}(t)$ — сигнал, обусловленный анализируемом символом $b_0^{(i)}$, $i \in \overline{0, M-1}$, M — мощность передаваемого алфавита; $s_{b_k}(t) = g_{\text{ост}}(t)$ — сигнал, который определяет остаточный сигнал МСИ, обусловленный символами, переданными до анализируемого; вектор b_k определяется цепочкой символов, предшествующих анализируемому; $s_{b_l} = g_{\text{сл}}(t)$ — сигнал, который определяет сигнал МСИ, обусловленный символами, переданными после анализируемого; вектор b_l определяется цепочкой символов, переданных после анализируемого [6].

Рассмотрим отношение правдоподобия на интервале $(L + 1)T$ [6]:

$$\Lambda_i(b_k, b_l) = \frac{w[r|b_k, b_0^{(i)}, b_l]}{w[r|ш]}, \qquad (1)$$

где $ш$ — шум. Оптимальный поэлементный приемник по правилу максимального правдоподобия должен выполнить усреднение (1) по всем возможным цепочкам символов b_k и b_l, а затем выбрать максимум $b_0^{(i)}$ по i. Алгоритм его работы можно записать как:

$$\hat{i}_0 = \operatorname{Arg}\max_i \{\overline{\Lambda_i(b_k, b_l)}\}, \qquad (2)$$

где $\overline{\Lambda_i(b_k, b_l)} = \sum_{k=1}^{M^L} \sum_{l=1}^{M^Q} P(b_k, b_l) \frac{w\lfloor r | b_k, b_0^{(i)}, b_l \rfloor}{w[r|w]}$;
Q — число интервалов анализа, обычно выбираемого как $Q \geqslant L$. Алгоритм (2) для каналов с МСИ был впервые предложен К. Хелстром [7], но не нашел широкого применения в практике по причине высокой вычислительной сложности.

Для упрощения схемы приема были предложены два алгоритма, реализующие оптимальный прием по принципу максимального правдоподобия. Первый из них был предложен Д. Д. Кловским в 1960 г. Суть его сводится к идее обратной связи по решению, т. е. оценки, полученные в приемнике до анализируемого символа, считаются достоверными [8]. Обработке подвергается разностный сигнал $r(t) - \hat{g}_{\text{ост}}(t)$, где $\hat{g}_{\text{ост}}(t)$ — надежная оценка МСИ, данный алгоритм описывается выражением

$$\hat{i}_0 = \operatorname{Arg}\max_i \times$$

$$\times \left\{ \sum_{l=1}^{M^Q} p(b_l) \frac{w[r(t) - \hat{g}_{\text{ост}}(t) | b_0^{(i)}, b_l]}{w[r(t) - \hat{g}_{\text{ост}}(t) | w]} \right\}. \qquad (3)$$

Алгоритм (3) учитывает энергию лишь на одном тактовом интервале. Для упрощения реализации и в целях полного использования энергии сигнала был предложен субоптимальный алгоритм, именуемый в литературе алгоритмом Кловского—Николаева (АКН) [9]:

$$\hat{i}_0 = \operatorname{Arg}\max_{il} \left\{ \frac{w[r(t) - \hat{g}_{\text{ост}}(t) | b_0^{(i)}, b_l]}{w[r(t) - \hat{g}_{\text{ост}}(t) | w]} \right\}. \qquad (4)$$

АКН характеризуется постоянной задержкой решения LT для элемента сигнала и постоянной глубиной принятия решения.

Второй подход был предложен Форни и заключается в следующем [9]: МСИ на выходе демодулятора можно представить как машину с конечным числом состояний, что позволяет представить выход канала с МСИ диаграммой решетки, а оценки максимального правдоподобия определяются наиболее вероятным путем по решетке. Очевидно, что алгоритм Витерби [10] обеспечивает эффективный поиск по такой решетке. Задержка при детектировании каждого информационного символа по алгоритму Витерби в отличие от АКН меняется [4].

Существующие алгоритмы, реализующие критерий МППП для канала с МСИ, имеют вычислительную сложность, которая возрастает экспоненциально с длиной временного рассеивания в канале. Для описанных выше случаев алгоритм Витерби и АКН вычисляют M^{L+1} метрик на каждый принимаемый символ соответственно. Можно сказать, что вычислительная сложность C_1 на один переданный символ равна $C_1 \cong M^{L+1}$. Таким образом, высокая вычислительная сложность существующих алгоритмов не позволяет использовать их для субгигабитных и гигабитных скоростей передачи данных.

Методы решения проблемы вычислительной сложности алгоритма МППП для декодирования принятой последовательности. Алгоритм оптимального приема в целом на основе субпоследовательностей

Следующий разработанный метод приемопередачи позволяет существенно упростить схему детектирования согласно критерию МППП.

Пусть число интерферирующих компонентов в канале связи равно $(L + 1)$. Обозначим период последовательности как $T_{\text{посл}} = (L + 1)T$. Разобьем передаваемую последовательность на субпоследовательности длительностью RT интервалов, причем $(L + 1) \geqslant R > 0$. Введем временной зазор (интервал релаксации) длительностью G интервалов между передачами двух соседних субпоследовательностей (рис. 1). Для независимости субпоследовательностей друг от друга с точки зрения МСИ должно выполняться условие $(G + R) \geqslant (L + 1)$. С точки зрения скорости передачи и пропускной способности мы должны выбирать длительность G как можно меньшей. Таким образом, оптимальный размер G равняется $(L + 1 - R)$.

Скважность D определим как

$$D = \frac{L+1}{R}. \qquad (5)$$

При таком способе организации передачи можно применить прием в целом, т. е. детектировать субпоследовательность целиком. Обозначим каждую субпоследовательность как $\nu_i = b_{0,i}, \ldots, b_{R,i}$. Очевидно векторы, определяемые предшествующими и последующими символами, по отношению к ν_i являются нулевыми. Тогда алгоритм оптимального приема выглядит как

$$\hat{i}_0 = \operatorname{Arg}\max_i \left\{ \sum_{i=1}^{M^R} p(\nu_i) \frac{w[r(t) | \nu_i]}{w[r(t) | w]} \right\}. \qquad (6)$$

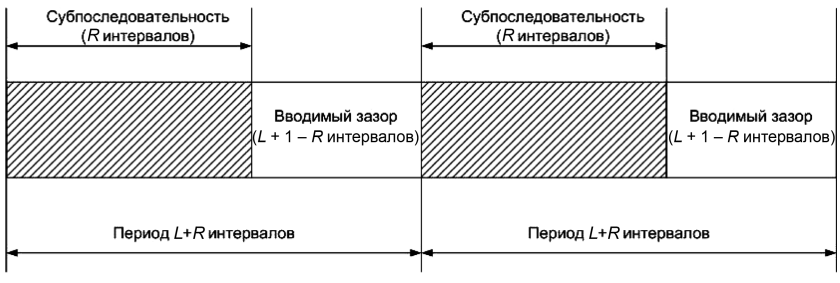


Рис. 1. Организация передачи с использованием субпоследовательностей

Для канала с квазибелым гауссовским стационарным шумом метрики могут быть определены как евклидовы следующим образом: $d^2(r,\ s_{\nu_i}) = \int_0^{RT} [r(t) - s_{\nu_i})t)]^2 dt$. При предположении передачи равновероятных субпоследовательностей выражение (6) принимает вид:

$$\hat{i}_0 = \operatorname{Arg}\min_i \left\{ \int_0^{RT} [r(t) - s_{\nu_i}(t)]^2 dt \right\}. \qquad (7)$$

Разработанный алгоритм характеризуется постоянной задержкой, причем длительность задержки меньше или равна длительности задержки при использовании АКН. Задержка при принятии решения о переданной субпоследовательности ν_i равна $T_{\text{зад}} = (R-1)T = (L-G)T$. Для полного использования энергии сигнала мы можем увеличить длительность анализа до LT, при этом возрастет помехоустойчивость по сравнению с АКН и алгоритмом Витерби. Согласно (6) вычислительная сложность разработанного алгоритма приема в целом может быть определена следующим образом:

$$C_2 \cong \frac{1}{R} M^R. \qquad (8)$$

Уменьшение вычислительной сложности алгоритма по сравнению с алгоритмом Витерби и АКН составляет

$$C_\Delta = \frac{C_1}{C_2} = \frac{M^{L+1}}{\frac{1}{R}M^R} RM^{L+1-R} = RV^{R(D-1)}. \qquad (9)$$

На рис. 2 представлен график данной зависимости для различных мощностей входного алфавита и значения $L = 10$.

Оценка параметров передачи проводилась с помощью оптимизации критериальной функции (10):

$$F(C_\Delta, S) = \frac{C_\Delta}{D} = \frac{RM^{R(D-1)}}{D} = \frac{\frac{L+1}{D} M^{\frac{L+1}{D}(D-1)}}{D} =$$

$$= \frac{(L+1)M^{\left(L+1-\frac{L+1}{D}\right)}}{D^2} = \frac{(L+1)M^{(L+1)}}{M^{\frac{L+1}{D}} D^2}, \qquad (10)$$

где S — функция скорости передачи, обратно пропорциональная скважности.

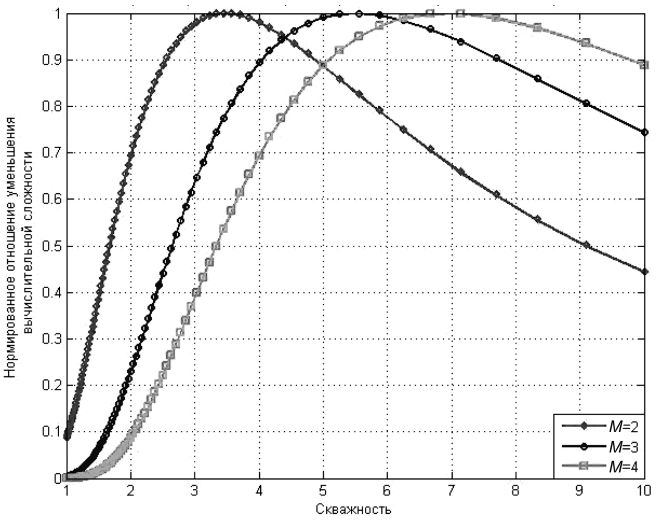


Рис. 2. Уменьшение вычислительной сложности приемника по отношению к скорости передачи

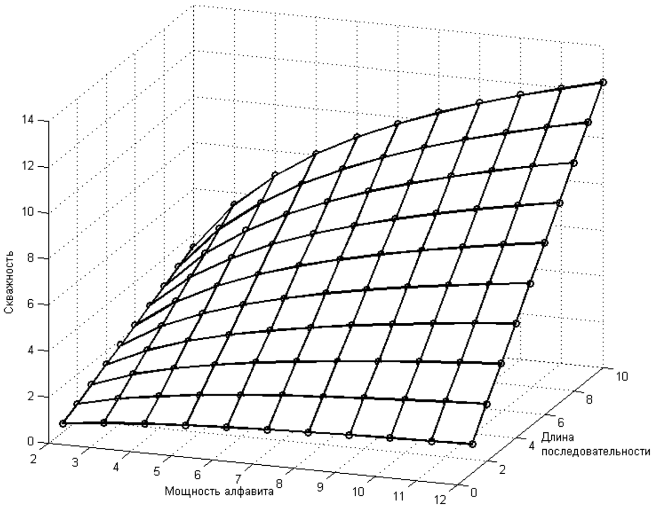


Рис. 3. Зависимость оптимального значения приведенной скважности для соотношения вычислительная сложность/скорость передачи от мощности алфавита передаваемых символов и периода последовательности

Оптимальное значение скважности по критерию вычислительная сложность к скорости определяется выражением

$$D = \frac{\ln M}{2}(L+1). \qquad (11)$$

Таким образом, выбор **оптимального значения скважности** зависит от мощности алфавита и числа интерферирующих компонентов в канале связи. То есть использование алфавитов малой мощности, а следовательно, простейших схем модуляции предпочтительно при данном способе организации передачи. Кроме того, физические свойства канала связи (длительность межсимвольной интерференции) накладывают дополнительные ограничения на определение оптимальных пара-

метров передачи. График зависимости оптимальной скважности от мощности алфавита и длины последовательности приведен на рис. 3.

Помехоустойчивость алгоритма МППП для канала связи с МСИ при использовании независимых субпоследовательностей

В общем случае оценка помехоустойчивости алгоритмов, реализующих критерий МППП, довольно сложна. Верхняя граница оценки вероятности ошибочного приема может быть описана следующим выражением [11]:

$$P_M \leqslant KQ\left[\frac{\overline{d(\varepsilon)}}{\sqrt{2N_0}}\right], \qquad (12)$$

здесь среднее расстояние $\overline{d(\varepsilon)}$ по Хеммингу может быть найдено из решения трансцендентного уравнения

$$Q\left[(2N_0)^{-\frac{1}{2}}\,\overline{d(\varepsilon)}\right] =$$

$$= \frac{1}{\textit{мощность}(S)} \sum_{\{S_{k+1} \in S\}} Q\left[(2N_0)^{-\frac{1}{2}}\, d(\varepsilon; \{S_{k+1}\})\right].$$

Рассмотрим вариант передачи с использованием независимых субпоследовательностей и оценим энергию запаздывающих лучей. При таком способе передачи вероятность ошибочного приема может быть описана как $P_M\left(\frac{E_s + \nu}{N_0}\right)$, где ν может быть рассмотрен как сдвиг отношения сигнал/шум за счет энергии запаздывающих лучей по отношению к классической непрерывной передаче (т. е. за счет увеличения длительности интервала анализа до LT). В реальных каналах связи с МСИ изменение средней мощности сигнала может быть аппроксимировано экспоненциальной функцией с коэффициентом затухания λ при передаче одиночных импульсов длительностью, много меньшей времени МСИ (L интервалов), и скважностью, большей или равной L; λ определяется физическими свойствами канала с замираниями и связана с длиной МСИ условием $\mathbf{e}^{-\lambda L} \to 0$.

Средняя энергия, получаемая при передаче одной субпоследовательности длительностью R интервалов при ведении непрерывной передачи

$$E_{\text{непр}} = \int_0^R \mathbf{e}^{-\lambda t} dt = \frac{1}{\lambda}(1 - \mathbf{e}^{-\lambda R}). \qquad (13)$$

И при передаче с использованием независимых субпоследовательностей

$$E_{\text{субпос}} = \int_0^L \mathbf{e}^{-\lambda t} dt = \frac{1}{\lambda}(1 - \mathbf{e}^{-\lambda L}) = \frac{1}{\lambda};$$

$$\mathbf{e}^{-\lambda L} \to 0. \qquad (14)$$

Здесь λ характеризует профиль канала и может быть оценен посредством нахождения длины МСИ. Сдвиг отношения сигнал/шум равен

$$\nu = \Delta E = E_{\text{субпос}} - E_{\text{непр}} = \frac{1}{\lambda}\mathbf{e}^{-\lambda R}. \qquad (15)$$

Таким образом, на основе выражений (12) и (15) для нелинейных каналов с ограниченной полосой верхняя граница вероятности ошибочного приема имеет вид:

$$P_M \leqslant KQ\left[\frac{\overline{d(\varepsilon, E_s)}}{\sqrt{2N_0}}\right] = KQ\left[\frac{\overline{d(\varepsilon, E_s)}}{\sqrt{2N_0}}\right] =$$

$$= KQ\left[\frac{\overline{d\left(\varepsilon, E_s + \frac{1}{\lambda}\mathbf{e}^{-\lambda R}\right)}}{\sqrt{2N_0}}\right]. \qquad (16)$$

Экспериментальные оценки качества разработанного алгоритма детектирования сигналов на основе моделирования приемопередающего тракта

Для количественной оценки была реализована модель физического уровня беспроводной высокоскоростной сети, работающей в области частот 3,1...10,6 ГГц. Точность оценки ИХ канала принималась равной текущему отношению сигнал/шум со сдвигом за счет усреднения шума при повторении тестовой последовательности и сос-

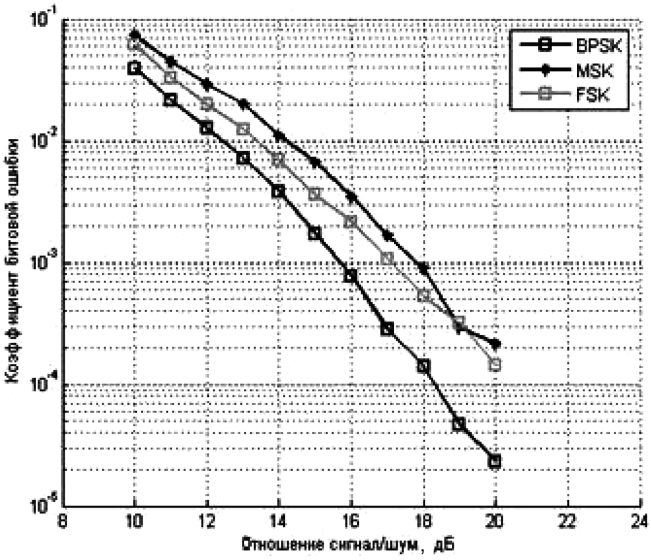


Рис. 4. Зависимость помехоустойчивости разработанной схемы приема от отношения сигнала/шум при использовании модели радиоканала СМ1. Неидеальная оценка ИХ канала

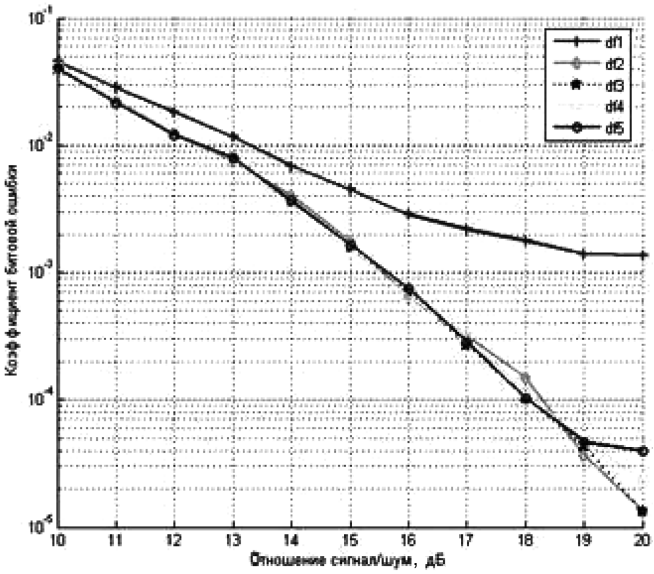


Рис. 5. Зависимость помехоустойчивости разработанной схемы приема от скважности. Модель канала CM1. Неидеальная оценка ИХ канала

тавляла 5 дБ. В значениях средней квадратичной ошибки погрешность была принята равной 0,15. Скважность передачи была выбрана равной 3, что позволяет сделать субпоследовательности независимыми и при этом оптимизировать критериальную функцию (10) при использовании двухпозиционной модуляции.

Был промоделирован один физический канал в полосе 4,6...5,2 ГГц, а также было выполнено моделирование в области основной полосы. Получены графики помехоустойчивости при использовании различных способов модуляции и параметров приемопередачи. В качестве моделей канала связи использовались теоретические модели канала связи, адекватно отражающие распространение сигналов в условиях помещений и рекомендованные комитетом IEEE: CM1, CM2, CM3, CM4 (CM — *Channel model*), с различными параметрами замираний и условий приема при наличии и отсутствии прямого луча [12]. На рис. 4 приведен график зависимости помехоустойчивости разработанного алгоритма от отношения сигнал/шум при использовании различных способов двухпозиционной модуляции (*binary phase shift keying* — бинарная фазовая модуляция, *minimum shift keying* — модуляция с минимальным сдвигом, *frequency shift keying* — частотная модуляция), модели канала CM1 и учетом аддитивного белого гауссовского шума. На рис. 5 приведен график зависимости помехоустойчивости для бинарной фазовой модуляции от отношения сигнал/шум для различных параметров скважности. Модель канала связи — CM1. Таким образом, минимальное значение скважности, при котором достижимо улучшение качества приема равно двум. Выбор данного параметра зависит от конкретных требований, предъявляемых к физическому уровню.

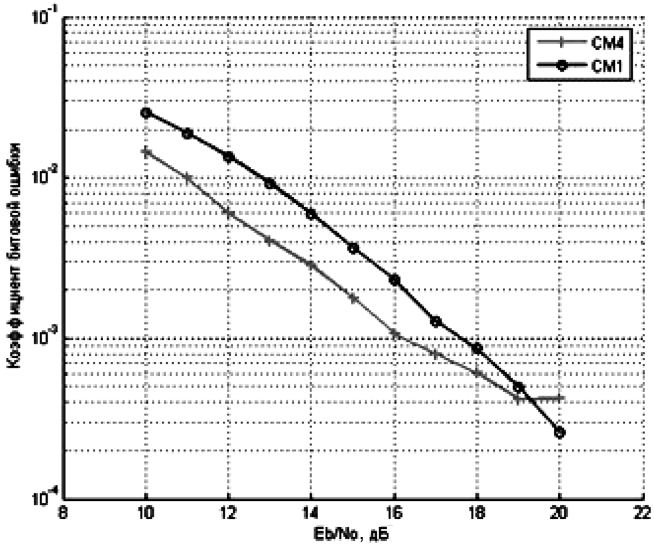


Рис. 6. Зависимость помехоустойчивости разработанного физического уровня от отношения энергия сигнала/спектральная плотность при использовании модели канала CM1, CM4. Идеальная оценка ИХ канала

Для сравнения качества разработанного метода приема с решением MB OFDM [13] были изменены параметры модели следующим образом: скважность выбрана равной 2, скорость передачи данных составила 400 Мбит/с на один физический канал; коэффициент частотной эффективности $B = 0{,}5$; был учтен шум аналого-цифрового преобразователя с разрядностью в 4 бита, импульсная характеристика канала считалась известной. График зависимости помехоустойчивости от отношения энергии сигнала к спектральной плотности мощности шума для CM1 и CM4 приведен на рис. 6. Сравнивая с системой MB OFDM [13], выигрыш при использовании разработанных способов организации передачи составляет 1 дБ в случае канала CM1 при значении коэффициента ошибок (КОШ), равном 0,01, и 3 дБ в случае канала CM4 при значении КОШ = 0,01. Увеличение выигрыша объясняется тем, что модель канала CM4 имеет существенно более выраженные многолучевые свойства, а разработанные методы приема используют энергию отраженных лучей.

Заключение

Для решения задачи помехоустойчивого приема в каналах с многолучевым распространением был разработан метод приемопередачи на основе субпоследовательностей. Сущность метода заключается в использовании независимых во временной области последовательностей символов простейших (бинарных) типов модуляции и орга-

низации приема в целом. Для поддержания необходимой скорости передачи возможно использование технологии быстрых скачков частоты, позволяющей компенсировать вводимые интервалы релаксации канала. В ходе теоретического анализа найдены оптимальные параметры передачи по критерию скорость/вычислительная сложность приема при различных мощностях используемого алфавита. С помощью моделирования были получены кривые помехоустойчивости для физического уровня. Сравнительный анализ с системой MB OFDM в средах с различным влиянием замираний показал преимущества разработанных методов и эффективность предложенной архитектуры.

Список литературы

1. **Federal** Communication Commission (FCC). 02-48 First Report and Order in the Matter of Revision of Part 15 of the Commission's Rules Regarding Ultra-wideband Transmission systems, adopted Feb. 14, 2002 [Электронный ресурс] / Federal Communication Commission. — Электрон. дан. — 2002. — http ://www. fcc.gov/.
2. **Найденов А.** Преобразование спектра наносекундного импульсного передатчика. — М.: Наука. — 1978. С. 10—18.
3. **Nicholson A. M.** Advances in subnanosecond pulse technology // Physics & Electronics Dept, Royal Radar Establishment, sem. — Malvern. —1972. — P. 5—20.
4. **Прокис Дж.** Цифровая связь: Пер. с англ. под ред. Д. Д. Кловского. — М.: Радио и Связь. — 2000. — 800 с.
5. **Schawartz M.** Information, Transmission, Modulation and Noise: / Second Edition // New York: MGraw-Hill. 1970. — 752 p.
6. **Теория** электрической связи: Учебник для вузов / Зюко А. Г., Кловский Д. Д., Коржик В. И., Назаров М. В.; под ред. Кловского Д. Д. — М.: Радио и связь. — 1999. — 432 с.
7. **Helstrom C. W.** Statistical Theory of Signal Detection [Текст] / Helstrom C. W. New York: Pergamon, 1960. — Sect. IV. 5.— 364 p.
8. **Кловский Д. Д.** Передача дискретных сообщений по радиоканалам: 2-е Изд. / Кловский Д. Д. — М.: Радио и связь. — 1982. — 304 с.
9. **Forney G. David.** Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference [Текст] / G. David Forney // IEEE Transactions of Information theory. — May 1972. Vol. IT-18. N. 3. P. 363—378.
10. **Viterbi A. J.** Convolution codecs and their performance in communication systems [Текст] / A. J. Viterbi // IEEE Trans. Commun. Technol. Oct. 1971. Vol. COM-19. P. 751—772.
11. **Dubey Vimak K.** Maximum Likelihood Sequence Detection for QPSK on Nonlinear, Band-Limited Channels [Текст] / Vimak K. Dubey // IEEE Transactions on Communications. December 1986. Vol. COM-34, N. 12. P. 1225—1235.
12. **Saleh A., Valenzueia R.** A Statistical Model for Indoor Multipath Propagation / A. Saleh and R. Valenzuela. // IEEE JSAC. Feb. 1987. Vol. SAC-5. — N. 2. — P. 128—137.
13. **Wessman Matts-Ola, Arne Svensson.** Performance of Coherent Impulse Radio and Multiband-OFDM on IEEE UWB Channels / Matts-Ola Wessman, Arne Svensson. — Communication System Group, Department of Signals and Systems: Gothenburg, Sweden, Chalmers University of Technology. — 2004. — P. 55.

УДК 004.421

С. А. Размахнин, аспирант, **А. И. Куприянов,** д-р техн. наук, проф.,
Московский авиационный институт (ГТУ)

Алгоритм разработки систем оперативно-розыскных мероприятий для сервисов, построенных на базе технологий мобильной связи

В России действует Федеральный Закон "О связи", согласно ему, любой сервис связи, с которым поставщик услуг выходит на российский рынок, должен быть интегрирован с системой оперативно-розыскных мероприятий (СОРМ). Расходы на интеграцию нового сервиса связи с системой СОРМ полностью ложатся на поставщика услуг связи. Это приводит к тому, что компании-операторы вынуждены самостоятельно внедрять для запускаемых сервисов технические комплексы, реализующие требования СОРМ, и решать проблемы, которые возникают при их создании.

Цель данной статьи — анализ тех проблем, с которыми сталкиваются технические специалисты операторов связи, занимающиеся вопросами разработки, внедрения и эксплуатации комплексов СОРМ, и обзор алгоритмов создания комплексов СОРМ для сервисов, построенных на базе технологий мобильной связи.

Ключевые слова: *система оперативно-розыскных мероприятий, законный перехват, мобильная связь.*

Введение

В настоящее время все большую популярность начинают приобретать сервисы мобильной связи, созданные на базе технологий *General Packet Radio Service* (GPRS), *Enhanced Data rates for GSM Evolution* (EDGE), *Universal Mobile Telecommunications System* (UMTS). Среди таких сервисов можно выделить сервисы "мгновенного" обмена сообщениями, сервисы корпоративной почты, сервисы электронного документооборота. Все эти сервисы создаются исходя из концепции "мобильный офис", при которой абонент (пользователь услуги) со своего мобильного терминала (телефона, смартфона или ноутбука с GPRS/EDGE модемом) может получить доступ к корпоративной почте, работать с документами, размещенными в корпоративной сети, вести переписку с коллегами.

В России действует Федеральный Закон "О связи", согласно которому, любой сервис связи, с которым поставщик услуг выходит на российский рынок, должен быть интегрирован с системой оперативно-розыскных мероприятий (СОРМ). В нашей стране, в отличие от других стран, разработка, внедрение и эксплуатация комплекса СОРМ для нового сервиса ложится на оператора связи [1, 2]. Это приводит к тому, что компании-операторы связи вынуждены самостоятельно разрабатывать для внедряемых сервисов технические комплексы, реализующие требования СОРМ, и решать технические проблемы, которые возникают при их создании.

Проблемы, которые возникают при разработке комплексов СОРМ

Типовая архитектура сервиса, основанного на технологиях мобильной связи GPRS/EDGE/UMTS, представлена на рис. 1, где 2.5—3 G Mobile Network — сеть оператора мобильной связи, построенная на базе технологии GPRS, EDGE, UMTS, используя которую абонент получает доступ к сети Интернет или к своей корпоративной сети; Relay Server — сервер-посредник, осуществляющий взаимодействие между мобильным терминалом абонента и ресурсом, которым пользуется данный абонент в своей корпоративной сети или в сети Интернет; Internet/Corporate Network — сеть Интернет или корпоративная сеть пользователя; Connector — программное обеспечение Relay Server, осуществляющее взаимодействие между мобильным терминалом абонента и ресурсом, которым пользуется данный абонент; Services (Mail Servers, Databases, Terminal Servers, Web Servers) — ИТ-сервисы (почтовые серверы, базы данных, терминальные серверы, веб-серверы).

В общем случае механизм работы услуги выглядит следующим образом. Пользователь с помощью терминала, на который установлена программа-клиент, или с помощью специализированного терминала, ориентированного только на данную услугу (далее клиентскую программу), соединяется с сервером-посредником, который, в свою очередь, выполняет запросы пользователя в корпоративной сети или в сети Интернет.

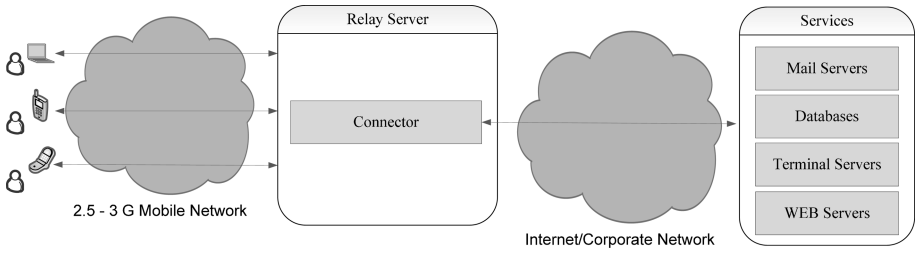


Рис. 1. Архитектура реализации услуги, базирующаяся на технологии передачи данных 2.3—3 G

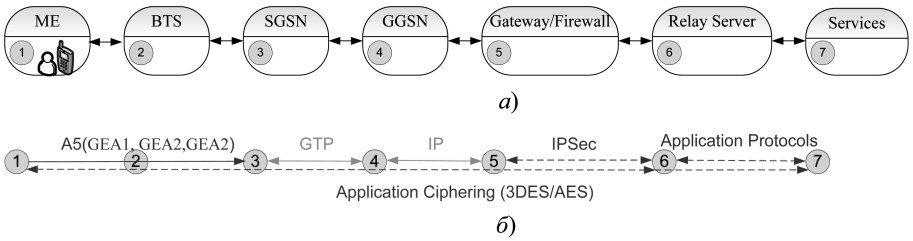
Существующие комплексы СОРМ для фиксированной и мобильной связи установлены и подключены к компонентам сети, показанным на рис. 2 (см. третью сторону обложки), следующим образом.

Комплекс СОРМ GPRS получает копию сетевого трафика, который передается между узлами SGSN (*Serving GPRS Support Node* — узел обслуживания абонентов GPRS) и GGSN (*Gateway GPRS Support Node* — узел, обеспечивающий маршрутизации данных GPRS-комплекса) GPRS комплекса оператора, и предназначен для контроля за клиентами оператора, пользующимися услугами мобильной связи. Копия сетевого трафика снимается с коммутатора, к которому подключены узлы SGSN и GGSN, посредством технологии SPAN. Съем информации комплексом СОРМ GPRS с данных интерфейсов объясняется тем, что трафик GPRS после узла SGSN не зашифрован, поскольку шифрование данных в технологиях GPRS/EDGE по алгоритму GEA1, GEA2,GEA2 осуществляется только на маршруте от пользовательского терминала (ME) до узла коммутации SGSN. Кроме того, трафик GPRS после узла SGSN содержит всю информацию, получаемую и передаваемую пользователем совместно с идентификаторами терминала (IMEI) и пользователя (IMSI, MSISDN).

В свою очередь, комплекс СОРМ Интернет получает копию сетевого трафика, который поступает на вход (внутренний интерфейс) шлюза сети оператора связи и предназначен для контроля клиентов и сотрудников оператора, использующих сеть оператора связи для выхода в Интернет. Съем информации комплексом СОРМ Интернет с данного интерфейса обусловлен тем, что сетевой трафик до прохождения шлюза не содержит транслированных (внешних) адресов оператора, назначенных клиентам, а содержит внутренние адреса, по которым этих клиентов можно однозначно идентифицировать.

Комплексы СОРМ GPRS и СОРМ Интернет были разработаны для контроля действий абонентов, использующих услугу выхода в сеть общего доступа (Интернет, корпоративная сеть) с помощью технологий мобильной и фиксированной связи.

Основной причиной, по которой эти существующие комплексы оказались не способны осуществлять контроль данных, передающихся с использованием новых сервисов, построенных на базе технологий GPRS/EDGE, стало то, что во всех современных сервисах используется шифрование дан-



а)

б)

Рис. 3. Схема взаимодействия функциональных компонентов сервиса

ных с использованием западных алгоритмов AES/3DES или с помощью алгоритмов собственной разработки. Схема механизмов шифрования, которые используются между функциональными компонентами сервиса, приведена на рис. 3.

К поставщику услуги связи предъявляется требование, согласно которому комплекс СОРМ должен обеспечить возможность съема информации, передаваемой и принимаемой любым конкретным пользователем в процессе предоставления любых услуг [3, 4]. Полностью снять шифрацию поставщик сервиса не может, поскольку ввиду использования сервисом на отдельных участках взаимодействия публичных сетей (сети Интернет, сетей провайдера, корпоративной сети) это сделает применение сервиса небезопасным. Поэтому поставщик услуги связи вынужден разрабатывать интерфейс, который бы позволял снимать в открытом виде всю передаваемую информацию посредством этого сервиса.

При этом поставщик услуги сталкивается со следующими проблемами:

- обеспечение снятия в открытом виде всей информации, передаваемой посредством данного сервиса и блокировки тех сообщений, которые не могут быть проконтролированы;
- обеспечение однозначного определения принадлежности сообщений тем идентификаторам клиента, которые использованы при его регистрации на данном сервисе и тем идентификаторам мобильного терминала и сим-карты, которые использует клиент.

Первым важным фактором, который влияет на разработку системы оперативно-розыскных мероприятий для сервиса связи является готовность разработчика услуги сотрудничать с оператором связи по данным вопросам или его отказ от сотрудничества. Второй важный фактор — протяженность использования прикладного шифрования: на всем пути обработки информации (от мобильного терминала до ИТ-сервиса) или данные зашифровываются только на участке от терминала до сервера-посредника (Relay Server). Третьим важным фактором является принадлежность серверов, на которых работают ИТ-сервисы, оператору или клиенту услуги. Именно от этих факторов зависит выбор точки съема информации и структура компонентов разрабатываемого комплексом СОРМ.

Выбор точки съема информации

Съем информации для комплекса СОРМ возможен с трех компонентов сервиса:

- мобильный терминал пользователя (ME — *Mobile Equipment*);
- сервер-посредник (*Relay Server*);
- серверы, на которых работают ИТ-сервисы (почтовые серверы, базы данных, терминальные серверы, веб-серверы):

Мобильный терминал клиента. Использование мобильного терминала клиента в качестве точки съема информации предполагает установку на пользовательский терминал дополнительного программного обеспечения (программы-агента), обеспечивающего перехват информации, с которой работает пользователь.

Это решение самое сложное в эксплуатации, поскольку число пользователей современных сервисов исчисляется тысячами и десятками тысяч, и это приводит к тому, что комплекс СОРМ начинает содержать такое же число компонентов, которые необходимо обслуживать, поддерживать, решать проблемы, связанные с их эксплуатацией, притом делая это скрытно от пользователя. Однако в том случае, если сервис использует прикладную шифрацию на всем участке своей работы (от терминала до конечного ИТ-сервиса) и разработчик сервиса отказывается от сотрудничества, то это решение оказывается единственно возможным.

Сервер-посредник. Использование сервера-посредника в качестве точки съема информации возможно в двух случаях.

Первый случай предполагает, что разработчик услуги готов доработать программное обеспечение сервера так, чтобы ключи шифрации, которые используют пользователи, и шифрованные сообщения сохранялись в отдельных промежуточных хранилищах и с помощью дополнительного разработанного функционала зашифрованные сообщения дешифровались с помощью пользовательских ключей и передавались на комплекс СОРМ.

Второй случай предполагает, что разработчик услуги отказывается от сотрудничества, прикладное шифрование данных обеспечивается на пути от мобильного терминала пользователя до сервера-посредника, и интерфейс обмена информацией между сервером-посредником и серверами, на которых работают ИТ-сервисы, находится под контролем оператора связи. В современных сер-

висах используется один—два сервера-посредника, таким образом, использование сервера-посредника в качестве точки съема информации позволяет создать единую точку контроля, что облегчает процесс эксплуатации и поддержки комплекса СОРМ и является оптимальным вариантом для его построения.

Серверы, на которых работают ИТ-сервисы (почтовые серверы, базы данных, терминальные серверы, веб-серверы). Использование серверов, на которых работают ИТ-сервисы, в качестве точки съема информации предполагает, что разработчик сервиса отказывается от сотрудничества с оператором, прикладное шифрование данных обеспечивается на пути от мобильного терминала пользователя до сервера-посредника, и использование сервера посредника в качестве точки съема информации невозможно по каким-либо причинам (например, выполнения сервера-посредника в виде единого аппаратно-программного комплекса, встраивание в работу которого сторонних компонентов приведет к нестабильной работе сервиса). Число серверов, на которых работают ИТ-сервисы услуги, может исчисляться десятками и сотнями, поэтому их использование не позволяет создать единую точку контроля и целесообразно только в случае невозможности проводить съем информации с сервера-посредника.

Определение принадлежности сообщений

Задача определения принадлежности сообщений заключается в однозначном сопоставлении каждого сообщения идентификаторам той учетной записи, которая была создана при регистрации пользователя в сервисе, идентификаторам мобильного терминала и сим-карты, используемым клиентом, ФИО и паспортным данным клиента.

Примером идентификатора учетной записи пользователя может служить название учетной записи пользователя или почтового ящика пользователя. Примером идентификатора терминала пользователя служит IMEI (*International mobile equipment identity* — международный идентификатор мобильного оборудования), а примером идентификаторов сим-карты — IMSI (*International mobile subscriber identity* — международный мобильный пользовательский идентификатор) и MSISDN (*Mobile station international ISDN number*(*s*) — телефонный номер абонента).

Решение задачи определения принадлежности осложняется тем, что спецификация и подробное описание интерфейса точки съема информации, а соответственно, и точный перечень идентификаторов абонента, которые нужно сопоставлять сообщениям сервиса, для систем документальной электросвязи (СДЭС) отсутствует.

Такое описание, а также описание механизмов взаимодействия систем связи и комплексов СОРМ должно приводиться в частных технических требованиях (ЧТТ) на комплексы СОРМ для данных видов услуг. Однако такие ЧТТ разработаны и утверждены только для услуг фиксированной и подвижной связи. Для систем документальной электросвязи (сервисов связи) такие частные технические требования не утверждены до сих пор, несмотря на то, что общие технические требования к СДЭС были разработаны и утверждены еще в марте 1999 года приказом Государственного комитета Российской Федерации по связи и информатизации № 47 "Об утверждении общих технических требований к системе технических средств по обеспечению функций оперативно-розыскных мероприятий на сетях (службах) документальной электросвязи".

Для каждого конкретного сервиса оператор вынужден согласовывать с государственным регулятором требования к интерфейсу источника съема информации отдельно. После согласования с государственным регулятором набора идентификаторов, которые необходимо сопоставлять с сообщениями сервиса, и формата сообщений, передаваемых на комплекс СОРМ, оператор вынужден самостоятельно реализовать эти требования.

Процедура их реализации выглядит следующим образом:

- заведение информации о пользователях при их регистрации на сервисе;
- разработка программного обеспечения, которое к перехваченным сообщениям добавляет идентификаторы пользователя в системе, идентификаторы его мобильного терминала и сим-карты, полученные из базы данных, заведенной при регистрации пользователя.

При работе пользователь может сменить сим-карту в своем терминале, перерегистрировать свой терминал на другую учетную запись в сервисе или выполнить другие действия, которые могут нарушить актуальность информации, собранной при регистрации пользователя в системе. Поэтому должна быть решена задача поддержания информации о пользователях и их идентификаторах в актуальном состоянии. Эта задача может быть решена двумя способами.

Первый способ заключается в доработке программы-клиента и программного обеспечения сервера-посредника таким образом, что будет выполнено одно из условий:

- реализован запрет на выполнение пользователем действий, которые бы могли привести к смене связки идентификаторов его терминала, сим-карты и учетной записи в системе;
- реализована процедура, которая при выполнении пользователем действий, приводящих к

смене связки идентификаторов, будет обновлять соответствующую информацию в базе данных, созданной при регистрации пользователей.

Второй способ заключается в разработке специального программного или аппаратного модуля, который бы мог анализировать трафик и отслеживать выполнение пользователем действий, приводящих к смене связки идентификатора его учетной записи в системе, идентификаторов терминала и сим-карты. Единственным участком сети, где в сетевом трафике одновременно присутствуют все данные идентификаторы, является участок между узлами SGSN и GGSN GPRS комплекса оператора. Извлечение из каждой пользовательской сессии связки идентификаторов IMSI, MSISDN, IMEI является простой задачей, поскольку данные идентификаторы передаются при каждом запросе на создание GPRS/EDGE соединения. Однако соотнесение этих идентификаторов с идентификатором сервисной пользовательской учетной записи возможно лишь в двух случаях:

- если программное обеспечение сервиса написано таким образом, что при каждом сеансе связи в ИТ-сервисе указывается связка сервисной пользовательской учетной записи и одного из перечня идентификаторов терминала или сим-карты (IMSI, MSISDN, IMEI) и эта информация может быть извлечена без нарушения работы сервиса;
- если название пользовательской учетной записи может быть извлечено из прикладного уровня сетевых пакетов. Все сервисы мобильной связи, построенные на базе технологий GPRS/EDSE, шифруют передаваемые данные прикладного уровня, однако информация, необходимая для маршрутизации пакетов зачастую не шифруется и среди этой информации может быть извлечено и название пользовательской учетной записи. Однако такой подход будет протоколозависимым и в случае смены прикладного протокола задачу поддержания информации, необходимой для персонализации сообщений, в актуальном состоянии придется решать повторно.

Таким образом, процедура интеграции нового сервиса в систему оперативно-розыскных мероприятий выглядит следующим образом:

- определение точки съема информации;
- согласование с государственным регулятором требований к идентификаторам пользовательского терминала, сим-карты и пользовательской учетной записи в сервисе, которыми необходимо персонализировать все передаваемые и принимаемые сообщения;
- обеспечение снятия в открытом виде всей информации, передаваемой посредством данного сервиса и блокировки тех сообщений, которые не могут быть проконтролированы;
- обеспечение однозначной персонализации сообщений, согласно утвержденным требованиям;
- поддержание информации, необходимой для персонализации сообщений, в актуальном состоянии.

Как было показано выше, интеграция нового сервиса связи в систему оперативно-розыскных мероприятий — это сложный, технический процесс, который может быть решен эффективно только при тесном взаимодействии оператора связи, разработчика услуги и государственного регулятора. В случае неэффективной работы или отказе от тесного сотрудничества хотя бы одного из участников процесса интеграция сервиса в систему СОРМ может существенно затянуться или вовсе не быть выполнена.

Список литературы

1. **Requirements** of Law Enforcement Agencies: Technical Specifications. ETSI TS 101 331 V1.2.1.06.2006.

2. **Concepts** of Interception in a Generic Network Architecture: Technical Report. ETSI TR 101 943 V2.2.1.2006.11.

3. **Технические** требования к узлам телематических служб и передачи данных для обеспечения проведения оперативно-розыскных мероприятий. Утверждены Министерством связи Российской Федерации 10.07.2003 г.

4. **Технические** требования к устройству системы технических средств по обеспечению функций оперативно-розыскных мероприятий на узлах телематических служб и передачи данных. Утверждены Министерством связи Российской Федерации 10.07.2003 г.

Новые книги

Васенин В. А., Водомеров А. Н., Конев И. М., Степанов Е. А. Т-подход к автоматизированному распараллеливанию программ: идеи, решения, перспективы / Под ред. академика РАН В. А. Садовничего. — М.: МЦНМО, 2008. — 358 с.

В книге изложен Т-подход к созданию систем автоматизированного динамического распараллеливания программ, включающий императивные и функциональные механизмы программирования. Такой подход представляет интерес для некоторых классов практически значимых вычислительных приложений, обладающих свойством, которое в контексте данной книги именуется "внутренним динамическим параллелизмом". Особенностью представленных в ней материалов является взаимодополняющее сочетание строгих математических моделей и алгоритмов, современных языков программирования и архитектурно-технологических решений, которые используются в процессе разработки, реализации и модификации программного кода сложной системы. Их содержание имеет не только исследовательский характер, но и несет явную учебно-образовательную функцию.

Книга полезна специалистам в области разработки и создания математического и программного обеспечения для высокопроизводительных вычислительных систем, а также как учебное пособие для студентов и аспирантов, изучающих эти вопросы в университетах.

УДК 004.3.015 + 658.014.1 + 004.8(3)

П. М. Евграфов, канд. техн. наук, нач. цикла,
Академия ГПС МЧС России,
ГОУ ДПО "Подольский учебный центр ФПС",
e-mail: indelig@mail.ru

Метод структурирования, представления и логического оценивания "неидеальных" знаний-решений

Рассмотрен метод структурирования и моделирования "неидеальных" знаний, характеризующихся недостатком истинной информации, наличием неверной или ненужной сопутствующей информации, нечеткостью и противоречивостью. Описанный подход применялся в системах интеллектуальной поддержки решений и в компьютерных обучающих системах.

Ключевые слова: *неидеальные знания, сложные знания, структурирование знаний, логическая оценка знаний, правильная часть знаний, неправильная часть знании, нечеткая часть знания, противоречивая часть знания, метод психологического моделирования сложных знаний.*

Введение

Задача повышения качества принимаемых решений через изыскание информационных технологий, более эффективных для достижения требуемого результата, актуальна во многих областях деятельности человека.

Решение всегда может быть рассмотрено как знание, знание же не всегда выражает собой некую практическую цель, присущую решениям. В этом смысле понятие "*решение*" шире понятия "*знание*". Под термином *знание-решение*, употребленным в названии статьи, мы подразумеваем информационную структуру, в которой отражается *семантика* некоего знания, описываемого определенным *понятием*, составленным в общем случае из произвольного числа исходных понятий, и/или отражается практическая *цель*, достигаемая с помощью соответствующего этому знанию решения. "*Неидеальными*" знаниями мы называем *реальное* информационное пространство, описывающее конкретный объект, характеризующееся недостатком истинной информации, наличием неверной или ненужной сопутствующей информации, нечеткостью и противоречивостью. Под "*идеальным*" *знанием* мы понимаем не философскую категорию абсолютной и недостижимой истины, а знание на сегодняшний день науки и практики, которому (независимо от оценивающего знание субъекта) можно поставить однозначно в соответствие определенный объект. Решение, принимаемое на основе "неидеальных" знаний, также является "*неидеальным*" знанием. Поэтому практический эффект от подобных решений зачастую отличается от требуемого или ожидаемого эффекта.

Повышение качества принимаемых решений, основанных на "*неидеальных*" *знаниях*, может достигаться посредством наиболее подробного *структурирования* и *моделирования* исследуемых объектов (знаний) с последующей их *оценкой*, хотя бы на логическом уровне.

Особенно ярко (из-за недостатка или отсутствия объективных числовых оценок) проблема "неидеальности" знаний проявляется в социальных областях принятия решений, управление в которых преимущественно основано на *речевой* (*текстовой*) *информации*. Насколько известно, первым, кто занялся структурированием и моделированием в текстовой информации, был Аристотель, предложивший *формальную логику* как науку строить правильные высказывания. Формальная логика не утратила своей актуальности и по сей день, в чем немало содействует ее *понятийность*, однако такие свойства классической логики, как непротиворечивость, требования от знаний *идеальности* (в том смысле, что мы определили выше) определяют естественные границы ее применимости. Реальное же человеческое мышление способно каким-то образом оперировать с "неидеальными" знаниями, принимая решения двухмерного логического формата. Поэтому в контексте создания различных компьютерных систем закономерна постановка задачи разработки математического аппарата некой "*реальной*" *логики*, более близко отражающей особенности человеческого мышления в сравнении с формальной логикой.

Введение Л. Заде [1] понятия *нечеткость*, под которой понимается свойство плавного, без резких границ, перехода признаков одного объекта в признаки другого объекта, а также разработка им аппарата, получившего название *нечеткой логики*, представляет собой движение в направлении повышения объективности оценок, основанных на "неидеальной" информации. Однако данный ап-

парат введением в нем "*функции нечеткости* (*принадлежности*)" стал аппаратом не логической, а численной (в самом широком смысле) оценки степени принадлежности объекта к тому либо иному классу объектов. К тому же, нечеткость, понимаемая с позиций Л. Заде, является лишь одним из возможных источников "неидеальности" знаний. Таким образом, с введением аппарата нечеткой логики задача дальнейшего структурирования, моделирования и оценки "неидеальных" знаний не утратила своей актуальности.

Структурирование и представление "неидеальных" знаний

В нескольких работах, например, [2] (наиболее подробно) и [3] (наиболее доступно в смысле возможности ознакомления) излагаются принципы так называемого *психологического моделирования сложных знаний*, предлагающего способ структурирования, представления и логического оценивания "неидеальных" сложных знаний. Употреблением термина *сложное знание* подчеркивается, что знание рассматривается не как нечто целое и неделимое, а как многоэлементная информационная структура, отдельные части которой находятся во взаимной связи и которой оперирует мышление (интеллект).

В *методе психологического моделирования* (МПМ) рассматриваются и логически оцениваются не только логические структуры сложного знания (СЗ), но и *комбинаторные* структуры СЗ, в которых важна *последовательность связи* между элементами, составляющими СЗ.

Термин *психологическое моделирование* (в техническом смысле) ввел Рейтман [4], трактуя его как набор интеллектуальных операций над исходным знанием по преобразованию его в новый интеллектуальный продукт. Мы дополнительно к его трактовке понимаем под этим термином приближение к реальным мыслительным процессам на уровне некой "*психологики*".

По прошествии времени в связи с расширением круга рассматриваемых прикладных задач были скорректированы некоторые формулировки МПМ. Структурирование СЗ стало более подробным за счет рассмотрения элементов СЗ, обладающих дополнительными свойствами. Предлагаемая ниже краткая версия МПМ является наиболее новой.

В табл. 1 приведены перечни моделей (форм) сложных знаний-решений, используемых в МПМ, и дана их краткая характеристика. Модели учитывают кроме *правильных* элементов знаний также элементы, придающие сложному знанию "неидеальность".

Таблица 1

Перечень моделей сложных знаний-решений

Модель	Описание
ПУ	Перечень Условий для достижения цели-понятия (близкий аналог логики И)
ПА	Перечень Альтернатив достижения цели-понятия (близкий аналог логики ИЛИ, может приближаться к форме ПУ)
ПД	Последовательность действий для достижения цели-понятия (описывает последовательное во времени выполнение операций)
А-ПУ	Альтернативные Перечни Условий достижения цели-понятия (комбинация ПА, наложенного на ПУ)
А-ПВ	Альтернативы достижения цели-понятия, указанные в решении в порядке Предпочтительности их Выбора (моделирует интуитивный выбор альтернатив и их взвешивание)
А-ПД	Альтернативные Последовательности Действий (комбинации ПА, наложенного на ПД)
А-ПУ-ПВ	Альтернативные Перечни Условий, указанные в решении в порядке Предпочтительности Выбора (комбинация А-ПВ, наложенного на ПУ)
А-ПД-ПВ	Альтернативные Последовательности Действий, указанные в решении в порядке Предпочтительности их Выбора (комбинация А-ПВ, наложенного на ПД)

Чисто *логическими моделями* СЗ являются формы ПУ, ПА и А-ПУ, чисто *комбинаторными моделями* СЗ — формы ПД и А-ПВ, остальные модели носят *комбинированный* характер.

Каждая из моделей сложного знания представляет собой функцию, параметрами которой являются элементы сложных знаний, указанные в прилагаемом к каждой модели *перечне* пронумерованных *частей знаний*.

Некое произвольное решение задачи можно записать, например, в следующем виде (индекс у типа психологической формы соответствует номеру *перечня частей знания* для этой формы):

$$P = ПУ_3(1, 9, 12, 4) \rightarrow ПД_1(2, 9, 1, 4) \rightarrow$$
$$\rightarrow ПА_4(2, 5, 6) \rightarrow А\text{-}ПУ_2(4, 2, 3)...$$

В перечнях могут присутствовать не только правильные элементы сложного знания, но и неправильные, и противоречивые *части знаний* (ЧЗ), которые в свою очередь могут быть нечеткими.

В МПМ используется термин *цель-понятие* — категория описания знаний, в которой заключен смысл данного знания, выражающийся в достижении некой практически полезной для индивида (общества) цели и/или в достижении классифицирующего это знание понятия.

СЗ состоит из слагающих его ЧЗ, каждая из которых имеет собственную цель-понятие. Максимальные размеры СЗ, как и размеры ЧЗ, не определены, они могут быть сколь угодно большими. Формат СЗ и ЧЗ как информационных единиц

Таблица 2

Состав частей знаний базовых форм СЗ

Форма	Части сложных знаний												
	Н	АН	ПН	ДН	УДН	П	ОП	НП	УНП	НЧ	ПР	ЖС	ДПП	ДПЭ
ПА	+					+				+	+			
ПУ	+	+	+	+	+	+	+	+	+	+	+			
ПД	+	+	+	+	+	+	+	+	+	+	+	+	+	+

может быть любым: текстовым, графическим, звуковым, цифровым. Независимо от формата и размера ЧЗ в мозгу оперирующего им индивида складывается *образ* данного ЧЗ, отражающий его цель-понятие. Любое описание данного образа в виде символа (например, знака интеграла), рисунка, текстового массива мы считаем *понятием*.

В табл. 2 приведены разновидности ЧЗ, рассматривающиеся в первичных базовых формах СЗ: ПУ, ПА и ПД. Описание этих разновидностей ЧЗ приводится ниже.

Неправильными элементами (*Н*) мы называем части знания, которые делают невозможным достижение требуемой цели-понятия данного СЗ, частью которых они являются, либо цели-понятия иного сложного знания, необходимого в контексте решаемой задачи; также ими могут признаваться ЧЗ, делающие СЗ неоптимальным для достижения цели-понятия.

Активная неправильная (*АН*) *ЧЗ* — неправильная ЧЗ, указание которой в знании приводит к невозможности достижения его цели-понятия.

Пассивная неправильная (*ПН*) *ЧЗ* — неправильная ЧЗ, указание которой в знании не влияет на достижение его цели-понятия. Подобные ЧЗ являются по сути лишними в СЗ, тем самым, придавая знанию-решению неоптимальность.

Допустимая неправильная (*ДН*) *ЧЗ* — неправильная ЧЗ, указание которой в знании приводит к уменьшению ожидаемого положительного эффекта от СЗ, но в рамках, допустимых для достижения его цели-понятия.

Условно допустимая неправильная (*УДН*) *ЧЗ* — допустимая неправильная ЧЗ, допустимость которой ставится в зависимость от присутствия в ответе других допустимых неправильных ЧЗ.

Правильная (*П*) *ЧЗ* — элемент сложного знания, наличие которого в нем содействует достижению его цели-понятия.

Обязательная правильная (*ОП*) *ЧЗ* — правильная ЧЗ некоего сложного знания, отсутствие которой приводит к недостижению цели-понятия этого сложного знания.

Необязательная правильная (*НП*) *ЧЗ* — правильная ЧЗ некоего сложного знания, вносящая свой "положительный" вклад в достижение цели-понятия данного сложного знания, отсутствие которой в знании приводит к уменьшению положительного эффекта от него, но в размерах, приемлемых в рамках достижения цели-понятия.

Условно необязательная правильная (*УНП*) *ЧЗ* — необязательная правильная ЧЗ, необязательность которой ставится в зависимость от присутствия в ответе других необязательных правильных ЧЗ.

Нечеткая (*НЧ*) *часть знания* — это ЧЗ, которой затруднительно поставить в однозначное соответствие объект реального мира вследствие либо нечеткого характера самого объекта, либо наших нечетких представлений о нем, либо ограниченных возможностей языка. Нечеткими могут быть любые ЧЗ — как правильные, так и неправильные.

Противоречивая (*ПР*) *ЧЗ* — это ЧЗ, оказывающая на достижение цели-понятия данного или иного необходимого в контексте решаемой задачи сложного знания точно не определенное, частично положительное, частично отрицательное влияние.

В *комбинаторных* видах сложного знания кроме перечисленных выше разновидностей ЧЗ имеются и другие ЧЗ.

Правильные части сложного комбинаторного знания (решения), для достижения цели-понятия которого последовательность следования их друг за другом не может быть изменена, называются *правильными частями знания жесткой структуры* (*ЖС*).

Определим *действиями произвольной последовательности* (*ДПП*) те элементы решения, изменение позиций которых внутри некоего *диапазона действий произвольной последовательности* (*ДДПП*) между некими элементами *жесткой структуры* либо не оказывает влияния, либо оказывает незначительное влияние на достижение цели-понятия.

Определим как *действие плавного эффекта* (*ДПЭ*) элемент знания-решения, изменение позиций которого внутри некоего *диапазона действий плавного эффекта* (*ДДПЭ*) относительно оптимальной позиции приводит к уменьшению эффективности решения, но это уменьшение эффекта не выходит за рамки минимально допустимого.

Логическое оценивание

Логическое оценивание достижения цели-понятия СЗ, имеющего в своем составе нечеткую ЧЗ, может проходить в следующих направлениях.

Первое направление предполагает возможность структурирования нечеткой ЧЗ и представления ее в виде логических форм СЗ: ПУ или ПА.

Если нечеткая ЧЗ представляется в виде формы ПУ, то необязательно, но может сложиться ситуация, когда *общая часть* вновь полученных ЧЗ будет иметь вполне четкий характер, т. е. от исходной нечеткой ЧЗ можно перейти к четкой ЧЗ. Обозначим исходную нечеткую ЧЗ как $\sim ЧЗ_0$, тогда допустим, что после соответствующего структурирования:

$$\sim ЧЗ_0 = ПУ(\sim ЧЗ_1; ЧЗ_2, \sim ЧЗ_3).$$

Пусть $ЧЗ_2$ есть общая часть для $\sim ЧЗ_1$ и $\sim ЧЗ_3$ и она является четкой. Тогда

$$\sim ЧЗ_0 = ПУ(ЧЗ_2, ЧЗ_2, ЧЗ_2) = ПУ(ЧЗ_2) \equiv ЧЗ_2.$$

Если нечеткая ЧЗ представляется в виде формы ПА и имеет в своем составе как нечеткие, так и четкие ЧЗ, то может быть приемлемым исключение из формы ПА ее нечетких компонентов. Полученное после сокращения нечетких компонентов СЗ формы ПА будет неэквивалентно исходному СЗ по числу возможных альтернатив, но оно будет достигать цель-понятие исходного знания. Так, если $\sim ЧЗ_0 = ПА(\sim ЧЗ_1, ЧЗ_2, ЧЗ_3)$, то можно осуществить преобразование $\sim ЧЗ_0 \approx ПА(ЧЗ_2, ЧЗ_3)$.

При невозможности структурирования нечеткой ЧЗ оценка достижения цели-понятия проводится на основе субъективного мнения оценивающего лица.

Противоречивые ЧЗ характеризуются нечеткостью их вклада в достижение цели-понятия, который не может быть оценен в целом ни как положительный, ни как отрицательный. Логическое оценивание противоречивых ЧЗ строится по тем же принципам, что и для нечетких ЧЗ (возможное структурирование и преобразование).

Алгоритмизация логического оценивания для каждой психологической формы сложного знания-решения обеспечивается введением *матрицы образа решения*, в которой описывается структура образцового решения (знания), характеризуются все ЧЗ перечня, приложенного к данной форме.

Для описанных психологических форм знания были получены модели логического оценивания достижения цели-понятия (логической ценности решения). Далее приводятся модели логического оценивания сложного знания базовых психологических форм: ПА, ПУ, ПД. Нечеткие и противоречивые ЧЗ в данных моделях рассматривать нецелесообразно вследствие специфичности операций структурирования и преобразований для каждого конкретного СЗ.

Логическая модель формы ПА

В табл. 1 отмечено, что форма ПА, являясь аналогом логики ИЛИ, по своим свойствам может приближаться к свойствам формы ПУ. Это означает, что кроме традиционного восприятия логики ИЛИ (строгой и нестрогой) форма ПА отражает такой мыслительный процесс, когда требуется представить *все возможные альтернативы* достижения цели-понятия. Такой случай соответствует планированию возможных альтернативных решений для достаточно неопределенной будущей ситуации. При этом логическая оценка достижения цели-понятия СЗ строится аналогично оценке формы ПУ, но с набором ЧЗ, соответствующим форме ПА.

Пусть:

$S = \bigcup\limits_{n}(v_i)$ — множество всех ЧЗ, из которых синтезируется знание (решение);

$S^a = \bigcup\limits_{l}(v_i^a)$ — множество ЧЗ, из которого синтезировано знание (решение), $S^a \subset S$;

$S^r(v_1^r, ..., v_m^r) = \bigcup\limits_{m} v_i^r$ — множество правильных ЧЗ и $S^r \subset S$.

Ценность знания (решения) $E(S^a)$ знания формы ПА в традиционном понимании логики ИЛИ представляется следующим образом:

$$E(S^a) = \begin{cases} 1, \text{ если } S^a \subset S^r; \\ 0, \text{ если } S^a \not\subset S^r. \end{cases}$$

Ценность знания (решения) для случая, когда требуется представить *все возможные альтернативы* достижения цели-понятия, определяется следующим образом:

$$E(S^a) = \begin{cases} 1, \text{ если } S^a = S^r; \\ 0, \text{ если } S^a \neq S^r. \end{cases}$$

Логическая модель формы ПУ

Пусть:

$S(v_1, ..., v_n)$ — множество всех ЧЗ, из которых синтезируется СЗ; $S^r(v_1^r, ..., v_m^r)$ — множество правильных ЧЗ и $S^r \subset S$; $S^u(v_1^u, ..., v_k^u)$ — множество неправильных ЧЗ и $S^u \subset S$; $S^{ru}(v_1^{ru}, ..., v_F^{ru})$ — правильные необязательные ЧЗ и $S^{ru} \subset S^r$; S^{ruu} — правильные безусловно необязательные ЧЗ; S^{ruc} — правильные условно необязательные ЧЗ; $S^{ruu} \cup S^{ruc} = S^{ru}$; S^{rn} — множество правильных обязательных ЧЗ и $S^{rn} \subset S^r$; $S^{ru} \cup S^{rn} = S^r$; $S^{uu}(v_1^{uu}, ..., v_L^{uu})$ — множество неправильных пассивных ЧЗ и $S^{uu} \subset S^u$; $S^{ua}(v_1^{ua}, ..., v_T^{ua})$ —

множество неправильных активных ЧЗ и $S^{ua} \subset S^u$; $S^{uu} \cup S^{ua} = S^u$ и $S = S^{ru} \cup S^{rn} \cup S^{uu} \cup S^{ua}$; $S^a(v_1^a, ..., v_p^a)$ — множество ЧЗ данного решения и $S^a \subset S$;

$$E(S^a) = \begin{cases} 1, \text{ если } S^a = (S^{ra} \cup (\bigcup\limits_f S^{ru})) \cup \\ \cup (\bigcup\limits_l (v_i^{uu})),\ 0 \leqslant l \leqslant L,\ \ 0 \leqslant f \leqslant F; \\ 0, \text{ если } S^{ra} \not\subset S^a; \\ 0, \text{ если } S^a \supset \bigcup\limits_t (v_i^{ua}),\ 1 \leqslant t \leqslant T. \end{cases}$$

Логическая модель формы ПД

Пусть

$S(v_1, ..., v_n)$ — неупорядоченное множество всех ЧЗ, из которых синтезируется ответ; $S^r(v_1^r, ..., v_m^r)$ — неупорядоченное множество правильных ЧЗ и $S^r \subset S$; $S^u(v_1^u, ..., v_k^u)$ — множество неправильных ЧЗ и $S^u \subset S$; $S^{ru}(v_1^{ru}, ..., v_F^{ru})$ — неупорядоченное множество правильных необязательных ЧЗ и $S^{ru} \subset S^r$; S^{rn} — неупорядоченное множество правильных обязательных ЧЗ и $S^{rn} \subset S^r$; $S^{ru} \cup S^{rn} = S^r$; $S^{uu}(v_1^{uu}, ..., v_L^{uu})$ — множество неправильных пассивных ЧЗ и $S^{uu} \subset S^u$; $S^{ua}(v_1^{ua}, ..., v_T^{ua})$ — множество неправильных активных ЧЗ и $S^{ua} \subset S^u$; $S^a(v_1^a, ..., v_p^a)$ — упорядоченное множество ЧЗ данного решения и $S^a \subset S$.

Определение. Минимально допустимой конфигурацией решения называется образ решения, являющийся упорядоченным множеством элементов ответа $S_{\min}$, состав и последовательность расположения составляющих элементов которого обеспечивает выполнение цели задания с минимально допустимым относительно поставленной цели эффектом.

Ценность решения

$$E(S^a) = \begin{cases} 1,\ S_{\min} \subset S^a; \\ 0,\ S_{\min} \not\subset S^a. \end{cases}$$

Условие достижения минимально допустимой конфигурации решения

$$C(S^a) = \bigcup\limits_k C_k,$$

где C_1 — отсутствие в S^a неправильных активных ЧЗ; C_2 — наличие в S^a всех правильных обязательных ЧЗ (включая условно необязательные элементы, признанные обязательными); C_3 — порядок следования элементов множества S^a "жесткой структуры" совпадает с порядком элементов "жесткой структуры" множества $S_{\min}$; C_4 — действия плавного эффекта ДПЭ множества S^a не выходят за границы соответствующих диапазонов действия линейного эффекта ДДЛЭ; C_5 — действия произвольной последовательности ДПП множества S^a не выходят за границы соответствующих диапазонов действия произвольной последовательности ДДПП.

Заключение

Предложенный подход к структурированию и моделированию сложных знаний-решений целесообразен особенно там, где принятие неадекватных ситуации решений чревато тяжелыми последствиями. Мы применяли описанные методы при разработке компьютерной обучающей системы в форме деловых игр и анализа конкретных ситуаций [2, 5], при построении систем интеллектуальной поддержки решений организаций при пожаре [6, 7]. Отметим также, что на МПМ основывается численный *метод вероятностной оценки сложных знаний* [2, 3, 6, 7], в контексте которого вводится дополнительное структурирование СЗ. В рамках обучающей системы в области юриспруденции с использованием МПМ разработан ряд конкретных ситуаций [8]. Представляется имеющей практический выход проведенная нами работа по структурированию существенно нечетких юридических критериев вины. В результате получен четкий и непротиворечивый критерий вины, который может использоваться в качестве элемента системы интеллектуальной поддержки юридических решений.

Список литературы

1. **Zadeh L. A.** Furry Sets // Information and Control. 1965. № 8. P. 338—353.
2. **Евграфов П. М., Глуховенко Ю. М.** Ноу-хау обучающих программ и деловых игр. М.: АРС, 2004. 222 с.
3. **www.indelig.narod.ru** — Интернет-сайт Научно-инновационной группы "Инделиг".
4. **Рейтман У. Р.** Познание и мышление. Моделирование на уровне информационных процессов. М.: Мир. 1968.
5. **Евграфов П. М.** О применении метода психологического моделирования в контрольно-обучающих программах и в психометрических тестированиях интеллекта // Научно-техническая информация. Сер. 1. Изд. ВИНИТИ. 2002. № 4. С. 15—18.
6. **Евграфов П. М., Евграфов И. П.** Система интеллектуальной поддержки принятия решений организации при пожаре // Пожаровзрывобезопасность. 2006. № 4. С. 10—18.
7. **Евграфов И. П.** Компьютерная обучающая система с деловыми играми // Информационные технологии. 2007. № 3. С. 49—52.
8. **Евграфов П. М.** Деловые игры в юриспруденции (методом психологического моделирования сложных знаний) // XV международная конференция-выставка "Информационные технологии в образовании". Сб. тр. участников конф. Ч. IV. М.: БИТ про. 2005. С. 296—298.

УДК 025.4.01:004.8

С. Г. Керимов, д-р техн. наук, проф.,
Азербайджанская государственная
нефтяная академия (г. Баку)

О модели онтологии предметной области, модели информационного поиска и коррекции запросов

Рассматриваются модель онтологии предметной области, модель поиска информации и принципы коррекции запросов в информационной системе, ориентированной на онтологии.

Ключевые слова: *информационная система, модели, онтология предметной области, информационный поиск, коррекция запросов.*

Введение

Круг технологий, связанных с использованием онтологии, расширяется. В настоящее время онтология нашла свое применение в мультиагентных системах, в технологии TEXT MINING, в информационно-поисковых системах, в автоматическом индексировании и аннотировании и т. д. В рамках SEMANTIC WEB информация описывается с помощью онтологии предметных областей [1]. Существует специальный европейский проект — KNOWLEDGEWEB, предназначенный для интеграции работ в области управления знаниями, и SEMANTIC WEB, который направлен на активизацию применения технологий на основе онтологии в корпоративных системах. Развертываются работы по реализации концепции SEMANTIC WEB в рамках отдельной организации и созданию корпоративной памяти, предназначенной для накопления и управления знаниями предприятия на основе онтологии [2].

Особенно эффективным является применение онтологии в информационном поиске. Роль онтологии в улучшении показателей поиска информации (особенно точности поиска) неоспорима [3—5].

В статье рассматриваются модель онтологии предметной области (ПО) применительно к информационному поиску, модель поиска информации на основе онтологии и принципы коррекции запросов.

Модель онтологии предметной области

Понятию, предмету и типам онтологии посвящены многочисленные публикации, в том числе работы [6—10]. В точных науках разделяют следующие типы онтологии:

- **метаонтология:** описывает наиболее общие понятия, которые не зависят от предметных областей;
- **онтология предметной области:** формальное описание предметной области. Применяется для уточнения понятий, определенных в метаонтологии и/или для определения терминологической базы предметной области;
- **онтология конкретной задачи:** определяет терминологическую базу задачи (проблемы);
- **сетевая онтология:** используется для описания конечных результатов действий, выполняемых объектами предметной области или задачи.

Формальную модель онтологии предметной области можно определить как

$$O = < T,\ R,\ F >,$$

где T — конечное множество терминов (понятий) предметной области; R — конечное множество отношений между терминами; F — функции интерпретации, заданных на терминах и/или отношениях онтологии O.

Множества T и R формируются по следующему алгоритму.

1. Выбирается представительный массив текстовых документов (МТ), относящихся к рассматриваемой ПО.

2. Путем извлечения лексических единиц (слов, словосочетаний и конструкций) из МТ создается тематический словник (ТС).

3. Каждый компонент ТС расширяется за счет всех связанных с ним словоформ.

4. На основании ТС формируются первоначальный список понятий (СП) предметной области.

5. Осуществляется классификация элементов СП в соответствии с базовыми семантическими категориями: объект, процесс, событие, свойство, значение и т. д.

6. Между элементами СП устанавливаются иерархические отношения: "целое—часть", "общее (класс)—частное (подкласс или экземпляр)" и отношения типа "объект—свойство" и т. д.

7. Сформированная онтология дополняется компонентами (терминами и отношениями), специфическими для данной ПО.

Таким образом, в первом приближении онтология считается готовой. Доведение онтологии до необходимой полноты осуществляется дальнейшим расширением массивов МТ, ТС и СП.

В онтологию, используемую в информационном поиске, целесообразно включить также отношения, характерные для информационно-поискового тезауруса, такие как "синонимия", "ассоциация", которые придают еще большую семантическую силу онтологическому словарю.

Таким образом, множество отношений R формально можно представить как

$$R = <ИО,\ OC,\ CO,\ AO>,$$

где $ИО$ — иерархические отношения; OC — отношения типа "объект—свойство"; CO — синонимические отношения; AO — ассоциативные отношения.

Каждый i-й термин T^i может иметь иерархические связи с одним вышестоящим (BC) термином типа "класс" или "целое" и несколькими нижестоящими (HC) терминами типа "подкласс" или "экземпляр":

$$ИО^i = \{BC^i,\ HC_k^i\},\ \ k = \overline{1, n},$$

где n — число нижестоящих терминов.

Множества BC и HC могут быть пустыми.

Отношение OC имеет место, когда термин T^i явно вступает как объект. В этом случае в онтологию включают основные свойства (характеристики) объекта, представляющие интерес в рамках данной предметной области:

$$OC^i = \{P_r^i\},\ \ r = \overline{1, m},\ P_r^i \subseteq P_0,$$

где m — число свойств объекта O^i; P_0 — множество свойств онтологии ПО.

Синонимические отношения включают в себя безусловные и условные (в пределах ПО) синонимы, в том числе различные словоформы термина T^i:

$$CO^i = \{S_q^i\},\ \ q = \overline{1, t},$$

где q — число безусловных и условных синонимов T^i.

Для унификации описания понятий онтологии соответствующие термины представляются в именительном падеже единственного числа.

Ассоциативные отношения определяют связи термина T^i с близкими по значению (в пределах ПО) терминами:

$$AO^i = \{A_f^i\},\ \ f = \overline{1, u},$$

где u — число ассоциативных терминов.

Множество AO может быть пустым. В некоторых информационных системах ассоциативные отношения не используются. Чтобы отличить основной термин от его синонимов, он каким-то образом выделяется. В дескрипторных языках выделенный термин называют дескриптором. При индексировании (составлении поисковых образов) документов и запросов все синонимичные термины, встречающиеся в документах и запросе, заменяются соответствующим дескриптором. В дальнейшем изложении мы также будем ориентироваться на дескрипторы.

Модель поиска информации на основе онтологии

Будем считать, что онтология используется на этапе формирования метаописания (поискового образа) текстовых документов и на этапе построения запроса к поисковой системе [5]. В качестве критерия смысловой близости (КСБ) примем общеизвестный и часто применяемый в документальном поиске критерий "вхождение поискового образа запроса в поисковый образ документа".

Обозначим через Z^i поисковый образ i-го запроса, а через D^i — поисковый образ j-го документа. Тогда КСБ в общем виде можно представить так: j-й документ отвечает i-му запросу, если

$$Z^i \subseteq D^j = T,$$

где T — есть предикат истинности.

В запросе предусматривается возможность использования между терминами логических отношений AND, OR, NOT. Смысл их очевиден. Отношение AND задается по умолчанию.

Поисковый образ запроса Z^i можно представить так:

$$Z^i = d_1^i\, \theta\, d_2^i\, \theta ... \theta\, d_n^i,$$

где $d_1^i,\ d_2^i,\ ...,\ d_n^i$ — дескрипторы (термины), входящие в Z^i; n — число дескрипторов в Z^i;

$$\theta \in R^l,\ R^l = \{AND,\ OR,\ NOT\}.$$

Если в запросе отсутствуют отношения OR и NOT, то при поиске на i-й запрос будут выданы документы, удовлетворяющие условию

$$(\forall k\ \exists j\, d_k^i \in D^j) = T,\ \ k = \overline{1, n}.$$

Если дескрипторы d_a^i и d_b^i, где $d_a^i \in Z^i$, $d_b^i \in Z^i$, связаны логическим отношением OR, то к выдаче подлежат документы, удовлетворяющие условию:

$$(\forall k\ k \neq a,\ k \neq b\ \exists_j d_k^i \in D^j)\ AND \rightarrow$$

$$\rightarrow (\forall a,\ b(d_a^i \in D^i)\ OR\ (d_b^i \in D^i)) = T,\ k = \overline{1, n}.$$

Если дескриптору d_c^i, где $d_c^i \in Z^i$, предшествует знак NOT, то условием поиска будет следующее:

$$(\forall k\ k \neq c,\ \exists_j d_k^i \in D^j)\ AND\ (\forall c,\ d_c^i \notin D^j) = T,$$

$$k = \overline{1, n}.$$

Коррекция запросов

В любой поисковой системе, в том числе в системе, основанной на онтологии, должна быть предусмотрена возможность коррекции запросов. Коррекция запроса проводится в случае, когда результаты поиска не удовлетворяют пользователя.

Коррекцию запроса можно проводить двумя способами:

1) вручную, когда пользователь в режиме диалога с системой может удалить некоторые термины исходного запроса, включить в запрос новые термины, заменить исходные термины другими;

2) автоматически, в этом случае на основании иерархических отношений онтологии термины исходного запроса заменяются вышестоящими или нижестоящими терминами.

Ручная коррекция запроса, проводимая на основе онтологии, носит итеративный характер. Операции удаления, включения и замены термина (дескриптора) выполняются шаг за шагом до тех пор, пока пользователь не получит искомые документы. Операция удаления термина проводится для увеличения полноты поиска; операция включения в запрос нового термина — для увеличения точности поиска, а операция замены термина может влиять как на полноту, так и на точность поиска. Замена исходного термина на вышестоящий служит для семантического обобщения значения понятия, представленного данным термином, замена исходного термина на нижестоящий служит для декомпозиции сложного понятия и сужает его семантическое значение.

Автоматическая коррекция запроса выполняется программным путем. При этом режим расширения или сужения поискового пространства по каждому термину (дескриптору) выбирается пользователем на основе анализа выдачи на предыдущем этапе поиска.

Для расширения поискового пространства по термину t_k^i i-го запроса выходом на онтологии этот термин заменяется на вышестоящий термин $BC(t_k^i)$ и повторяется поиск. В случае необходимости эту процедуру можно продолжить, т. е. заменить термин $BC(t_k^i)$ термином, более высокого класса.

Для сужения поискового пространства по термину t_k^i i-го запроса выходом на онтологии шаг за шагом этот термин заменяется на нижестоящие термины $HC(t_n^i)$ и процесс поиска повторяется.

Рассмотренные варианты коррекции запросов дают возможность проведения итеративного поиска с вариацией полноты и точности поиска. Каждый вариант коррекции дает очередную итерацию поиска.

Список литературы

1. **Berners-Lee T., Hendler J., Lassila O.** The Semantic Web // Scientific American. 2001. May.
2. **Кудрявцев Д.** Технология применения онтологий. http://bigzpb.ru/theory/km/onto-technologies.php.
3. **Россеева О. И., Загорулько Ю. О.** Организация эффективного поиска на основании онтологии. www.dialog-21.ru/archive/2001/volime_2/2-49.htm.
4. **Сизиков Е. В., Сошников Д. В.** Онтологическая поисковая система для реализации интеллектуального поиска в Интернет-Интранет-сетях. www.soshnikov.com/publications/trud-mai-siz.pdf.
5. **Керимов С. Г.** Интеллектуальный поиск информации, основанный на онтологии // Информационные технологии. 2004. № 11. С. 12—16.
6. **Онтология.** Материал из Википедии — свободной энциклопедии. http://ru.wikipedia.org/wiki.
7. **Гаврилова Т. А., Хорошевский Б. Ф.** Базы знаний интеллектуальных систем. — СПб.: Питер, 2001.
8. **Клещев А. С., Артемьева И. Л.** Математические модели онтологии предметных областей. Ч. 1. Существующие подходы к определению понятия "Онтология" // Научно-техническая информация. Сер. 2. 2001. № 2. С. 12—18.
9. **Гладун А. Я., Рогушина Ю. В.** Онтология в корпоративных системах // Корпоративные системы. 2006. № 1. http://www.management.com.ua/ims/imsll6.html.
10. **Жыжырий Е. А., Щербак С. С.** Математическое обеспечение систем поиска, основанных на онтологиях. http://shcherbak.net/mat_obez/.

УДК 301:004.89

С. Л. Гольдштейн, д-р техн. наук, зав. каф., **А. Г. Кудрявцев,** канд. физ.-мат. наук, доц.,
Уральский государственный технический университет

Проблематика создания системного интеллектуального подсказчика по разрешению проблемных ситуаций

Рассмотрена проблематика создания системы поддержки принятия решений, способной к поддержке разрешения проблемных ситуаций со сложными объектами, пополнению знаний пользователя и интеллектуальной самопомощи.

Ключевые слова: *системный интеллектуальный подсказчик (СИП); система поддержки принятия решений (СППР); проблемная ситуация; разрешение проблемных ситуаций; поддержка разрешения проблемных ситуаций; система ситуационного управления; система обнаружения знаний; автоматизированная обучающая система (АОС).*

Актуальность и постановка задачи

Общество в процессе своей информатизации признало необходимость инженерии знаний и управления ими [1—20]. В рамках этой глобальной цели существует локальная цель поддержки лица, принимающего решение (ЛПР), в разрешении проблемных ситуаций, связанных с необходимостью перевода того или иного сложного объекта [12, 13, 21] (например, социоорганизационного, технического, экономического, биологического и т. п.) в новое качество [2—6, 12—18, 20, 22, 23]. При этом недостаточное число специали-

стов-экспертов, способных поддержать ЛПР в нужное время и в нужном месте, а также необязательность их реакции при отсутствии (недостатке) стимулирования, диктуют необходимость полной или частичной замены их естественного интеллекта на искусственный, например, на интеллектуальную Интернет-систему [1, 24], а при необходимости работы с корпоративными знаниями — на какую-либо другую компьютерную систему поддержки принятия решений (СППР) [25, 26].

В соответствии с работой [25] всякая СППР способна в той или иной мере поддержать разрешение проблемных ситуаций со сложными объектами. В то же время наиболее пригодными для данной задачи оказываются системы ситуационного управления [12—14]. Это связано прежде всего с тем, что рассматриваемые системы предваряют поддержку принятия решения фиксацией, анализом и моделированием ситуации. В то же время системы ситуационного управления ориентированы исключительно на прямую поддержку принятия решений [26], что может привести к непониманию ЛПР выдаваемой рекомендации при недостаточности знаний последнего. Кроме того, согласно [20], традиционно используемые формы ситуационных моделей и способы их построения (по Д. А. Поспелову) [12, 13] существенно затрудняют (делая практически невозможным) создание базы надпредметных (системных) метазнаний для систем ситуационного управления, а значит, и реализацию ими функции интеллектуальной самопомощи (или выдачи системной подсказки) [2—4, 16, 18, 20, 23, 27—31], которая, согласно [16, 23], может оказаться существенной при поддержке разрешения проблемных ситуаций.

Таким образом, актуальна задача создания СППР, способной к поддержке разрешения проблемной ситуации со сложными объектами, генерированию системных подсказок и пополнению знаний ЛПР. При этом естественно требовать возможности создания универсальной базы знаний (для реализации всех обозначенных функций) по единой процедуре и единства технологии управления знаниями (т. е. связности соответствующей инфотехнологической схемы как графа).

В соответствии с ранними публикациями [2—6, 16, 18, 20, 27, 29, 30] будем называть СППР, которую требуется создать, системным интеллектуальным подсказчиком (СИП) по разрешению проблемных ситуаций со сложными объектами.

Предпосылки создания системного интеллектуального подсказчика

Поддержка разрешения проблемных ситуаций со сложными объектами. Как отмечено в предыдущем разделе, в качестве базового типа существующих СППР, реализующих данную функцию, следует рассматривать системы ситуационного управления.

Аналоги СИП и их оценки

Аналоги		Оценка способности: к разрешению проблемных ситуаций со сложными объектами	Оценка способности: к системной подсказке	Оценка способности: к обучению ЛПР	Суммарный балл
Экспертные системы [1, 25, 41]		0,5	0,25	0,6	1,35
Системы динамического прогнозирования (альтернативные экспертным) [42]		0,3	0,25	0,1	0,65
Системы ситуационного управления [12—14]		1	0,25	0	1,25
Расчетно-диагностические системы [25]		0,4	0,25	0,25	0,9
АОС [25, 39, 40]		0,2	0,25	1	1,45
Системы нейросетевых вычислений [25, 43, 44]		0,1	0,25	0,25	0,6
Системы эволюционного моделирования [25, 43, 44]		0,1	0,25	0,1	0,45
Лингвистические СППР [4, 9—11, 31—36]	Системы общения [33, 34]	0,1	1	0,1	1,2
	Системы обнаружения знаний [4, 9—11, 31, 32, 35]	0,6	1	0,5	2,1
	Прочие лингвистические СППР (основанные на онтологиях) [36]	0,3	1	0,4	1,7
Интеллектуальные Интернет-системы [1]		0,5	0,25	0,7	1,45
Партнерские системы [25, 44—46]		0,6	0,25	0,7	1,55

Генерирование системных подсказок. Согласно [2, 4] наиболее приспособленными к реализации данной функции оказываются лингвистические СППР [4, 9—11, 31—36], включая развитый вариант [4, 9—11, 31, 32, 35], способный в значительной мере поддерживать разрешение проблемных ситуаций [9—11], основанный на тезаурусно-онтологически представленных знаниях [1, 36, 37] и называемый системой обнаружения знаний в соответствии с терминологией [9—11]. Это связано с присутствием в составе указанных систем при-

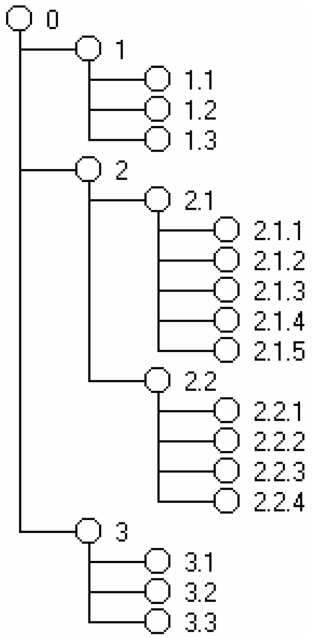


Рис. 1. Иерархическая структура системы обнаружения знаний

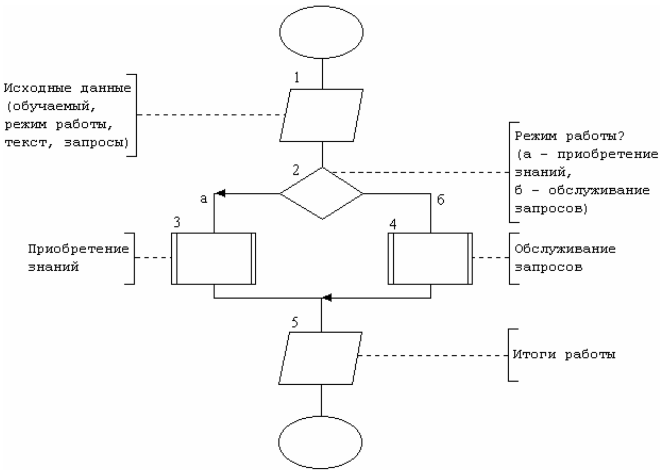


Рис. 2. Алгоритм лингвистического обнаружения знаний в целом

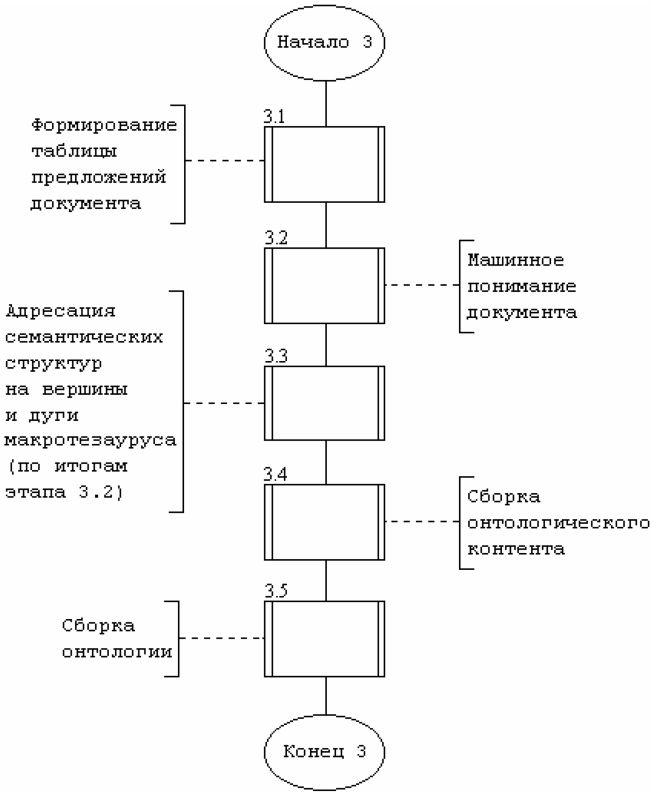


Рис. 3. Алгоритм приобретения знаний (блок 3 на рис. 2)

обретателя знаний [25] и возможностью подключения системы протокольного сопровождения (СПС) с текстовым выходом [38], что в совокупности позволяет достаточно легко строить базы системных знаний на основе текстового описания используемой лингвистической СППР и отчета по ее функционированию, сформированного СПС.

Пополнение знаний ЛПР. Согласно [25, 39, 40] для реализации данной функции целесообразно использовать интеллектуальные автоматизированные обучающие системы (АОС) с электронным источником знаний (информационной базой), блоком контроля и маршрутизатором обучения [5, 6] в своем составе.

Аналоги СИП. В качестве аналогов будем рассматривать СППР различных типов, оценки которых даны выше, в таблице.

Видно, что ни одна из существующих СППР не способна реализовать все функции СИП в совокупности.

Прототип СИП. За прототип следует принять систему обнаружения знаний, как имеющую максимальный суммарный балл согласно таблице.

Иерархическая структура прототипа СИП и исполняемый им алгоритм (на основе сведений из [4, 9—11, 31—33]) показаны на рис. 1—4.

На рис. 1 даны следующие обозначения:

0 — система обнаружения знаний; 1 — система (база) знаний (СЗ); 1.1 — текстовый блок; 1.2 — сетевой блок; 1.3 — онтологический блок; 2 — система управления знаниями (СУЗ); 2.1 — блок приобретения знаний; 2.1.1 — узел формирования таблицы предложений исходного текста (документа); 2.1.2 — узел машинного понимания [33] документов; 2.1.3 — адресатор семантических структур [5, 6, 25, 33, 35]* на элементы макротезауруса [2, 3, 35]*; 2.1.4 — сборщик онтологического контента [36]; 2.1.5 — сборщик онтологии; 2.2 — блок обслуживания запросов; 2.2.1 — узел машинного понимания запросов**; 2.2.2 — опре-

* Семантические структуры и макротезаурус получены в процессе машинного понимания текста.

** Для запросов процесс машинного понимания проще, чем для документов [5, 6, 25, 33].

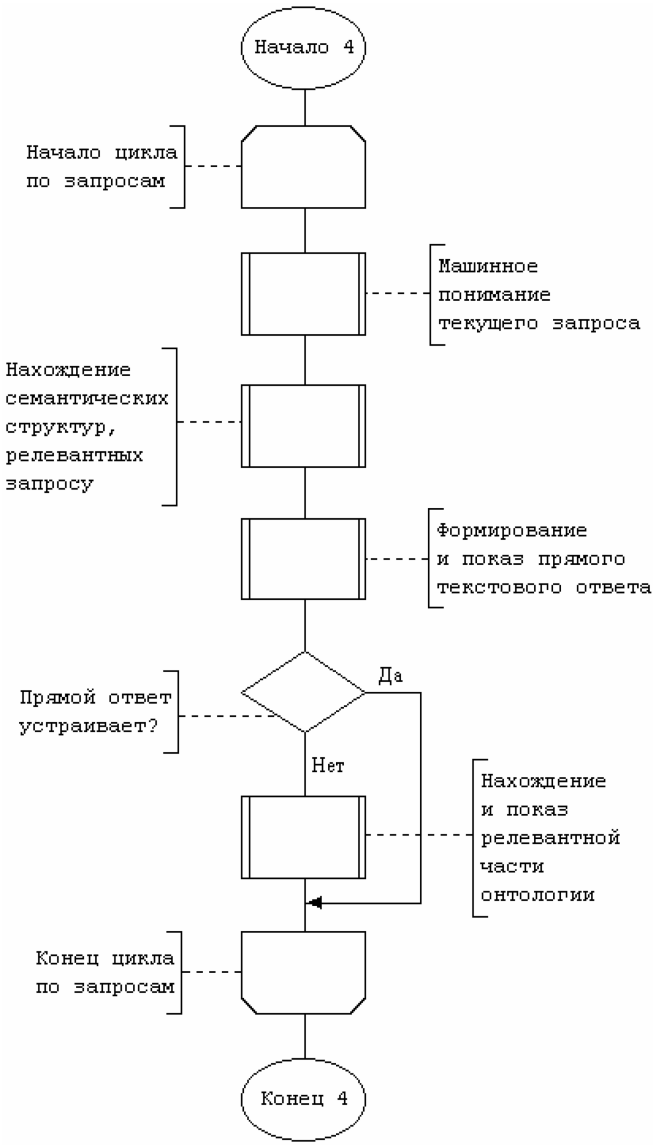


Рис. 4. Алгоритм обслуживания естественно-языковых запросов (блок 4 на рис. 2)

делитель семантических структур документа, релевантных запросу; 2.2.3 — генератор прямого текстового ответа [5, 6, 35]; 2.2.4 — определитель релевантной части онтологии; 3 — система протокольного сопровождения (СПС)***; 3.1 — память состояний [41]; 3.2 — генератор текстовых отчетов; 3.3 — память отчетов).

Недостатки системы обнаружения знаний как прототипа СИП. По алгоритму — отсутствие в полной мере представленных функций разрешения проблемных ситуаций и обучения ЛПР. По структуре — нерешенность вопроса о бесконфликтной интеграции с АОС и особенно системой ситуационного управления (ввиду использования принципиально иной формы представления знаний в существующих вариантах последней [12, 13], что, в свою очередь, исключает возможность использования единой процедуры создания знаний для систем ситуационного управления и обнаружения знаний).

Задача на дальнейшее исследование: развить прототип СИП путем его бесконфликтной интеграции с системой ситуационного управления и АОС (с возможным внесением изменений в системы, объединяемые с прототипом).

*** Присутствие СПС в составе системы обнаружения знаний не обязательно, но может сыграть существенную роль при организации системных подсказок (см. выше).

Результаты и выводы

1. Рассмотрены предпосылки создания системного интеллектуального подсказчика по разрешению проблемных ситуаций со сложными объектами.

2. Предложены критериальные модели различных типов систем поддержки принятия решений как аналогов системного интеллектуального подсказчика.

3. На основе предложенных критериальных моделей выбран прототип системного интеллектуального подсказчика.

4. На основе сведений из литературных источников даны описания иерархической структуры и исполняемого алгоритма для прототипа системного интеллектуального подсказчика.

5. Приведены недостатки прототипа системного интеллектуального подсказчика по структуре и алгоритму.

6. Поставлена задача развития прототипа системного интеллектуального подсказчика путем его бесконфликтной интеграции с другими системами поддержки принятия решений.

Вывод: в результате проведенного исследования обозначен путь к созданию системного интеллектуального подсказчика на основе интеграции систем поддержки принятия решений различных типов.

В ближайшей перспективе целесообразно рассмотреть возможные способы указанной интеграции, что, в свою очередь, должно позволить описать технологию СИП.

Список литературы

1. **Гаврилова Т. А., Хорошевский В. Ф.** Базы знаний интеллектуальных систем: Учеб. пособие для вузов. СПб.: ПИТЕР, 2000. 384 с.

2. **Ткаченко Т. Я.** Инструментальная среда системотехнического обслуживания сложных объектов. Екатеринбург: Изд. ГОУ ВПО "УГТУ-УПИ", 2002. 203 с.

3. **Гольдштейн С. Л., Ткаченко Т. Я., Бельков С. А.** и др. Системный аспект информатизации правоохранительных органов: выход на системные интеллектуальные подсказчики по управлению переводом в новое качество. Екатеринбург: Изд. УГТУ-УПИ, 1995. 190 с.

4. **Гольдштейн С. Л., Ткаченко Т. Я.** Концептуально-системная модель инструментальной среды системотехнического

обслуживания сложных объектов / Урал. полит. ин-т. — Деп. в ВИНИТИ, 1991. — № 3707.

5. **Кудрявцев А. Г.** Модели и алгоритмы в интересах развития компьютерных подсказчиков: Дисс. ... канд. физ.-мат. наук. — Екатеринбург, 2005.

6. **Гольдштейн С. Л., Кудрявцев А. Г.** Разрешение проблемных ситуаций при поддержке систем, основанных на знаниях: Учеб. пособие. Екатеринбург: ИД "ПироговЪ", 2006. 218 с.

7. **Фрапаоло К., Томс В.** Управление знаниями: от Земли неведомой к земле обетованной // Электронный офис. 1998. № 2.

8. **Боуэн Т. С., Скэннел Э.** Это таинственное управление знаниями // Computer World. 1999. № 9. С. 27.

9. **Satyadas A.** Growing with Knowledge Management, 2003; http://www.line56.com/articles/default.asp? Article ID = 4408.

10. **Satyadas A.** Knowledge discovery strikes the balance between business control and innovation. 2002, Nov 5. http://www.lotus.com/news/news.nsf/public/1D77D354B2F9AB4285256C5D0046BD44.

11. **Kajmo D.** Knowledge Management in R5. 03 May 1999; http://www.10.lotus.com/Idd/today.nsf/DisplayForm/FAB0891EECD6D98 98525670500776F59?OpenDocument.

12. **Поспелов Д. А.** Большие системы. Ситуационное управление. М.: Знание, 1975. 64 с.

13. **Поспелов Д. А.** Ситуационное управление: Теория и практика. М.: Наука, 1986. 284 с.

14. **Данчул А. Н.** Информационно-аналитические технологии и ситуационные центры // Государственная служба. 2004. № 4.

15. **Инюшкина О. Г.** Модели и комплексы программ для развития систем управления знаниями: Дисс. ... канд. физ.-мат. наук. Екатеринбург, 2005.

16. **Гольдштейн С. Л.** Системная интеграция бизнеса, интеллекта, компьютера. Кн. 1: Введение в проблематику и постановку задач: Учеб. пособие. Екатеринбург: ИД "ПироговЪ", 2006. 392 с.

17. **Гольдштейн С. Л., Инюшкина О. Г., Кормышев В. М.** Развитие системы управления знаниями для разрешения ситуаций в бизнесе. Екатеринбург: ИД "ПироговЪ", 2006. 220 с.

18. **Гольдштейн С. Л., Ткаченко Т. Я.** Разработка системного интеллектуального подсказчика по типологии управления сложными объектами / ГОУ ВПО "УГТУ-УПИ". — Деп. в ВИНИТИ. 1996. № 1602.

19. **Инюшкина О. Г., Гольдштейн С. Л., Макаров Э. П.** Свидетельство об официальной регистрации программы для ЭВМ № 2002610271 "КИРП-Р-НПУ".

20. **Ткаченко Т. Я.** Интеллектуально-информационная поддержка нечетких наукоемких технологий: Дисс. ... докт. тех. наук. — Екатеринбург, 2002.

21. **Гольдштейн С. Л., Ткаченко Т. Я.** Введение в системологию и системотехнику. Екатеринбург: ИРРО, 1994. 198 с.

22. **Печеркин С. С.** Теоретическое описание и развитие системной интеграции для научно-практических структур: Дисс. ... канд. физ.-мат. наук. Екатеринбург, 2002.

23. **Гольдштейн С. Л., Печеркин С. О., Ткаченко Т. Я.** Системная интеграция: системно-информациологический подход к проблеме знаний и управления знаниями // Инфор. — 2000. № 1. С. 27—44.

24. **Информационная** биржа знаний; www.yaznau.ru

25. **Романов А. Н., Одинцов Б. Е.** Советующие информационные системы в экономике: Учеб. пособие для студентов вузов. М.: ЮНИТИ-ДАНА, 2000. 487 с.

26. **Уточнение** понятия "система поддержки принятия решений". www.gorskiy.ru/articles.html

27. **Гольдштейн С. Л., Ткаченко Т. Я., Яремко Н. Л.** Моделирование инструментальной оболочки ИС СОСО // Интеллектуальные информационные технологии в управленческой деятельности: Сборник. Екатеринбург: ИПК УГТУ-УПИ, 2002. С. 107—125.

28. **Гольдштейн С. Л., Павленко А. П., Блохина С. И.** О синтезе интеллектуального интегрированного АРМ специалиста-исследователя научно-практического учреждения // Инфор—2000. № 1. С. 75—78.

29. **Гольдштейн С. Л.** Научные направления, разработки и интересы кафедры вычислительной техники // Наука — среда обитания: Сб. статей. Екатеринбург: УГТУ-УПИ, 1999. С. 97—126.

30. **Гольдштейн С. Л., Ткаченко Т. Я., Бельков С. А.** Базово-уровневые концепции для разработки интеллектуальной информационной среды // Новые информационные технологии в исследовании дискретных структур: Сборник. Екатеринбург. УрО РАН, 1998. С. 18—29.

31. **Блохина С. И., Гольдштейн С. Л., Ткаченко Т. Я.** Методология и инструментарий медико-технической интеграции // Вестник уральской медицинской академической науки. Екатеринбург: СУНЦ РАМН. 2003. № 2. С. 3—6.

32. **Бодякин В. И.** Экспертные системы нового поколения; http://mosaica.narod.ru/Phisics/ExpertSystemsNG.htm.

33. **Попов Э. В**. Общение с ЭВМ на естественном языке. М.: Наука, 1982. 360 с.

34. **Дмитриев А. С.** Детерминированный хаос и информационные технологии // Компьютерра. 1998. № 47. С. 27—30.

35. **Гольдштейн С. Л., Кудрявцев А. Г.** Система наполнения и обнаружения знаний для системного интеллектуального подсказчика // Новые образовательные технологии в вузе: сборник докладов пятой Международной научно-методической конференции, 4—6 февраля 2008 года. В 2-х частях. Ч. 1. Екатеринбург: ГОУ ВПО "УГТУ-УПИ", 2008. С. 188—192.

36. **Овдей О. М., Проскудина Г. Ю.** Обзор инструментов инженерии онтологий; www.rcdl.ru/papers/2005/sek_32_paper.pdf.

37. **Нариньяни А. С.** Кентавр по имени ТЕОН: Тезаурус + Онтология. www.artint.ru/articles/narin/teon.htm.

38. **Джонс К. Д.** и др. Комбинированные средства системного аудита; www.intunit.ru/department/security/issec/12/.

39. **Обучающие** машины, системы и комплексы: Справочник. Киев: Вища шк., 1986. 303 с.

40. **Шкутина Л. А.** Автоматизированные обучающие системы как компонент современных технологий обучения // Телекоммуникации и информатизация образования. 2002. № 5. С. 60—62.

41. **Статические** и динамические экспертные системы: учеб. пособие для вузов / Э. В. Попов, И. Б. Фоминых, Е. Б. Кисель, М. Д. Шапот. — М.: Финансы и статистика, 1996. 320 с.

42. **Фролов Ю. В.** Интеллектуальные системы и управленческие решения. М.: МГПУ, 2000. 294 с.

43. **Шемакин Ю. И.** Семантика самоорганизующихся систем. М.: Академический проект, 2003. 176 с.

44. **Герович В. А.** Проблема самоорганизации в исследованиях по кибернетике и искусственному интеллекту // Концепция самоорганизации в исторической ретроспективе: Сборник. М.: Наука, 1994. С. 123—145.

45. **Загоруйко Н. Г.** Прикладные методы анализа данных и знаний. Новосибирск: Изд-во Ин-та математики, 1999. 270 с.

46. **Мазуров В. Д.** Распознавание образов и нейронные сети в моделировании технико-экономических систем // Интеллектуальные информационные технологии в управленческой деятельности: Сборник. Томск. С. 136—139.

УДК 50.51.17:50.41.29:73.37.81

В. И. Макаренко,
начальник научно-технического отдела,
Н. Н. Подольская, ведущий инженер,
Всероссийский НИИ радиоаппаратуры,
г. Санкт-Петербург,
e-mail: nonna@vniira-ovd.com, podolsky@rol.ru

Полезные приемы интерактивного проектирования программного обеспечения модифицируемых систем управления

Модифицируемость является неотъемлемым свойством современных систем управления и контроля. Показаны пути обеспечения высокой степени модифицируемости программного обеспечения на этапе его разработки при использовании приемов интерактивного проектирования. Примером служат элементы инструментария разработки компонентов интерфейса диспетчера автоматизированной системы управления воздушным движением.

Ключевые слова: *интерактивное проектирование программного обеспечения, модифицируемость программного обеспечения, человеко-машинный интерфейс, система управления воздушным движением, формуляр сопровождения воздушного судна.*

Введение

К современным цифровым системам управления и контроля, в особенности в сфере критических технологий, предъявляются повышенные требования, касающиеся быстрого и надежного проектирования и обеспечения высокой степени модифицируемости.

Предположим, что предприятие, имеющее опыт создания программного обеспечения (ПО) семейства успешно эксплуатируемых систем управления и контроля, получило заказ на проектирование очередной системы. Разработчик в этом случае может исходя из функциональной спецификации выбрать за основу наиболее подходящий член семейства, создать необходимые новые модули и внести изменения в старые. Новая версия ПО будет содержать уникальный набор файлов, который потребует отдельного сопровождения.

Невозможно создать систему, которая не потребует изменений в будущем. Для программ, эксплуатируемых в реальных условиях, модернизация — это необходимость. На 65 % модернизация обусловлена требованиями изменения иди расширения функциональных возможностей системы; в 18 % случаев происходят изменения рабочего окружения системы с непредусмотренным влиянием на ПО [1]. Система должна неизбежно изменяться, чтобы соответствовать новым требованиям. Новый программный проект разрабатывается 1—2 года, а эволюционирует 6—7 лет. На его сопровождение тратится 61 % затрат против 39 % на разработку [2]. Сопровождение ПО является непрерывным.

Внесение изменений в систему после ее поставки заказчику высоко затратно, поскольку для этого требуется хорошее знание системы и проведение квалифицированного анализа реализации этих изменений, иначе структура и программный код постепенно могут потерять целостность. Усилия, потраченные во время разработки на снижение стоимости такого анализа, снижают затраты на сопровождение. Вообще говоря, лучше предусмотреть заранее, какие изменения возможны в системе и с какими компонентами будет больше всего проблем при сопровождении.

Если разработка ПО базируется на семействах версий, то модификация очередной версии системы чревата появлением новых ошибок. Требуется последующая доработка и их исправление. Могут выявиться ошибки, общие для новой и предыдущих версий, и в каждой версии их надо будет исправлять.

Создание очередной системы и вместе с ним ее сопровождение значительно упростились бы, если бы разработчик прикладного ПО имел дело лишь с одним набором файлов, а не с их семейством, и выполнял в основном несложные действия по его "настройке", а непредусмотренные прежде функции внедрялись в ПО единообразным, прозрачным способом. Тогда процесс создания ПО по своему характеру приближался бы к процессу адаптации ПО к новым условиям функционирования, который должен быть более простым, по определению.

Вместе с тем известно, что для системы, работающей в реальных условиях, "правила", по которым она функционирует, могут быть изменены пользователем. Ему предоставляется возможность включения/отключения выполнения некоторых функциональных задач, а также изменения числовых параметров, причем набор задач и параметров назначается заказчиком.

Некоторые из таких изменений реализуются в рамках ПО целевой системы управления и контроля "на лету" с помощью встроенного в нее человеко-машинного интерфейса (ЧМИ).

Для выполнения других создается специальное ПО, не входящее в состав ПО целевой системы. Это отдельное ПО предоставляет пользователю свой ЧМИ настройки функциональности и параметризации целевой системы. Механизм же реализации пользовательских настроек распределен между обеими частями ПО.

В статье показаны преимущества переноса такого подхода на реализацию элементов проектирования ПО с предоставлением разработчику возможностей интерактивного проектирования.

Мы рассматриваем лишь аспекты внутренней структуры прикладного ПО, полагая, что смена платформы или среды разработки, отражая нефункциональные требования к системе, менее принципиально влияет на степень модифицируемости.

Конструктор формуляра сопровождения

Формуляр сопровождения (ФС) — это информационный блок, сопровождающий образ воздушного судна на диспетчерском экране автоматизированной системы управления воздушным движением.

На примере программы Конструктора ФС покажем, как формируется вид и функциональность компонента ЧМИ целевой системы посредством составления желаемого сочетания его свойств. При этом экранное представление и поведение компонента изменяются благодаря автоматической "настройке" программного кода, обеспечивающего функционирование целевой системы по прямому назначению, на начальной стадии реального режима ее работы.

Разработчику ПО предлагается окно, содержащее несколько однотипных (выполненных как объекты одного класса) наборов органов управления, организованных в таблицы (см. рисунок).

Число таблиц отражает разнообразие видов ФС с учетом статуса ФС (стандартный/выделенный), статуса полета (прилет/вылет/транзит), сектора управления (подход/круг/посадка) и др. Число строк каждой таблицы равно максимальному числу строк ФС, число столбцов — максимальному числу полей в строке ФС.

Каждая ячейка таблицы обслуживает поле ФС, определяемое значениями номера строки и номера поля в строке. Ячейка содержит два выпадающих меню. Список элементов первого меню отражает все множество типов полей ФС (Callsign — позывной, AFL — текущая высота, VTrend — тенденция изменения высоты и др.). Список элементов второго меню содержит набор всех возможных значений длины полей ФС.

Настраивая содержимое и длину каждого поля, разработчик конструирует вид ФС, который будет впоследствии присутствовать на диспетчерском экране целевой системы. Формирование вида ФС путем выполнения простых операций выбора из меню, допускающих выбор одного элемента, исключает свободно конструируемый пользовательский ввод информации и минимизирует ошибки.

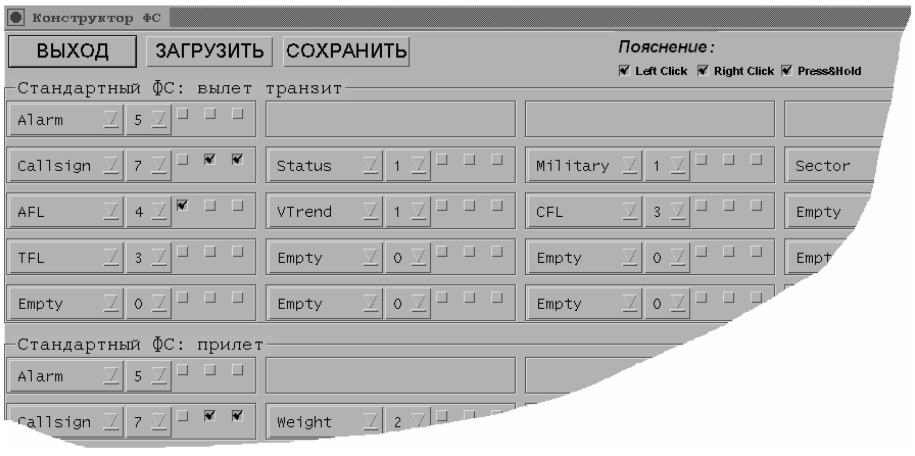
Кроме указанных меню каждая ячейка таблицы содержит три кнопки-переключателя для указания того, должна ли целевая система реагировать на следующие действия диспетчера:

- щелчок левой клавишей мыши по полю ФС;
- щелчок правой клавишей мыши по полю ФС;
- нажатие и удерживание правой клавиши мыши на поле ФС.

Окно может также содержать элементы настройки свойств, характеризующих ФС в целом, а не отдельные поля, например, приоритетность позиций автоматического отброса ФС [3].

Имеется кнопка "Сохранить", которая служит для записи сформированных конструкций всех видов ФС в файл конфигурации. Для каждого вида ФС файл содержит строку, идентифицирующую вид ФС, и строки, по структуре дублирующие информацию строк таблицы органов управления в окне.

Вновь запущенная программа Конструктора ФС, обнаруживая наличие такого файла, по умолчанию устанавливает выбранные элементы меню в соответствии с информацией, сохраненной в файле. Если файл конфигурации отсутствует, выбранные элементы меню соответствуют пустым полям.



Фрагмент окна Конструктора ФС

Таблица 1

Соответствие указателей на функции, связанные с полями ФС, типам полей

Тип поля ФС	Указатель на функцию			
	формирования поля ФС	обработки щелчка левой клавишей мыши	обработки щелчка правой клавишей мыши	обработки нажатия и удерживания правой клавиши мыши
Empty Callsign AFL VTrend ...	&emptyCreate &createCallsign &createAFL &createVTrend ...	&emptyLClick &LClickCallsign &LClickAFL &LClickVTrend ...	&emptyRClick &RClickCallsign RClickAFL &RClickVTrend ...	&emptyPressHold &pressHoldCallsign pressHoldAFL &pressHoldVTrend ...

Проведя некоторые манипуляции с меню, разработчик может признать их ошибочными, и тогда кнопка "Загрузить" послужит ему для возврата к сохраненной конфигурации.

В свою очередь, ПО целевой системы содержит описание структуры, содержащей следующие поля:

- тип поля ФС;
- указатель на функцию формирования поля ФС данного типа, параметрами которой являются номер строки и номер поля в строке, где поле должно располагаться в составе ФС;
- указатель на функцию обработки щелчка левой клавишей мыши по полю;
- указатель на функцию обработки щелчка правой клавишей мыши по полю;
- указатель на функцию обработки нажатия и удерживания правой клавиши мыши на поле.

Массив таких структур размером, равным числу возможных типов полей ФС, заполняется на стадии создания программного кода. Такой массив можно рассматривать как таблицу, где ***тип поля ФС*** является первичным ключом (табл. 1).

Программа целевой системы на начальной стадии своей работы загружает информацию из файла. Эта информация, по сути, может рассматриваться как таблица, каждая строка которой содержит тип поля ФС и сопутствующую информацию, включая номер строки и номер поля в строке (табл. 2). Для этой таблицы ***тип поля ФС*** можно рассматривать как внешний ключ. Значение ***типа поля ФС*** "связывает" информацию двух таблиц воображаемой реляционной базы данных — первой, содержащей постоянную, и второй, содержащей изменяемую (разработчиком) информацию.

На начальной стадии работы программы целевой системы на основе информации, извлекаемой из описанной взаимосвязанной пары информационных таблиц, для каждого вида ФС заполняется набор изначально пустых (или заполненных по умолчанию) следующих таблиц:

- указателей на функции формирования полей ФС (табл. 3);
- значений длины полей ФС;
- указателей на функции обработки щелчков левой клавишей мыши по полям ФС;
- указателей на функции обработки щелчков правой клавишей мыши по полям ФС;
- указателей на функции обработки нажатия и удерживания правой клавиши мыши на полях ФС.

Структура этих таблиц отражает структуру ФС в том отношении, что ячейки таблиц обслуживают поля ФС, определяемые значениями номера строки и номера поля в строке.

На основной стадии реального режима работы целевой системы, т. е. на стадии функционирования целевой системы по прямому назначению, формирование полей ФС и обработка связанных с ними пользовательских действий осуществляется посредством простых и единообразных функций, которые по номеру строки и номеру поля в строке выбирают функции из готовых таблиц и передают им управление.

Описанный способ обеспечения модифицируемости как совокупности параметров ФС, так и функциональности соответствующего ЧМИ повышает надежность программного кода, функ-

Таблица 2

Задание свойств полей ФС

Номер строки	Номер поля в строке	Тип поля ФС	Длина поля ФС	Обработка		
				щелчка левой клавишей мыши	щелчка правой клавишей мыши	нажатия и удерживания правой клавиши мыши
1 2 2 2 ...	1 1 2 6 ...	Callsign AFL VTrend Empty ...	7 4 1 0 ...	Нет Да Нет Нет ...	Да Нет Нет Нет ...	Да Нет Нет Нет ...

Таблица 3

Результирующая таблица указателей на функции формирования полей ФС

	1	2	3	4	5	6
1 2 3 4	&createCallsign &createAFL	... &createVTrend	... ...	... ...	... ...	... &emptyCreate ... ...

ционирующего на основной стадии реального режима работы целевой системы.

Отметим, что для хранения данных можно использовать отдельные файлы, но более предпочтительно централизованное хранение всей изменяемой информации, относящейся к виду и функциональности компонентов целевой системы, в базах данных, так как в основе последних лежат логические и физические модели данных.

Конфигурирование функциональных свойств системы

Рассмотрим способ обеспечения возможности конфигурирования любых функциональных свойств целевой системы, к которым предъявляется требование модифицируемости. Способ основан на том, что траектория исполнения многовариантного программного кода, обеспечивающего функционирование целевой системы по прямому назначению, генерируется заранее.

Для примера рассмотрим настройку алфавита, используемого при вводе информации в те или иные поля: только русского, только латинского или того и другого.

Параметр-признак с тремя возможными значениями может быть задан, наряду с другими изменяемыми параметрами системы, в рамках ПО, специально предназначенного для конфигурирования свойств. Программа целевой системы, прочитав и распознав этот параметр, в первых двух случаях должна сделать выбор: проводить или не проводить перекодировку введенных с клавиатуры символов для их отображения, в третьем же случае, помимо принятия такого решения, программа должна обеспечить наличие интерфейса выбора языка ввода текста для соответствующей группы полей.

Следовательно, указанный признак должен учитываться программой целевой системы:

- на начальной стадии работы — при создании экранных объектов ЧМИ;
- многократно на основной стадии работы — при обработке ввода каждого символа в соответствующие текстовые поля.

Для того чтобы избежать излишнего дублирования программного кода и повысить его надежность, в рамках ПО целевой системы можно создать три (по числу значений параметра-признака) класса текстовых полей с полиморфным родительским классом. Они будут различаться единственным методом — тем, который реализует обработку ввода символа. Программа целевой системы, анализируя параметр-признак, на начальной стадии своей работы создаст текстовые поля как объекты одного из трех классов.

Разновидность такого подхода заключается в том, что текстовые поля создаются как объекты соответствующего класса стандартной графической библиотеки и сопровождаются указателем на функцию обработки ввода символа. Пишутся три функции обработки с такими же типами параметров и возвращаемого значения, как у указателя. На начальной стадии своей работы программа целевой системы, анализируя параметр-признак, присваивает указателю значение, соответствующее одной из трех функций. Обработка ввода символа в режиме реального времени состоит в вызове функции по содержимому указателя.

Таков способ унификации кода создания экранных объектов ЧМИ в части задания вида обработки ввода символов в текстовые поля.

Подобной унификации, заблаговременно определяющей траекторию исполнения начальной стадии целевой программы, подвергается и код создания интерфейса выбора языка ввода текста. Та или иная функция построения интерфейса будет вызвана в зависимости от содержимого некоторого указателя, зависящего от значения того же параметра-признака.

Очевидно, что сам программный код обработки ввода символа был бы не менее единообразным, чем в рассматриваемом варианте, если бы представлял собой единственную функцию, внутри которой всякий раз выполнялся бы анализ предварительно заданного признака. Однако вследствие того, что требуемая функциональность может изменяться не только от системы к системе, но и в зависимости от нюансов использования системы, эта вызываемая в режиме реального времени универсальная функция может неоправданно усложниться.

Предположим, к примеру, что некий технический ресурс, состояние которого отслеживается системой, может характеризоваться статусом "Занятости". Для индикации этого статуса соответствующий элемент мнемосхемы на экранах операторских рабочих мест может окрашиваться цветом тревоги, и в зависимости от пожелания заказчика, это может относиться ко всем рабочим местам или к тому или иному их подмножеству. Это подмножество, а также сам признак "Полное множество/Подмножество", разумеется, можно задать в виде изменяемых параметров системы, наряду с признаком принципиальной необходимости окрашивания. На основной стадии работы программа целевой системы могла бы определять:

а) значение признака принципиальной необходимости окрашивания;

б) наличие статуса "Занятости" ресурса;

в) значение признака "Полное множество/Подмножество" рабочих мест;

г) наличие типа "своего" рабочего места среди заданного подмножества.

Однако для отдельно взятой системы в основном режиме ее работы условие (б) может изменяться сравнительно часто, условие (г), связанное с конфигурированием рабочих мест системы, — редко, условия (а) и (в) и подмножество из условия (г) — никогда. Поэтому на основную стадию функционирования целевой системы целесообразно возложить лишь анализ статуса ресурса типа (б) (периодический или по событию поступления информации) и анализ статуса рабочего места в отношении этого ресурса типа (г) (по событию конфигурирования рабочих мест) — и то, при адекватных предварительно полученных результатах анализов (а) и, при необходимости, (в).

В рамках описанного выше подхода в составе ПО целевой системы создаются три пары альтернативных функций и, соответственно, три указателя на функции; в каждой паре одна из функций пустая. Функции первой пары выполняются на начальной стадии реального режима работы (непустая функция проводит анализ типа (в)), второй и третьей пар — на основной. Непустые функции формируют содержимое указателей на функции других пар. Тем самым строится траектория исполнения программного кода, исключающая излишний анализ информации. Самой первой на начальной стадии выполняется непарная функция, которая проводит анализ типа (а). В случае если признак принципиальной необходимости окрашивания не установлен, она формирует три указателя на пустые функции, и никакие из перечисленных проверок в основном режиме выполняться не будут.

Для того чтобы ПО целевой системы допускало возможность модификации посредством некоторого ЧМИ, оно должно строиться из модулей, которые не только предоставляют операции, обеспечивающие доступ к их состоянию, но, к тому же, отражают стабильные абстракции предметной области.

Однако в отдельных случаях наглядное, ясное и выразительное представление вариантов функциональности в ЧМИ ПО, предоставляемого разработчику, создать трудно. Тогда описанная методика использования набора альтернативных функций и соответствующего им указателя может быть реализована разработчиком ПО непосредственно на стадии создания программного кода. При этом преимущества хорошей понимаемости и надежности программ сохраняются.

Прозрачность программного кода для восприятия выигрывает, если формирование всех подобных указателей на функции не только выполняется по возможности "в одно и то же время" в начале работы ПО целевой системы, но и сосредоточено в "одном и том же месте" в рамках одного файла.

Формирование траектории исполнения многовариантного программного кода практически оказывается более удобным для разработчика, чем условная компиляция, — при том, что оба подхода обеспечивают возможность работы с одним набором файлов для ПО нескольких систем взамен работы с семейством таких наборов.

Заключение

Очевидно, что описанный Конструктор ФС по своему назначению подобен инструментам интерактивного проектирования визуальных объектов в универсальных средах объектно-ориентированного программирования. Однако благодаря тому, что он несет в себе специфику типа объекта, его использование не требует мысленного перехода от универсальных категорий к категориям предметной области, выбора в рамках универсальной среды подходящих средств и разработки методики конструирования.

Специализированный инструментарий, по определению, удобнее в использовании, чем универсальный. Однако его возможности всегда ограничены представлением авторов инструментария о целевом продукте проектирования.

Попытки создать всеобъемлющую среду проектирования специализированных систем управления, которая предназначалась бы для всего сообщества разработчиков определенного класса систем и представляла собой коммерческий продукт [4], мало результативны, по крайней мере, по двум причинам. Во-первых, технический прогресс в большой степени препятствует учету многообразия потребностей сегодняшнего и завтрашнего дня в инструментарии, сам срок разработки которого велик. Во-вторых, фантазии конкурирующих между собой разработчиков в их стремлении усовершенствовать функциональность систем многообразны и выходят за рамки возможностей созданного ранее специализированного инструментария.

Поэтому на сегодняшний день средства, повышающие эффективность и качество проектирования специализированных систем управления, целесообразно создавать и применять лишь для нужд конкретного предприятия, и при этом не стоит затрачивать усилия на создание "всеядной" среды проектирования — вариативность свойств систем и их преемственность в этом случае очерчены достаточно определенно.

Список литературы

1. **Соммервилл И.** Инженерия программного обеспечения. М.: Издательский дом "Вильямс", 2002. 624 с.

2. **Лаврищева Е. М., Петрухин В. А.** Методы и средства инженерии программного обеспечения. М.: МФТИ, 2006. 304 с.

3. **Подольская Н. Н.** Алгоритмы автоматического отброса формуляров для интерактивных графических приложений // Информационные технологии. 2007. № 9. С. 45—50.

4. **OPS Toolbox.** Development toolbox for operational display systems. // http://www.barco.com/barcoview/downloads/ODSToolbox5.3.pdf.

УДК 004.4'422

А. Н. Филиппов, науч. сотр.,
ЗАО "МЦСТ",
e-mail: filipov@mcst.ru

Метод нумерации значений и использование его результатов при оптимизации программ

Рассматривается вопрос ускорения работы программ за счет их оптимизации на этапе компиляции. Описан метод анализа промежуточного представления и связанные с ним оптимизирующие преобразования, реализованные в промышленном оптимизирующем компиляторе. Исследовано влияние описанной оптимизации на время исполнения ряда задач.

Ключевые слова: *оптимизирующий компилятор, нумерация значений, анализ потока данных, скалярная оптимизация, удаление общих подвыражений.*

Введение

Для некоторых оптимизирующих преобразований требуется дополнительная информация, получаемая в процессе предварительного анализа программы. Одним из видов такого анализа является нумерация значений (*value numbering*), результатом которой является разбиение множества операций промежуточного представления на классы эквивалентности, называемые также классами конгруэнтности. Две операции принадлежат одному классу конгруэнтности только в том случае, если компилятору удалось доказать, что они всегда вырабатывают одинаковый результат.

Настоящая работа посвящена анализу методом нумерации значений. Целью работы является рассмотрение возможностей эффективного применения нумерации значений в составе оптимизирующего компилятора. В работе описан ряд встречающихся в литературе алгоритмов анализа и приведены необходимые для них аналитические структуры. Кроме того, рассмотрены возможности использования результатов анализа рядом оптимизирующих преобразований.

Алгоритм анализа и использующие его оптимизирующие преобразования были реализованы в рамках оптимизирующего компилятора для архитектур Эльбрус-3М, Эльбрус-90 и Sparc. Проведенные замеры производительности позволяют говорить о практической ценности нумерации значений.

Основные понятия и определения

Для проведения нумерации значений нам понадобятся некоторые надстройки над промежуточным представлением программы. Опишем важнейшие из них.

Линейным участком программы назовем совокупность операций промежуточного представления с выделенными входной и выходной операциями. Передача управления на линейный участок может осуществляться только через входную операцию. В свою очередь, передача управления из линейного участка осуществляется только через выходную операцию.

Граф управления (*control flow graph*, CFG) — аналитическая структура, являющаяся управляющей надстройкой над промежуточным представлением. Граф управления представляет собой направленный связный граф, вершинам которого соответствуют линейные участки, а дугам — управляющие связи между ними, отображающие передачу управления.

Форма статического единственного присваивания (*static single assignment*, SSA) является одной из самых распространенных форм представления программы и активно используется в большинстве современных оптимизирующих компиляторов [1]. В программе, представленной в SSA-форме, для любой переменной существует единственная операция, ее вырабатывающая. Для того, чтобы соблюсти это ограничение и тем самым перевести программу в SSA-форму, необходимо выполнить следующие действия:

- разместить в точках схождения потока управления φ-функции; φ-функция для некоторой переменной — это псевдооперация, выбирающая среди множества значений переменной нужное;
- переименовать все переменные так, чтобы каждому определению соответствовала своя уникальная переменная.

Благодаря SSA-форме для любого аргумента, читающего значение некоторой переменной, всегда можно найти одно и только одно определение этой переменной.

Определение всегда *доминирует* над использованием, т. е. все пути программы от ее начала до точки использования переменной проходят через ее определение. На практике переименование, требуемое канонической SSA-формой, затрудняет распределение переменных на регистры, кроме того, поддержка SSA-формы при проведении каждого оптимизирующего преобразования требует значительных усилий.

Однако же главное свойство SSA-формы (возможность для любого аргумента получить доминирующее определение) очень полезно. Поэтому мы

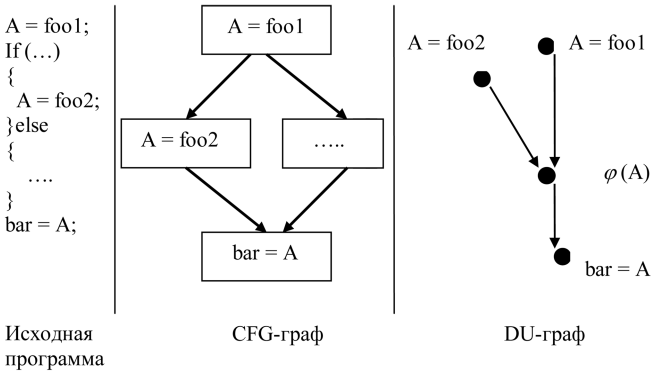


Рис. 1. Пример построения CFG- и DU-графов

будем использовать другую структуру данных, обладающую тем же свойством — *глобальный граф определений и использований* (Def-Use, DU-граф) [2].

DU-граф представляет собой направленный граф с узлами двух типов. Одни узлы соответствуют операциям промежуточного представления, другие являются φ-узлами. Дуги DU-графа отражают потоковые связи между узлами. Другими словами, узел-преемник дуги потребляет результат, выработанный узлом-предшественником,

В дугах DU-графа и в φ-узлах сохраняются ссылки на переменные, для которых они строятся. Операция может обращаться к некоторому множеству переменных, поэтому в узел DU-графа, который соответствует операции, могут входить дуги для разных переменных. Свойства φ-узлов DU-графа полностью аналогичны свойствам φ-функций SSA-представления. Все входящие в φ-узел дуги взаимнооднозначно соответствуют входящим в узел управляющего графа дугам. Построенные φ-узлы сохраняются в списке узла управляющего графа. В свою очередь, в каждом φ-узле сохраняется ссылка на узел управляющего графа, которому он соответствует.

Для отображения неинициализированных переменных в DU-графе создается φ-узел неопределенного типа. У этого узла переменная не указывается. На него замыкаются все неинициализированные использования переменных. На рис. 1 представлен пример построения CFG и DU-графов.

Забегая вперед, отметим, что практически все известные алгоритмы нумерации значений базируются на SSA-форме, тогда как мы будем работать на DU-графе. Существенных отличий в алгоритмах это не вызовет, однако использование DU-графа представляется автору более удобным на практике.

Алгоритмы нумерации значений

Нумерация значений на линейном участке. Одним из первых и наиболее простых способов нумерации значений можно считать алгоритм, представленный в работе [3]. Простота алгоритма обусловлена тем, что базисом для анализа выбран линейный участок. Другими словами, необходимым условием эквивалентности операций является принадлежность их одному и тому же узлу CFG-графа. Ниже представлен алгоритм проведения такого анализа.

Создать хэш-таблицу классов конгруэнтности для данного линейного участка;
Цикл по всем операциям линейного участка, начиная со стартовой
  Создать вектор key_vect и добавить в него имя операции;
  Цикл по всем аргументам операции
    Если предшественник аргумента в DU-графе — операция, принадлежащая этому же линейному участку
      arg_con_class = класс конгруэнтности предшественника;
    иначе
      arg_con_class = новый класс конгруэнтности;
    добавить в вектор key_vect значение arg_con_class;
  Конец цикла
  Найти в хэш-таблице запись, соответствующую key_vect;
  Если найдено
    присвоить операции класс конгруэнтности, найденный в таблице;
    иначе
      создать новый класс конгруэнтности и **присвоить** его операции;
      создать в таблице запись с ключом key_vect и положить в качестве клиентской ссылки созданный класс;
Конец цикла

Вначале для линейного участка создается хэш-таблица номеров значений. В качестве ключа в ней выступает вектор, состоящий из имени операции и классов конгруэнтности ее аргументов, а в поле записи содержится ссылка на класс конгруэнтности.

Обход линейного участка начинается со стартовой операции последовательно вниз, пока не будет достигнута конечная операция. Такое правило обхода гарантирует нам, что к моменту рассмотрения конкретной операции всем ее аргументам будут присвоены свои классы конгруэнтности. Здесь под классом конгруэнтности аргумента понимается класс конгруэнтности операции, вырабатывающей этот аргумент или, если такой операции в пределах линейного участка нет, новый произвольный класс конгруэнтности. В случае же, если аргумент операции — константа, в качестве класса конгруэнтности аргумента используется значение этой константы.

Условия принадлежности операций одному классу можно записать следующим образом:

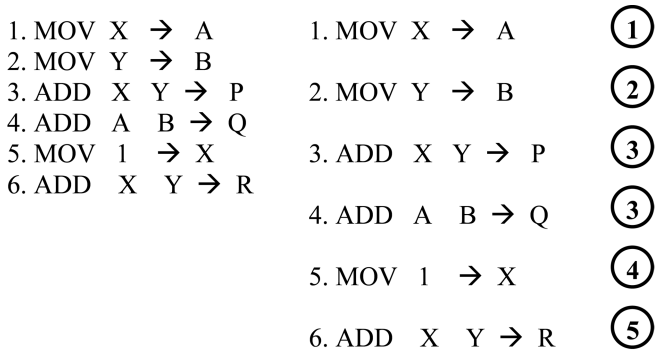


Рис. 2. Пример вычисления классов конгруэнтности операций на линейном участке

- операции принадлежат одному линейному участку;
- операции имеют одно и то же имя;
- аргументы операций имеют одинаковые классы конгруэнтности.

Отдельно стоит выделить операцию пересылки. Ей попросту присваивается класс конгруэнтности ее аргумента.

Описанный алгоритм нумерации значений можно легко расширить, если в качестве базиса выбрать множество линейных участков (B_1, ..., B_n) таким образом, что B_i является единственным предшественником B_{i+1} в графе управления. Обход в этом случае начинается со стартовой операции B_1 и заканчивается конечной операцией B_n. Требование единственности предшественника каждого B_i ($i > 1$) гарантирует наличие вычисленных классов конгруэнтности ее аргументов к моменту рассмотрения каждой конкретной операции.

Продемонстрируем работу алгоритма на несложном примере (рис. 2). В правой части рисунка для каждой операции справа от нее приведен номер класса конгруэнтности, присвоенный алгоритмом. Так, например, можно увидеть, что операции 3 и 4 будут признаны эквивалентными. Что касается операции 6, то она получит другой класс конгруэнтности, поскольку перед ней есть перезапись переменной *X*.

Нумерация значений на произвольном ациклическом участке. Прежде чем приступить к описанию алгоритма, введем понятие *RPO-нумерации* узлов графа (*reverse post order*, нумерация от конца к началу). Рассмотрим ациклический направленный граф с выделенным стартовым узлом, содержащий *N* узлов. *Топологической сортировкой* узлов (также известной как *RPO-нумерация* или *N-нумерация*) ациклического графа называется алгоритм, присваивающий узлам графа номера от 1 до *N* таким образом, что все предшественники узла с номером *M* имеют номера меньше *M*. Существует множество алгоритмов проведения RPO-нумерации, например, алгоритм, описанный в работе [1].

Далее мы представим алгоритм нумерации значений на произвольном ациклическом участке CFG-графа. Одна из версий этого алгоритма приведена в работе [5], здесь мы лишь адаптируем ее применительно к DU-графу.

Создать хэш-таблицу классов конгруэнтности
Цикл по всем CFG-узлам ациклического участка в порядке RPO-нумерации
  Цикл по всем φ-узлам данного CFG-узла и операциям, начиная со стартовой
    Если рассматриваем операцию
      Создать вектор key_vect и **добавить** в него имя операции;
      Цикл по всем аргументам операции
        arg_pred = предшественник операции по данному аргументу в DU-графе
        arg_con_class = класс конгруэнтности arg_pred;
        добавить в вектор key_vect значение arg_con_class;
      Конец цикла
    иначе (рассматриваем φ-узел)
      Создать вектор key_vect и **добавить** в него CFG-узел;
      Цикл по всем входящим в CFG-узел дугам
        du_edge = DU-дуга, входящая в φ-узел, соответствующая данной CFG-дуге;
        arg_pred = предшественник du_edge;
        arg_con_class = класс конгруэнтности arg_pred;
        добавить в вектор key_vect значение arg_con_class;
      Конец цикла
    Найти в хэш-таблице запись, соответствующую key_vect;
    Если найдено
      присвоить операции или φ-узлу класс конгруэнтности, найденный в таблице;
    иначе
      создать новый класс конгруэнтности и **присвоить** его операции или φ-узлу;
      создать в таблице запись с ключом key_vect и положить в качестве клиентской ссылки созданный класс;
  Конец цикла
Конец цикла

В отличие от предыдущего алгоритма, где для каждого CFG-узла создавалась своя хэш-таблица классов конгруэнтности, здесь достаточно одной таблицы на весь ациклический участок.

В соответствии с алгоритмом две операции принадлежат одному классу конгруэнтности если:

- они принадлежат одному ациклическому участку;
- они имеют одно и то же имя;
- их аргументы имеют одинаковые классы конгруэнтности.

Два φ-узла принадлежат одному классу конгруэнтности если:

- они принадлежат одному и тому же узлу CFG-графа;
- все их DU-предшественники попарно эквивалентны в плане классов конгруэнтности. Здесь слова "попарно эквивалентны" означают эквивалентность предшественников, соответствующих одной CFG-дуге.

Заметим, что в момент рассмотрения каждой операции (или φ-узла) классы конгруэнтности всех ее DU-предшественников уже вычислены. Это требование достигается за счет ацикличности исходного участка программы и организации обхода узлов CFG-графа в порядке их RPO-нумерации.

Можно привести другую модификацию описанного выше алгоритма. А именно, можно обходить в порядке RPO-нумерации не CFG-, а DU-граф. Суть алгоритма не изменится: для каждого DU-узла создается ключ, по которому можно однозначно установить эквивалентность узлов друг другу. Запись с этим ключом ищется в таблице. В случае, если таковая найдена, DU-узлу присваивается класс конгруэнтности, соответствующий найденной записи. В противном случае DU-узлу присваивается новый созданный класс конгруэнтности, и затем этот класс с вычисленным ключом вносится в таблицу.

Результаты работы обеих модификаций будут идентичны как в плане полученного результата, так и в отношении затраченного времени и памяти.

Универсальный алгоритм нумерации значений. Универсальный алгоритм проведения нумерации значений на произвольном представлении впервые представлен в работе [5]. В этой части статьи будет описана полная версия алгоритма, работающая на DU-графе.

Рассмотрим произвольный граф, содержащий циклические пути. Независимо от конкретного алгоритма некоторые дуги этого графа не будут удовлетворять требованиям топологической сортировки (то есть будут иметь предшественника с номером меньшим или равным номеру преемника) — эти дуги называют *обратными*. Понятно, что для конкретного графа существует множество нумераций, удовлетворяющих требованиям топологической сортировки, и каждая из них может обнаружить разный набор обратных дуг.

Очевидно, что далеко не для каждой программы ее DU-граф будет ациклическим. Как следствие, наличие обратных дуг нарушает основное требование приведенных выше алгоритмов нумерации значений, так как теперь в момент вычисления класса конгруэнтности операции (или φ-узла) часть ее аргументов могут не иметь своих классов конгруэнтности.

Введем понятие *сильно связанной компоненты* (Strongly Connected Component — SCC). Нетривиальная сильно связанная компонента есть подмножество узлов графа таких, что каждый достижим из каждого (в том числе и каждый из себя самого). При этом если два узла достижимы друг из друга, то они должны принадлежать одной компоненте. Под тривиальной сильно связанной компонентой будем понимать компоненту, состоящую из единственного узла графа. Применительно к DU-графу тривиальной компонентой будет являться каждая операция или φ-узел, не входящие ни в одну нетривиальную SCC. Одним из наиболее удобных и распространенных алгоритмов поиска SCC для произвольного графа является алгоритм Тарьяна [1].

Если теперь мы факторизуем DU-граф таким образом, что каждая сильно связанная компонента получит единственного представителя (стянем SCC в один узел), то наш граф станет ациклическим. Назовем *головой* сильно связанной компоненты узел, имеющий максимальный из всех входящих в нее узлов RPO-номер. Каждой сильно связанной компоненте присвоим номер ее головы. Такая нумерация факторизованного графа, в свою очередь, тоже удовлетворяет требованиям топологической сортировки. В таком случае к этому графу можно применить алгоритм нумерации значений, описанный в предыдущем пункте. Действительно, ацикличность факторизованного графа и обход согласно топологической сортировке гарантируют, что к моменту рассмотрения каждой сильно связанной компоненты все предшественники дуг, входящих в эту компоненту, будут уже обработаны.

Теперь рассмотрим вопрос проведения нумерации значений внутри нетривиальной сильно связанной компоненты. Ниже представлен итерационный алгоритм обработки отдельной SCC.

Присвоить всем операциям SCC начальный класс конгруэнтности;
Создать хэш-таблицу классов конгруэнтности для данной SCC;
(*)
i_changed = FALSE;
Цикл по всем DU-узлам SCC-порядке RPO-нумерации
  Если рассматриваем операцию
    Создать вектор key_vect и **добавить** в него имя операции;

    Цикл по всем аргументам операции
      arg_pred = предшественник операции по данному аргументу в DU-графе
      arg_con_class = класс конгруэнтности arg_pred;
      добавить в вектор key_vect значение arg_con_class;
    Конец цикла

```
    иначе (рассматриваем φ-узел)
        cfg_node = CFG-узел, соответствующий
        φ-узлу
        Создать вектор key_vect и добавить в него
        cfg_node;
        Цикл по всем входящим в cfg_node дугам
            du_edge = DU-дуга, входящая в φ-узел,
            соответствующая данной CFG-дуге;
            arg_pred = предшественник du_edge;
            arg_con_class = класс конгруэнтности
            arg_pred;
            добавить в вектор key_vect значение
            arg_con_class;
        Конец цикла
    con_class = взять класс конгруэнтности
    DU-узла;

    Найти в хэш-таблице запись, соответствующую
    key_vect;
    Если найдено
        Если найденный класс не совпадает
        с con_class
            is_changed = TRUE;
        присвоить DU-узлу класс конгруэнтности,
        найденный в таблице;
    иначе
        создать новый класс конгруэнтности и
        присвоить его DU-узлу;
        создать в таблице запись с ключом key_vect и
        положить в качестве клиентской ссылки соз-
        данный класс;
        is_changed = TRUE;
Конец цикла

Если (is_changed = = TRUE)
    goto (*);
```

Наличие обратных дуг внутри нетривиальной SCC не позволяет нам достоверно утверждать, что все предшественники DU-дуг будут обработаны раньше их преемников. Поэтому в начале алгоритма мы присвоили всем узлам, входящим в SCC, один и тот же, начальный класс конгруэнтности. В процессе работы будем следить, изменился ли класс конгруэнтности хотя бы одного DU-узла из SCC. В случае, если это произошло (параметр is_changed примет значение TRUE) — придется повторить процедуру заново.

Алгоритм останавливается тогда, когда каждый узел SCC получит свой класс конгруэнтности, который не будет изменен на последней итерации. В работе [5] показано, что остановка алгоритма произойдет гарантированно. Более того, доказано, что число итераций алгоритма не превосходит D(SSC) + + 2, где D(SSC) — максимальное число обратных дуг по всем ациклическим путям внутри SCC.

В начале работы описанного алгоритма делается "оптимистическое" предположение, что все операции SCC имеют один и тот же класс конгруэнтности. Поэтому хэш-таблица, используемая в алгоритме, получила название *optimistic*-таблицы. На практике, можно не создавать отдельную *optimistic*-таблицу для каждой SCC. Достаточно вначале работы создать одну таблицу и использовать ее для всех SCC.

Итак, теперь мы можем привести окончательный алгоритм нумерации значений для произвольного промежуточного представления.

```
Найти список сильно связанных компонент DU-
графа (алгоритм Тарьяна);
Упорядочить список SCC в порядке возрастания
номеров их голов;
Присвоить всем DU-узлам начальный класс кон-
груэнтности;
Создать valid-версию хэш-таблицы;
Создать optimistic-версию хэш-таблицы;

Цикл по упорядоченному списку сильно связан-
ных компонент
    Если SCC тривиальна
        вычислить класс конгруэнтности узла, ис-
        пользуя valid-версию хэш-таблицы;
    иначе (SCC нетривиальна)
        вычислить классы конгруэнтности узлов,
        входящих в SCC, используя optimistic-версию
        хэш-таблицы;
        вычислить классы конгруэнтности узлов,
        входящих в SCC, используя valid-версию
        хэш-таблицы;
Конец цикла
```

Алгоритм получился довольно простым. Вначале ищется список всех сильно связанных компонент DU-графа, который упорядочивается в порядке возрастания RPO-номеров их голов. После этого всем узлам DU-графа присваивается один и тот же, начальный класс конгруэнтности. Затем создаются две версии хэш-таблицы, называемые *pessimistic* (или *valid*) и *optimistic*. *Pessimistic*-версия используется для обработки DU-узлов, при условии, что все предшественники узла уже обработаны и имеют свой класс конгруэнтности. Что касается *optimistic*-версии, то она используется в итерационной части алгоритма, которая представлена ранее. Далее мы обходим упорядоченный список сильно связанных компонент и вычисляем классы конгруэнтности узлов, входящих в каждую компоненту. В случае тривиальной SCC мы можем использовать *pessimistic*-версию хэш-таблицы, так как доподлинно известно, что все аргументы единственного узла из SCC уже обработаны. В случае нетривиальной SCC мы сначала задействуем *optimistic*-версию таблицы и только потом, когда итерационный алгоритм закончит свою работу, используем *pessimistic*-таблицу.

Использование результатов нумерации значений в оптимизирующих преобразованиях

В этой части мы рассмотрим возможности использования результатов нумерации значений в оптимизирующих преобразованиях программ.

Глобальный сбор общих подвыражений. Одним из наиболее распространенных оптимизирующих преобразований, использующих результаты *Value Numbering*, является глобальный сбор общих подвыражений (*global common subexpressions elimination* — GCSE). Цель данного преобразования — удаление избыточных вычислений в теле программы. Назовем операцию промежуточного представления избыточной, если вырабатываемый ею результат уже был выработан ранее по ходу программы некоторой другой операцией. Очевидно, что избыточную операцию можно удалить, используя вместо ее результата результат, выработанный ранее. Далее представлен алгоритм выявления избыточных операций и φ-узлов.

Положить $avail_B = \varnothing$ для каждого CFG-узла B
Цикл по всем CFG-узлам в порядке RPO-нумерации
  $avail_B = \cap avail_C$, где пересечение производится по всем предшественникам узла B, за исключением предшественников по обратным дугам
  Цикл по всем φ-узлам и операциям линейного участка
    con_class = класс конгруэнтности операции или φ-узла
    Если con_class $\in avail_B$
      Занести операцию или φ-узел в список избыточных
    Иначе
      Занести con_class в множество $avail_B$
  Конец цикла
Конец цикла

Алгоритм ставит в соответствие каждому линейному участку B множество $avail_B$, в котором содержатся классы конгруэнтности, доступные (уже вычисленные) в данной точке программы. В начале работы для всех CFG-узлов указанное множество полагается пустым. Перед началом анализа каждого конкретного CFG-узла B его множество $avail_B$ инициализируется пересечением соответствующих множеств каждого из предшественников B в CFG-графе, за исключением предшественников по обратным дугам. Действительно, для того, чтобы класс конгруэнтности был доступен в начале узла, он должен быть доступен по каждому из путей, в этот узел входящих.

Далее, обходя линейный участок сверху вниз, мы заносим в множество *avail* классы конгруэнтности всех встретившихся операций и φ-узлов. В случае, если класс конгруэнтности уже присутствует в нашем множестве, операция или φ-узел полагаются избыточными.

Удаление частичных избыточностей. Еще одним примером оптимизации, использующей результаты нумерации значений, является удаление частичных избыточностей (*partial redundancy elimination* — PRE). Она сочетает в себе сбор и удаление общих подвыражений, вынос инвариантного кода из циклов и перемещение вычислений из часто исполняемых участков программы в менее часто исполняемые.

Назовем вычисление *частично избыточным*, если оно избыточно по некоторым, но не обязательно всем, путям исполнения. Примером частично избыточного выражения можно считать инвариантный код в цикле: инвариантная операция, не являясь полностью избыточной, тем не менее, избыточна для пути, проходящего через обратную дугу цикла.

Алгоритм проведения PRE является достаточно сложным и выходит за рамки данной работы. Подробно ознакомиться с ним можно в работах [4, 6]. Здесь мы лишь проиллюстрируем результат работы PRE на простом примере.

Рассмотрим фрагмент CFG-графа, изображенный в левой части рис. 3. На рисунке видно, что вычисление переменной Z в блоке B4 не является полностью избыточным, так как не избыточно по пути B1—B2—B4, и просто удалить его нельзя. В то же время это вычисление избыточно по пути B1—B3—B4.

Оптимизация PRE устраняет эту неоптимальность способом, представленным в правой части рисунка. Действительно, мы видим, что в результате преобразования число операций вдоль пути

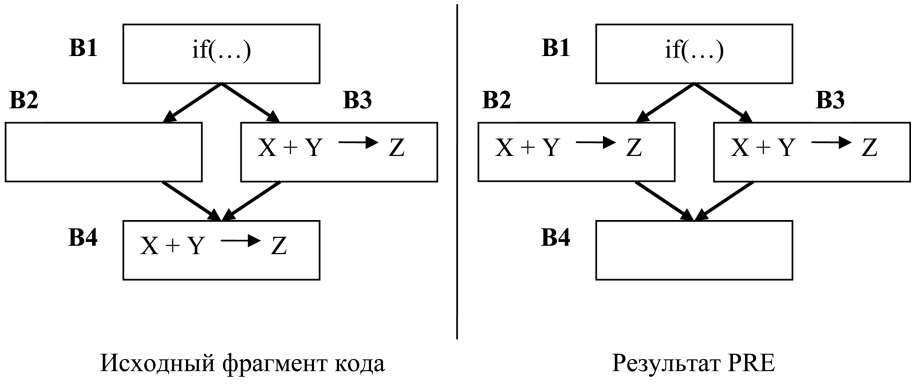


Рис. 3. Пример работы оптимизации удаления частичных избыточностей

B1—B2—B4 осталось неизменным, между тем как число операций вдоль пути B1—B3—B4 уменьшилось.

Результаты экспериментов

Описанный в работе алгоритм нумерации значений, базирующийся на сильно связанных компонентах, был реализован в промышленном оптимизирующем компиляторе для архитектур Эльбрус-3М, Эльбрус-90 и Sparc. Сам по себе этот анализ бесполезен, поэтому в компиляторе были также реализованы оптимизирующие преобразования GCSE и PRE. Об эффективности работы этих преобразований можно судить по приведенным на рис. 4 замерам производительности.

На рисунке представлено влияние GCSE и PRE на некоторые тесты пакета Spec95. В процессе эксперимента для каждой задачи определялось отношение времени ее исполнения, полученного при компиляции без этих преобразований, к времени исполнения, соответствующему компиляции с применением техники GCSE и PRE.

Отметим, что указанные оптимизирующие преобразования несут в себе не только положительные, с точки зрения производительности, качества. Например, к отрицательным качествам PRE можно отнести то, что оно увеличивает давление на регистры, так как переносит вычисления "вверх" на столько, на сколько это возможно. Что касается GCSE, то оно удлиняет время жизни результатов операций, взамен сокращая время жизни их аргументов. В частности, можно заметить, что на тесте **li** эффект от преобразований отрицательный.

Заключение

В работе представлены результаты комплексного исследования различных аспектов метода нумерации значений. Описан ряд алгоритмов анализа: простейшая нумерация значений на линейном участке, более сложный алгоритм на ациклическом регионе и, наконец, алгоритм, работающий на общем графе определений и использований, пригодном для работы с любой программой. Кроме того, предложены варианты практического использования результатов анализа в оптимизирующих преобразованиях, таких как сбор общих подвыражений и удаление частичных избыточностей. Экспериментально установлено, что указанные преобразования в большинстве случаев крайне положительно влияют на время работы программ. Так, на наборе из 8 тестов пакета SPEC95 время работы в среднем сократилось на 10 %, а на некоторых задачах ускорение достигло 80 %.

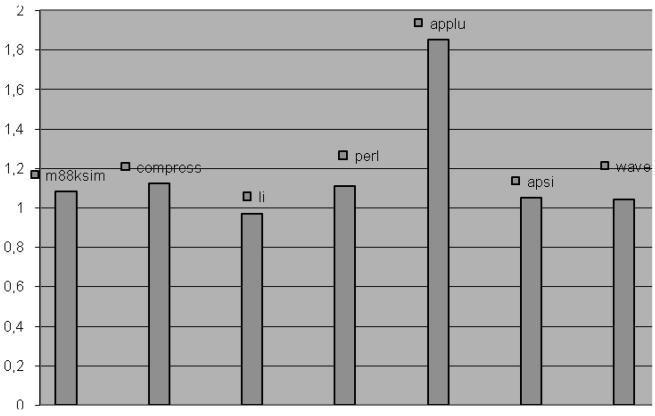


Рис. 4. Результаты измерений производительности

Некоторые представляющие значительный интерес аспекты нумерации значений, оставшиеся за рамками данной работы, будут рассмотрены в дальнейшем. В частности, автор планирует описать технику нумерации значений для операций обращения к памяти. Кроме того, планируется задействовать в анализе технику *ps*-форм, представляющих собой полином вида

$$c_0 + c_1 x_{11} x_{12} \ldots x_1 x_{1k_1} + \ldots + c_n x_{n1} x_{n2} \ldots x_{nk_n}.$$

Как показывает практика, применение *ps*-форм при вычислении классов конгруэнтности операций приводит к более полным результатам анализа и, как следствие, влечет за собой более эффективное применение оптимизирующих преобразований.

Список литературы

1. **Muchnick, Steven S.** Advanced Compiler Design and Implementation. San Francisco: Morgan Kauffman, 1997.

2. **Дроздов А. Ю., Новиков С. В., Боханко А. С., Галазин А. Б.** Глобальный граф потока данных и его роль в проведении оптимизирующих преобразований программ // Высокопроизводительные вычислительные системы и микропроцессоры. Сб. науч. тр. Вып. 8. М.: Изд. ИМВС РАН, 2005. С. 78—87.

3. **Cocke J., Schwartz J. T.** Programming languages and their compilers: Preliminary notes / Technical report. Courant Institute of Mathematical Science, New York University, 1970.

4. **Филиппов А. Н., Шлыков С. Л.** Удаление частичных избыточностей // Высокопроизводительные вычислительные системы и микропроцессоры. Сб. науч. тр. Вып. 9. М.: Изд. ИМВС РАН, 2006. С. 49—57.

5. **Simpson L. T.** Value-Driven Redundancy Elimination / Ph. D. Thesis, Rise University, Houston, Texas, 1996.

6. **Briggs P., Cooper K. D.** Effective Partial Redundancy Elimination. Department of Computer Science. Rice University, Houston, Texas. 1994.

УДК 002.55

А. С. Зуев, канд. техн. наук, ст. препод.,
О. Б. Кучеров, студент,
e-mail: zuev_andrey@mail.ru,
Московский государственный университет
приборостроения и информатики

Модификация принципов работы с дочерними окнами программ, панелями инструментов, главными и контекстными меню*

Представлены результаты модификации принципов работы с дочерними окнами программ, вызываемыми с помощью элементов интерфейса "главное меню", "контекстное меню" и "панель инструментов". Предложенные улучшения реализованы в модели WordModel, для которой приведено описание и сравнение с редакторами MS Word2003 и MS Word2007.

Ключевые слова: *человеко-компьютерное взаимодействие, графический пользовательский интерфейс, оптимизационное геометрическое проектирование, эргономика программного обеспечения, проектирование графических интерфейсов, дочерние окна программ, главное меню, контекстное меню, панель инструментов, оптимизация графических интерфейсов.*

Введение

Неотъемлемой частью программного обеспечения (ПО), используемого в интерактивном режиме, является графический пользовательский интерфейс (ГПИ) — средство человеко-компьютерного взаимодействия, отвечающее за предоставление пользователю средств реализации функциональных возможностей программ. В настоящее время в большинстве видов профессиональной деятельности различных специалистов применяется специализированное ПО, призванное не только автоматизировать выполнение операций с данными, но и создавать для пользователя комфортные условия работы в плане удобства доступа к информации и средствам ее обработки. В зарубежной практике производства ПО качество и эргономичность ГПИ давно являются важными критериями оценки программных продуктов.

Крупные компании, занимающиеся разработкой ПО, ведут исследования в области человеко-компьютерного взаимодействия (*Human Computer Interaction* — HCI) и совершенствуют ГПИ разрабатываемых ими программных средств, стремясь обеспечить им дополнительные конкурентные преимущества. Наглядным примером является деятельность корпорации Microsoft, которая за последние 10 лет три раза существенно модернизировала интерфейс выпускаемой ею операционный системы Windows.

Показателем эффективности ГПИ могут являться затраты времени пользователя на выполнение определенных наборов операций с информацией посредством воздействия на элементы интерфейса. В свою очередь, данные затраты времени обусловлены необходимостью перемещения курсора с помощью некоторого средства манипулирования (например, "мыши") между элементами ГПИ на расстояния, определяемые их расположением относительно друг друга [1]. Поэтому в общем виде постановка задачи оптимизации ГПИ может быть сформулирована следующим образом: элементы интерфейса требуется разместить в окнах программы таким образом, чтобы суммарное расстояние выполняемых пользователем перемещений курсора было минимально, а получаемый результат (интерфейс) был эргономичен — понятен, удобен и прост в эксплуатации. В данном контексте под эргономичностью понимается четкая последовательность и непрерывность перемещения фокусировки внимания пользователя между элементами ГПИ.

Оптимальность ГПИ по представленному критерию может быть достигнута в результате применения методов оптимизационного геометрического проектирования для размещения элементов интерфейса и окон программы относительно друг друга [2]. Вместе с тем, разработка новых технических решений по организации человеко-компьютерного взаимодействия может позволить исключить необходимость выполнения пользователем некоторых перемещений курсора и ослабить отрицательные последствия его ошибок при выполнении манипуляций с элементами интерфейса. Таким образом, проектирование ГПИ сочетает формализацию, представленную оптимизационным геометрическим проектированием, и творческий процесс, реализуемый специалистами по HCI над результатами, получаемыми с применением соответствующих математических методов.

На основании представленного выше материала будем рассматривать эффективность ГПИ с точки зрения количества и расстояний требующихся перемещений курсора — числа и сложности воздействий, осуществляемых пользователем, на элементы интерфейса при выполнении определенного набора операций. Целевой функцией, значение которой требуется минимизировать, будет являться суммарное расстояние перемещений курсора при выполнении определенного набора последовательных воздействий на элементы ГПИ.

В работах [1—3] изложены теоретические исследования, позволившие получить практические результаты и модели интерфейсов [4, 5], по многим параметрам превосходящие современные передовые разработки. В данной статье продолжено изложение результатов применения методов оптимизационного геометрического проектирования к разработке и совершенствованию ГПИ [6]. Результаты решения задачи оптимизации ГПИ представлены на примере модификации принципов работы с дочерними окнами программ, вызываемыми с помощью элементов интерфейса "главное меню", "контекстное меню" и "панель инструментов". Представлено описание модели WordModel, разработанной авторами на основе текстового редактора MS Word с использованием полученных результатов, приведено ее сравнение с версиями MS Word, выпущенными в 2003 и 2007 годах.

* Работа выполнена при финансовой поддержке грантов Президента РФ молодым российским ученым-кандидатам наук (грант № МК-948.2008.9).

Описание предложенной модификации и разработанной модели

В современных программных продуктах множество опций (функциональных возможностей) представлено в главном меню, располагающемся в верхней части окна программы. На рис. 1 представлен вид главного меню текстового редактора MS Word 2003.

Так как область окна программы ограничена, а главное меню не может содержать все опции, то некоторым пунктам списков главного меню (например, *Сохранить как* на рис. 1) соответствуют дочерние окна (рис. 2).

Главное меню является типовым элементом библиотек визуальных компонентов объектно-ориентированных языков программирования. При воздействии на пункт списка, соответствующего разделу меню, список закрывается, а вызываемое дочернее окно отображается либо в центре экрана монитора (по умолчанию), либо в том положении, где его разместил пользователь при предыдущем обращении к нему. Данный принцип вызова дочерних окон связан с нарушением фокусировки внимания пользователя, наблюдаемым в момент закрытия списка и отображения окна.

Описанная особенность вызова дочерних окон имеет следствие, заключающееся в том, что ошибка пользователя при выборе пункта списка влечет за собой необходимость повторного выполнения всех действий, начиная с воздействия на раздел главного меню. На рис. 3 представлен пример графа, описывающего работу пользователя со списком, соответствующим разделу меню (*Файл*, см. рис. 1), и дочерним окном, соответствующим пункту данного списка (*Сохранение документа*, см. рис. 2). Вершины графа обозначают элементы интерфейса, на которые воздействует пользователь, а дуги — расстояния перемещения курсора в экранных пикселях.

В графе, изображенном на рис. 3, использованы следующие обозначения вершин, соответствующих элементам интерфейса на рис. 1 и 2:

- $v_{0,1}$ — раздел главного меню (*Файл*);
- $v_{1,1}$, $v_{1,2}$, $v_{1,3}$ — пункты списка, соответствующего разделу меню (*Создать*, *Открыть*, *Сохранить как*);
- $v_{3,1}$ — дочернее окно (*Сохранение документа*), соответствующее пункту списка $v_{1,3}$ (*Сохранить как*).

На рис. 3 дугами графа обозначены расстояния перемещения курсора:

- $e_{1,1}$, $e_{1,2}$, $e_{1,3}$ — от раздела меню к соответствующим пунктам списка;
- $e_{3,1}$ — к дочернему окну от соответствующего пункта списка;
- $e_{1,0}$ — к разделу меню в случае вызова другого дочернего окна;
- $e_{3,0}$ — в случае ошибки пользователя при выборе пункта списка.

Графы, описывающие все возможные перемещения курсора между элементами интерфейса в главном меню, могут быть очень сложны. Поэтому ограничимся рассмотрением графа, представленного на рис. 3, так как обозначения его дуг и вершин могут быть распространены на все элементы главного меню и вызываемые с его помощью дочерние окна. Дуга $e_{3,1}$ соответствует перемещению курсора к дочернему окну, результат минимизации ее длины с помощью методов оптимизационного геометрического проектирования представлен на рис. 4.

Результат, представленный на рис. 4, был положен в основу разработанной модификации принципов работы с главным меню и вызываемыми с его помощью дочерними окнами. Для практической реализации предложенных улучшений потребовалось решить следующие задачи:

- минимизировать перемещения курсора (сложность выполняемых пользователем действий) — длины дуг, аналогичных дуге $e_{3,1}$ на рис. 3;
- исключить необходимость повторного воздействия на раздел меню при ошибке выбора пункта списка — дуги, аналогичные дуге $e_{3,0}$ на рис. 3;
- обеспечить последовательный вызов дочерних окон без выполнения воздействий на раздел меню — исключить дуги, аналогичные дуге $e_{1,0}$ на рис. 3.

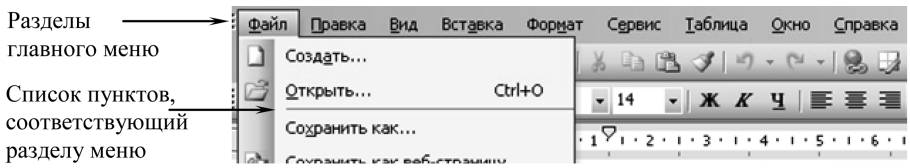


Рис. 1. Главное меню текстового редактора MS Word 2003

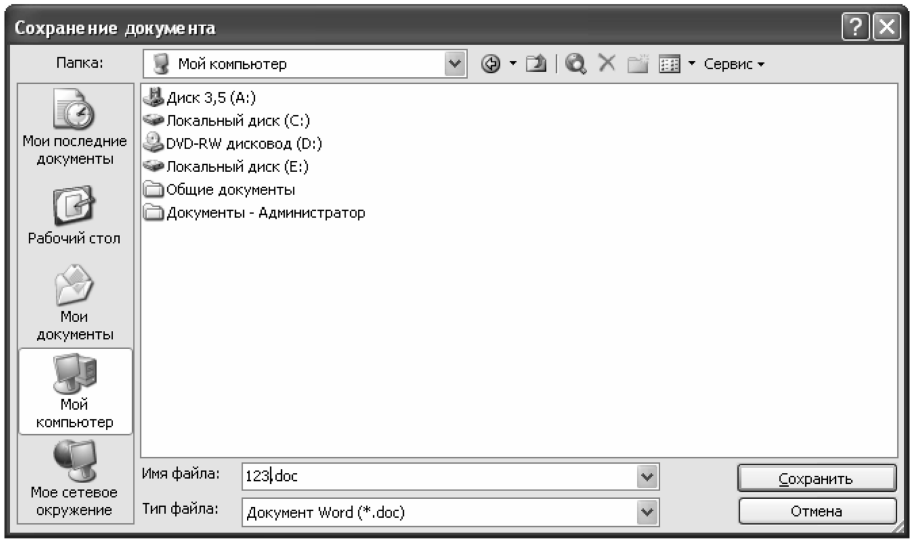


Рис. 2. Дочернее окно MS Word 2003, вызываемое из главного меню

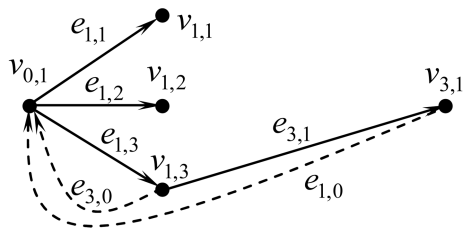


Рис. 3. Граф воздействий на элементы интерфейса

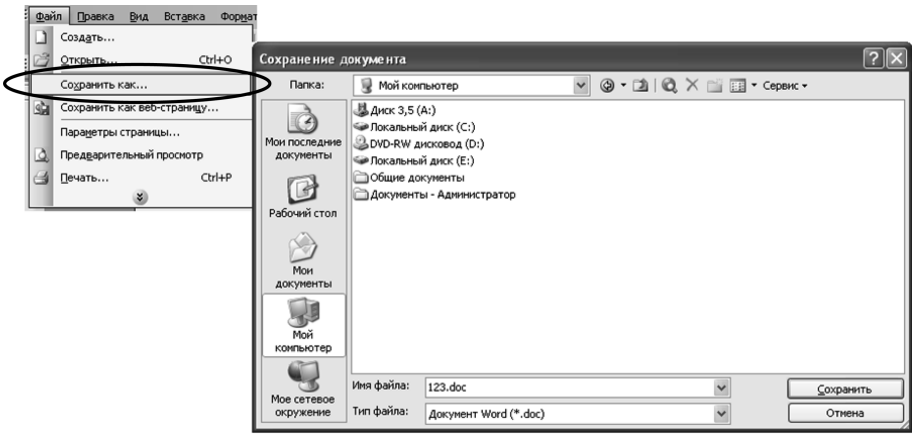


Рис. 4. Результат минимизации расстояния перемещения курсора

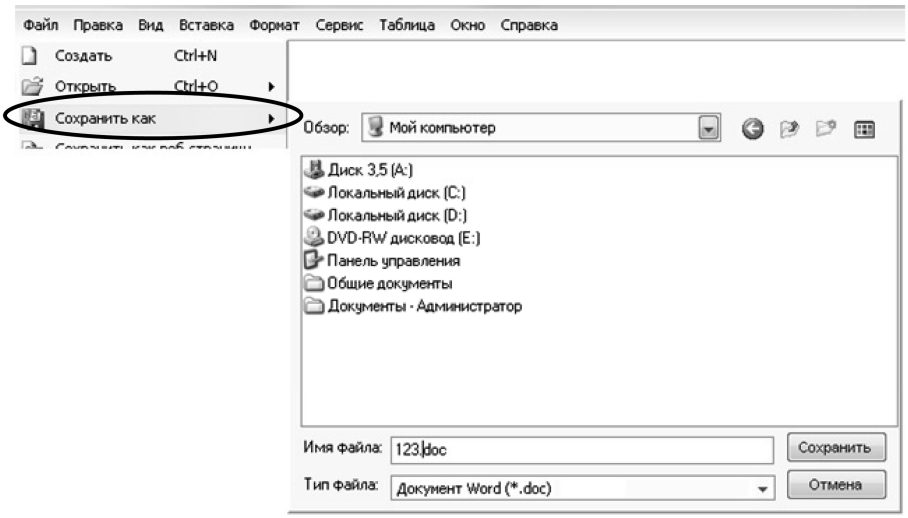


Рис. 5. Модифицированный принцип работы с главным меню и дочерними окнами

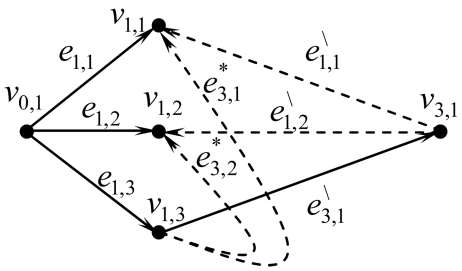


Рис. 6. Модифицированный граф воздействий на элементы интерфейса

Для решения перечисленных задач потребовалось обеспечить выполнение следующих условий:

- дочерние окна должны располагаться в непосредственной близости от соответствующих им пунктов списка (рис. 4);
- список не должен закрываться при воздействии на какой-либо из его пунктов, отвечающих за вызов дочернего окна.

В результате решения поставленных задач было разработано модифицированное главное меню, в котором списки не закрываются при вызове дочерних окон, а сами окна реализованы с помощью компонента Panel (панель). Результат модификации исходного варианта, представленного на рис. 1 и 2, изображен на рис. 5.

На рис. 6 представлен граф, описывающий работу пользователя со списком главного меню и дочерним окном, изображенными на рис. 5. Данный граф является модификацией графа, представленного на рис. 3.

Результаты сравнения графов, изображенных на рис. 6 и рис. 3, таковы:

- длины дуг $e_{1,1}$, $e_{1,2}$, $e_{1,3}$ эквивалентны;
- дуга $e_{3,0}$ исключена и воздействия на пункты списка возможны без повторных воздействий на раздел меню — введены дуги $e^{*}_{3,1}$ и $e^{*}_{3,2}$, длины которых эквивалентны соответственно длинам дуг $e_{1,2}$ и $e_{1,1}$;
- дуга $e_{1,0}$ исключена и последовательный вызов дочерних окон возможен без выполнения воздействий на раздел меню — введены дуги $e^{\setminus}_{1,1}$ и $e^{\setminus}_{1,2}$, длины которых меньше, чем длина дуги $e_{1,0}$;
- длина дуги $e_{3,1}$ минимизирована, в результате получена дуга $e^{\setminus}_{3,1}$.

Как видно из представленных результатов сравнения графов, предложенная модификация позволяет сократить число и сложность требующихся от пользователя воздействий на элементы интерфейса. Суммарное расстояние перемещений курсора в экранных пикселях сокращено на величину S, значение которой может быть оценено с помощью следующей формулы, в которой под обозначениями дуг понимаются их длины:

$$S = a_{3,0}e_{3,0} + \\ + a^{\setminus}_{1,1}(e_{1,0} - e^{\setminus}_{1,1} + e_{1,1}) + \\ + a^{\setminus}_{1,2}(e_{1,0} - e^{\setminus}_{1,2} + e_{1,2}) + \\ + a^{\setminus}_{3,1}(e_{3,1} - e^{\setminus}_{3,1}), \qquad (1)$$

где $a_{3,0}$, $a^{\setminus}_{1,1}$, $a^{\setminus}_{1,2}$ и $a^{\setminus}_{3,1}$ — число повторений перемещений курсора, обозначенных соответственно дугами $e_{3,0}$, $e_{1,1}$, $e_{1,2}$ и $e_{3,1}$ в графе, соответствующем исходному варианту принципов работы с главным меню и дочерним окном (см. рис. 3).

Заметим, что при практической реализации предложенных улучшений сокращение суммарного расстояния перемещений курсора аддитивно не только по числу их повторений, но и по таким параметрам, как число разделов меню, обеспечивающих вызов дочерних окон, число пунктов в соответствующих им списках, а также численность вызываемых дочерних окон.

Вызов некоторых дочерних окон MS Word 2003

№	Раздел меню	Пункт списка	Название окна
1	Файл	Открыть	Открытие документа
2	Файл	Сохранить	Сохранение документа
3	Файл	Параметры страницы	Параметры страницы
4	Файл	Печать	Печать
5	Правка	Найти (Заменить, Перейти)	Найти и заменить
6	Вставка	Номера страниц	Номера страниц
7	Вставка	Дата и время	Дата и время
8	Формат	Шрифт	Шрифт
9	Формат	Абзац	Абзац
10	Формат	Список	Список
11	Формат	Границы и заливка	Границы и заливка
12	Формат	Табуляция	Табуляция

Для оценки затрат времени пользователя на работу с исходным и модифицированным вариантами главного меню могут быть применены методы, изложенные в [5, 6] и учитывающие расстояния перемещений курсора в экранных пикселях. Исходными данными для рассматриваемого примера могут быть длины дуг графов, изображенных на рис. 3 и рис. 6, а также значение S, определенное по формуле (1). Отметим, что результаты оценки зависят от состава действий пользователя, числа их повторений, а также от таких факторов, как размер и разрешающая способность экрана монитора.

Необходимо также отметить, что в предложенной модификации отсутствует нарушение фокусировки внимания пользователя при переходе от списка к дочернему окну, так как список не закрывается, а области концентрации внимания оператора расположены на минимальном расстоянии друг от друга. В результате предложенная модификация принципов работы с главным меню превосходит исходный вариант как по эргономичности, так и по минимальным затратам времени пользователя, требующимся на работу с ней.

Для апробации предложенных улучшений была разработана модель WordModel текстового редактора MS Word, позволяющая оценить преимущества модификационных принципов работы с главным меню и дочерними окнами. В текстовом редакторе MS Word 2003 предусмотрено более 20 дочерних окон, вызываемых с помощью главного меню. В модели WordModel модифицирована работа с каждым из них. В таблице приведены действия, выполняемые пользователем при доступе к некоторым дочерним окнам: указан раздел меню, пункт списка и вызываемое окно.

В разработанной модели для изучения реакции пользователей на предложенную модификацию принципов работы с главным меню и дочерними окнами предусмотрена опция позиционирования окон. Помимо размещения окон, аналогичного представленному на рис. 5, возможно их расположение непосредственно под разделами главного меню или выше соответствующего пункта списка на указанное число экранных пикселей (рис. 7).

Данная опция позволяет минимизировать суммарное расстояние перемещения курсора от пункта списка ко всем элементам интерфейса в дочернем окне. Вызов дочернего окна происходит не только при воздействии на пункт списка, но и при задержке курсора на нем более чем на определенное задаваемое в настройках модели время. Установлено, что оптимальной задержкой отображения дочерних окон является одна секунда.

В разработанной модели также модифицирован принцип работы с дочерними окнами, вызываемыми с помощью элементов (кнопок) на панелях инструментов. Окна отображаются непосредственно под панелью, предусмотрена опция их позиционирования относительно левого края самой панели или элементов, отвечающих за их вызов. Пример результатов представлен на рис. 8, очевидно, что они могут быть распространены также на случай вертикального расположения панелей инструментов.

В разработанной модели улучшены принципы работы с контекстным меню, вызываемым нажатием правой клавиши мыши на области отображения текста. Исходный и модифицированный варианты представлены на рис. 9.

В исходном варианте контекстное меню позволяет вызывать дочерние окна *Шрифт*, *Абзац* и *Список* с помощью соответствующих пунктов, при воздействии на которые само меню закрывается. Такой принцип вызова дочерних окон приводит к нарушению фокусировки внимания пользователя (аналогично рассмотренному ранее для исходного варианта главного меню), а также к необходимости повторного вызова меню в случае ошибки при выборе пункта или при последовательной работе

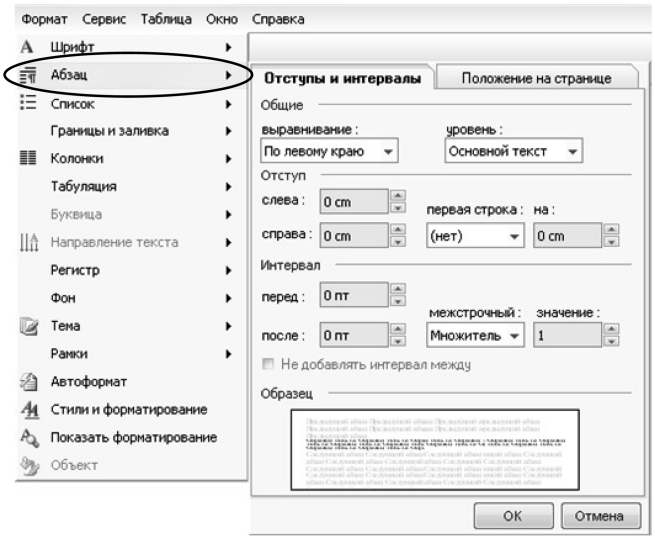


Рис. 7. Пример варианта позиционирования дочернего окна

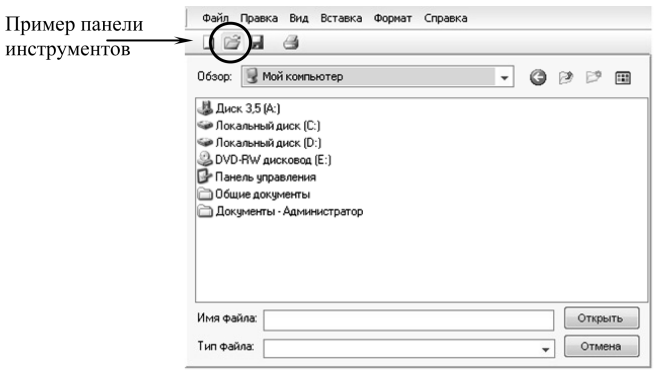


Рис. 8. Модификация принципов работы с панелями инструментов и дочерними окнами

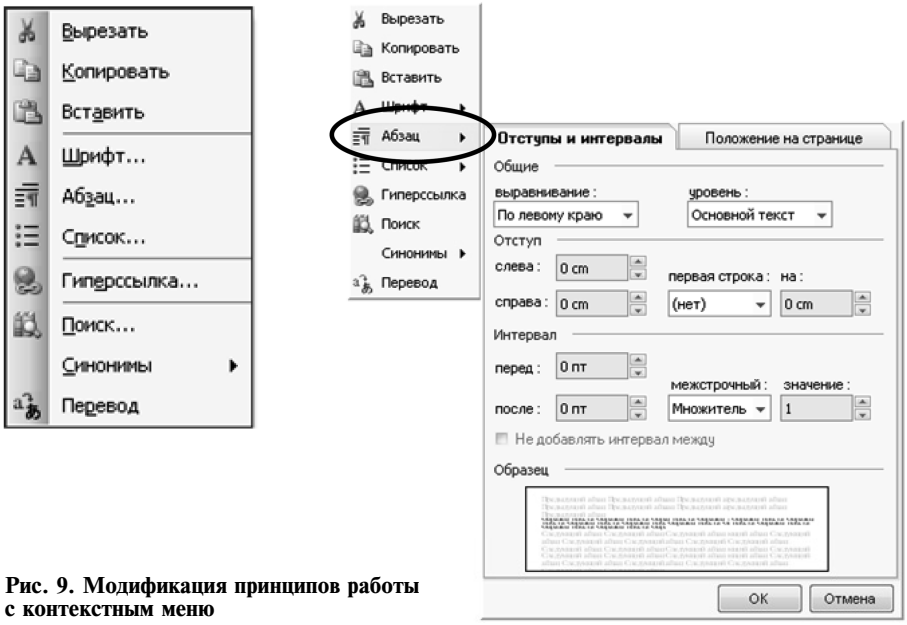


Рис. 9. Модификация принципов работы с контекстным меню

с несколькими дочерними окнами. В модифицированном варианте дочерние окна реализованы с помощью компонента Panel (панель), а принцип их отображения аналогичен рассмотренному ранее для главного меню. В результате контекстное меню не закрывается, доступна последовательная работа с дочерними окнами, а нарушение фокусировки внимания оператора отсутствует.

Сокращение суммарного расстояния перемещений курсора и затрат времени пользователя, наблюдаемое в результате реализации улучшений, предложенных для панелей инструментов и контекстного меню, может быть оценено аналогично представленному ранее для модифицированных принципов работы с главным меню.

Реализованная в модели WordModel модификация принципов работы с главным и контекстным меню, панелями инструментов и дочерними окнами удобна не только при использовании манипуляторов типа "мышь". Данная модификация адаптирована к работе с клавиатурой, так как переход фокуса (выделения выбранного элемента интерфейса) к дочернему окну от меню и панели инструментов выполняется с помощью клавиш с указателями стрелок.

В разработанной модели также реализован результат геометрической оптимизации размещения дочернего окна, содержащего запрос на сохранение изменений в документе при выходе из программы (рис. 10, см. третью сторону обложки).

Результат размещения окна, представленный на рис. 10, позволяет минимизировать расстояние перемещения курсора, а также исключить нарушение фокусировки внимания оператора, наблюдаемое в момент воздействия на элемент (1) и отображения окна в центре экрана монитора. Аналогичным является случай выхода из программы с помощью контекстного меню и панели задач операционной системы Windows (рис. 11, см. третью сторону обложки).

Представленный материал обосновывает превосходство принципов работы с интерфейсом разработанной авторами модели над принципами работы с главным и контекстным меню, панелями инструментов и дочерними окнами текстового редактора MS Word 2003. Полученные результаты обладают теоретической и практической ценностью, так как, во-первых, расширяют представление об особенностях работы пользователя с ГПИ и, во-вторых, могут быть реализованы в большом количестве программных продуктов.

Сравнение модели с интерфейсом редактора MS Word 2007

В 2007 г. корпорацией Microsoft была выпущена новая версия пакета программ MS Office, включающая текстовый редактор MS Word, интерфейс которого кардинально отличается от версии 2003 г. Основополагающим различием является группировка элементов интерфейса в верхней части окна программы под главным меню (рис. 12). Заметим, что данные различия вызвали существенные нарекания и недовольство пользователей.

В результате такого расположения элементов управления достигается сокращение расстояний перемещений курсора, требующихся от пользователя. Однако в рассматриваемом случае актуален также такой критерий оптимизации ГПИ, как размер области окна программы, доступной для представления пользователю информации. В текстовом редакторе MS Word 2007 при разрешающей способности экрана монитора 1024 на 768 пикселей главное меню занимает шестую часть окна программы, что ограничивает возможности пользователя по работе с отображаемым в нем текстом.

Модель WordModel обладает сопоставимыми расстояниями перемещения курсора, однако реализованное в ней главное меню, как и в версии MS Word 2003, занимает десятую часть окна программы. Вместе с тем, принцип работы с главным и контекстным меню, панелями инструментов и дочерними окнами модели WordModel основан на особенностях работы с интерфейсом

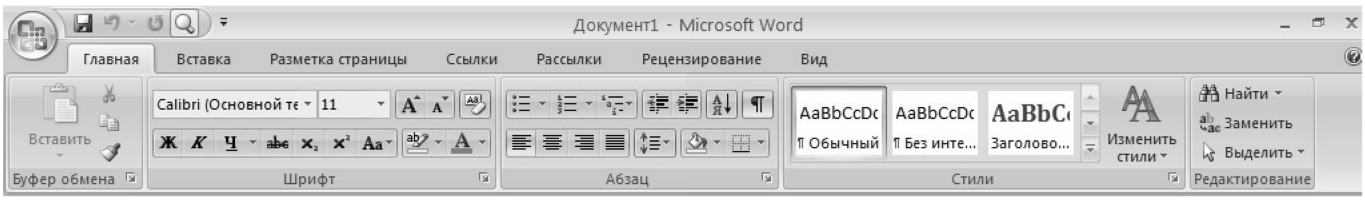


Рис. 12. Главное меню MS Word 2007

MS Word 2003. Это обеспечивает практически полную их совместимость в плане преемственности навыков пользователей, что значительно облегчило бы внедрение новой версии редактора MS Word (и других программ пакета MS Office) с предложенным авторами интерфейсом.

Заключение

Представленные в статье материалы обосновывают целесообразность и практическую полезность предложенной модификации принципов работы с главным и контекстным меню, панелями инструментов и дочерними окнами программ. Разработанная модель WordModel текстового редактора MS Word является конкурентоспособной оригинальной разработкой, соответствующей передовому уровню организации человеко-компьютерного взаимодействия. Данная модель является прототипом, позволяющим оценить реализованные в ней технические решения, в том числе — с точки зрения оценки целесообразности их реализации в прикладном программном обеспечении.

В работах [4—6] были представлены результаты модернизации адресной строки, проводника данных, окна операционной системы Windows и принципов работы с таблицами в текстовом редакторе MS Word. В совокупности с материалами настоящей статьи результаты, полученные в данном направлении исследований, позволяют утверждать о создании набора оригинальных технических решений, представляющих собой перспективное направление совершенствования графических пользовательских интерфейсов. В настоящее время авторы продолжают исследования, по имеющимся и получаемым результатам разрабатывают дополнительные элементы для библиотек визуальных компонентов объектно-ориентированных языков программирования и приглашают к сотрудничеству всех заинтересованных специалистов и представителей организаций.

Список литературы

1. **Зуев А. С.** Управление компьютерными программами посредством графических интерфейсов // Изв. РАН. ТиСУ. 2005. № 6. С. 127—142.
2. **Зуев А. С.** Некоторые вопросы исследования и проектирования интерфейсов компьютерных программ // Информационные технологии. 2006. № 10. С. 43—52.
3. **Зуев А. С.** Некоторые вопросы исследования и геометрического проектирования графических интерфейсов компьютерных программ // Изв. РАН. ТиСУ. 2008. № 1. С. 52—67.
4. **Зуев А. С.** Подход к разработке и модернизации структур интерфейсов компьютерных программ // Информационные технологии. 2007. № 1. С. 55—62.
5. **Зуев А. С., Петров Ю. И.** Описание модификации строки адреса проводника Windows Explorer // Информационные технологии. 2009. № 3. С. 11—19.
6. **Зуев А. С.** Математическое и программное обеспечение средств проектирования и совершенствования интерактивных графических человеко-машинных интерфейсов [Текст]: дис. ... канд. технич. наук / А. С. Зуев; Московский гос. ун-т приборостроения и информатики. Москва, 2006. 125 с.

СЕТИ И СИСТЕМЫ СВЯЗИ

УДК 004.3:621.391.82

В. А. Огнев, аспирант,
С. Р. Иванов, канд. техн. наук, доц.,
МГТУ им. Н. Э. Баумана,
e-mail: smarserg@mtu-net.ru

Методы повышения помехоустойчивости аппаратуры потребителей спутниковых навигационных систем

На основе анализа публикаций обобщаются методы улучшения работы аппаратуры потребителей систем ГЛОНАСС/GPS/Galileo в условиях действия помех, приводится сравнение эффективности различных методов, выявляются проблемы построения средств помехозащиты и намечаются направления дальнейшего исследования.

Ключевые слова: *спутниковая навигационная система, аппаратура потребителей, помехоустойчивость, навигационный сигнал, радиопомеха.*

В последние несколько лет происходит активное развитие технологии определения координат и текущего времени по сигналам спутниковых навигационных систем (СНС). В настоящее время развернуты американская СНС GPS и российская — ГЛОНАСС, Европейским Союзом ведется создание СНС Galileo [1—3].

Каждая СНС включает в себя три составляющие:

- космический сегмент (орбитальная группировка навигационных космических аппаратов (НКА), излучающих навигационные радиосигналы, и средства запуска);
- наземный сегмент (Центр управления системой и сети станций слежения и управления);
- сегмент потребителей (аппаратура, непосредственно предназначенная для выполнения навигационно-временных определений).

Согласно [4], наиболее активно развивается последняя составляющая, и именно на улучшение характеристик аппаратуры потребителей (АП) направлено данное исследование.

Активное применение аппаратуры спутниковой навигации в системах управления и диспетчеризации на транспорте (в авиации и на железных дорогах), для синхронизации телекоммуникационных сетей (в технологиях SONET/SDH, GSM, CDMA) и особенно в военной сфере (для топопривязки боевых машин и в качестве навигационного датчика на боеприпасах, беспилотных ле-

тательных аппаратах) делает многие процессы зависимыми от качества и надежности функционирования АП СНС [5]. Встает вопрос о необходимости обеспечения функционирования АП при воздействии помех различного происхождения.

Под *помехоустойчивостью* АП СНС понимают ее способность работать в условиях внешних радиопомех. Помехоустойчивость характеризуется предельно допустимым (наибольшим) значением отношения мощности сигнала помехи к мощности полезного сигнала $K_{\text{п}} = P_{\text{п}}/P_{\text{с}}$, при котором система еще может решать целевую задачу (выполнять навигационно-временные определения) с заданными характеристиками. Здесь $P_{\text{с}}$ — мощность полезного сигнала, $P_{\text{п}}$ — мощность помехи в полосе полезного сигнала. Параметр $K_{\text{п}}$ называется *коэффициентом подавления* и часто выражается в децибелах: $\tilde{K}_{\text{п}} = 10\ \lg(K_{\text{п}})$.

Сигналы СНС у поверхности Земли имеют малую мощность (на 30 дБ ниже уровня тепловых шумов), поэтому помехи даже небольшого уровня могут приводить к невозможности работы АП. Источниками непредумышленных помех [6, 7] могут являться гармоники сигналов, излучаемых различными (например, телевизионными) передатчиками. Первый постановщик помех военного применения для АП СНС [8] был разработан в 1999 г. Излучение этим устройством всего 8 Вт энергии в диапазонах L1 (1570 ... 1615 МГц) и L2 (1222 ... 1256 МГц) приводит к неработоспособности АП СНС в пределах прямой радиовидимости (в зоне радиусом до 150 км).

Исследователи, как правило [6, 9, 10], рассматривают помехоустойчивость АП СНС к следующим типам помех:

- гармоническим — описываются частотой ω и амплитудой A: $n_I(t) = A\cos(\omega,t)$;
- узкополосным — описываются типом модуляции и занимаемой полосой частот, могут быть представлены в виде $n_I(t) = \sum_{i=1}^{M} A_i(t)\mathbf{e}^{j(\omega_i(t)+\varphi_i)}$;
- широкополосным шумоподобным (гауссовым) помехам — их математической моделью является белый шум со спектральной плотностью N_I и шириной спектра Δf;
- импульсным — импульсы электромагнитной энергии, задаваемые амплитудой, частотой и скважностью;
- имитационным — особый тип помех, повторяющий по структуре сигналы НКА, их целью является не сделать невозможной работу АП СНС, а заставить ее использовать при выполнении навигационно-временных определений искаженные сигналы (излучаемые псевдоспутником — постановщиком помех, установленным на поверхности Земли).

В присутствии помех на вход АП СНС поступает смесь:

$$y(t) = \sum_{i=1}^{N} S_i(t) + n_I(t) + n_0(t),$$

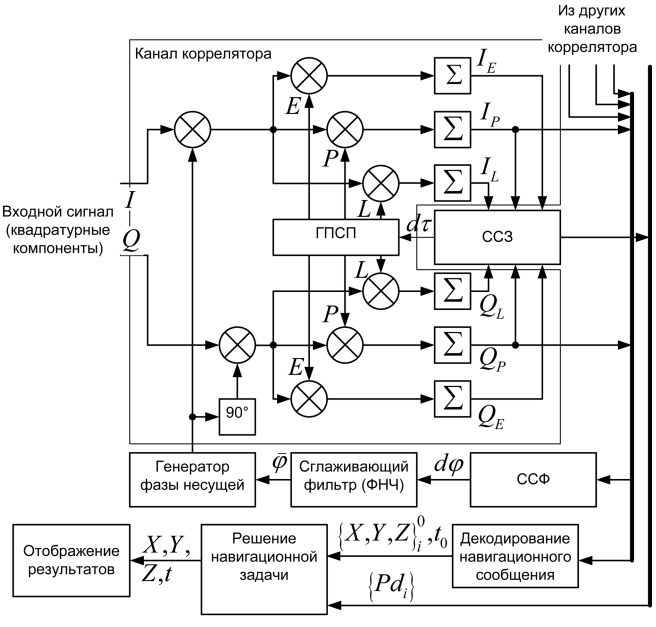
где $S_i(t)$ — сигнал от i-го НКА; $n_I(t)$ — помеха; $n_0(t)$ — тепловой белый Гауссов шум со спектральной плотностью N_0.

Аналитически сигнал S i-го НКА можно представить (см. [11]) в виде

$$S_i(t) = A_i G_{\text{ПСП}}(t-\tau_i) G_{\text{НС}}(t-\tau_i) \times \\ \times \cos((\omega_{0i} + 2\pi f_{\text{доп}\ i})t + \varphi_{0i}),$$

где A_i — амплитуда сигнала; $G_{\text{ПСП}}(t-\tau_i)$ — функция модуляции дальномерным кодом, в качестве которого высыпают псевдослучайные последовательности (ПСП), принимает значения −1 или 1; $G_{\text{НС}}(t-\tau_i)$ — функция модуляции навигационным сообщением, принимает значения −1 или 1; ω_{0i} — несущая частота; $f_{\text{доп}\ i}$ — доплеровский сдвиг частоты (возникающий вследствие взаимного перемещения НКА и АП); φ_{0i} — случайная начальная фаза сигнала.

Из теории оптимальной фильтрации [11] следует, что для оптимальной обработки в АП СНС должна вычисляться взаимокорреляционная функция принимаемого сигнала и его локально генерируемой копии. Однако для получения локальной копии необходимо оценивать (соответствующие вопросы на данный момент уже хорошо исследованы) задержки τ_i и фазы $(\omega_{0i} + 2\pi f_{\text{доп}\ i})t + \varphi_{0i} = d\varphi_i$ принимаемого сигнала. В [12] показано, что эти параметры можно описать в виде компонент марковского процесса третьего порядка. В существующих образцах АП СНС оценивание этих компонент для сигналов разных НКА ведется независимо. В результате АП представляет собой многоканальную следящую систему (рис. 1) с петлями слежения за задержкой τ_i (схема слежения за задержкой, ССЗ) и фазой $d\varphi_i$ (схема слежения за фазой, ССФ) сигнала в каждом канале. В [6] отмечается, что сужение полос пропускания данных петель слежения ведет к увеличению помехоустойчивости АП в ее основном режиме работы — режиме сопровождения сигналов НКА. Однако без принятия дополнительных мер это приводит к ухудшению точностных характеристик



ГПСП – генератор псевдослучайных последовательностей

Рис. 1. Схема обработки сигнала в АП СНС (на рисунке показан только один канал коррелятора)

АП СНС при ее размещении на высокодинамичных подвижных объектах (когда параметры принимаемого сигнала изменяются в широком диапазоне) вследствие затягивания переходных процессов, возникающих при маневрировании, в схемах слежения за параметрами сигналов.

Перед вхождением в режим слежения за сигналами НКА аппаратура потребителя должна пройти этап обнаружения сигналов навигационных космических аппаратов. Задачей этапа обнаружения (поиска) сигналов НКА является получение заключения о наличии в точке приема сигналов заданного набора НКА и формирование предварительной (грубой) оценки параметров τ_i и $f_{\text{доп } i}$ этих сигналов.

Таким образом, помехоустойчивость АП СНС необходимо обеспечивать в двух режимах работы: обнаружения и слежения за сигналами НКА.

В США проблеме повышения помехозащищенности в течение последних 10—15 лет уделяется значительное внимание. В результате фирмами Rocwell Collins, L3 и др. было создано несколько поколений помехозащищенной аппаратуры (см. [13]), в основном военного назначения. В то же время в России работы в этом направлении начались в последние 2—3 года — в рамках ФЦП "ГЛОНАСС" [14] проводятся ОКР "Актив-Н", "Квиток-М".

Сравнение параметров образцов зарубежной и отечественной [15—17] АП СНС (табл. 1, 2), с одной стороны, показывает значительное отставание в данном вопросе российских разработок, но с другой стороны, доказывает наличие потенциала для существенного улучшения характеристик отечественной АП.

В настоящее время единого стандарта, устанавливающего требования к помехозащищенности АП СНС для различных применений, не существует. В [18] приводится обоснование требуемого уровня помехозащищенности для авиационной АП СНС (работающей по сигналам GPS C/A и ГЛОНАСС ВТ), полученные результаты утверждены Международной радиотехнической комиссией RTCA в виде стандарта.

В последние годы в результате бурного развития СНС появились новые навигационные сигналы (табл. 3). Причем тенденцией является (см. [19]) использование двухкомпонентных меандровых сигналов с расщеплением спектра (Binary Offset Carrier, или BOC-сигналы — см. [19]). Широкая полоса частот, занимаемая этими сигналами и наличие в их структуре так называемых пилот-сигналов (компонент, не модулированных навигационным сообщением $G_{\text{НС}}(t - \tau_i)$) потенциально должны обеспечить повышенную помехоустойчивость АП СНС.

Однако помехоустойчивость АП при использовании этих перспективных сигналов пока исследована слабо, так как подавляющее большинство работ зарубежных исследователей посвящено анализу работы существующей АП, использующей только гражданский сигнал C/A GPS.

Поэтому целесообразно провести исследование поведения АП в целях получения обобщенных оценок помехоустойчивости приемника, работающего по перспективным сигналам СНС в присутствии помех разных типов. Эта задача связана с поиском границ марковского процесса [11].

Для достижения преследуемой цели (повышения помехозащищенности) необходимо построить упрощенную аналитическую модель АП и упрощенную модель помехи, получить аналитические зависимости и выпол-

Таблица 1

Параметры зарубежных приемоиндикаторов СНС

Параметр	Изделие			
	NavStorm	TruTrack	DIGAR	IGS
Производитель	Rockwell Collins	L3 IEC	Rockwell Collins	Honeywell
Страна	США	США	США	США
Сигналы каких СНС принимают	GPS	GPS	GPS	GPS
Обрабатываемые сигналы	C/A, P(Y)	C/A, P(Y)	C/A, P(Y)	C/A, P(Y)
Точность — (среднеквадратичное отклонение) измерения плановых координат, м	5	3	5	5
Время выдачи первого отчета при отсутствии априорной информации, с	10	6	—	10
Коэффициент подавления помехи, дБ	80—90	55	95	90
Максимальное число поставщиков помех	1	3	15	5

Примечание: C/A — сигнал стандартной точности системы GPS; P(Y) — криптозащищенный сигнал системы GPS для выполнения навигационных определений с повышенной точностью.

Таблица 2

Параметры существующих приемоиндикаторов СНС российских производителей

Параметр	Изделие			
	ПНС-2001	АТ-302	"Грот"	СН-3700
Производитель	"МКБ "Компас"	РИРВ	НИИ КП	"КБ "Навис"
Год разработки	2001	2003	1998	2003
Сигналы каких СНС принимают	GPS и ГЛОНАСС	GPS и ГЛОНАСС	GPS и ГЛОНАСС	GPS и ГЛОНАСС
Обрабатываемые сигналы	GPS C/A, ГЛОНАСС ПТ, ГЛОНАСС ВТ	GPS C/A, ГЛОНАСС ПТ	GPS C/A, ГЛОНАСС ПТ, ГЛОНАСС ВТ	GPS C/A, ГЛОНАСС ПТ
Точность — (среднеквадратичное отклонение) измерения плановых координат, м	15	25	30	25
Время выдачи первого отчета при отсутствии априорной информации, с	180	180	120	180

Таблица 3

Новые сигналы СНС ГЛОНАСС, GPS и GALILEO

Тип системы	Обозначение сигнала	Несущая частота, МГц	Тип сигнала (кода)	Длина кода, бит	Тип кода при передаче данных
ГЛОНАСС	L3	1198...1207			
GPS	L1C	1575,42	MBOC (6, 1,1/11)	10230	Циклический
	L2C (состоит из двух компонент — L2CM и L2CL)	1227,6	BPSK	L2CM→ 10230 L2CL→ 767250	Сверточный
	L5	1176,6	BPSK	10230	Сверточный
GALILEO	E1	1575,42	MBOC (6, 1,1/11)	4092	Сверточный
	E5a, E5b	1176,45 1207,14	BOC (15, 10) BOC (15, 10)	10230	Сверточный

Примечание: BPSK (Binary Phase Shift Keying) — двоичная фазовая манипуляция; BOC (Binary Offset Carrier) — меандровый шумоподобный сигнал с расщеплением спектра; MBOC — (multiplexed BOC) — сложный сигнал, состоящий из двух компонент типа BOC.

нить имитационное моделирование. Отметим, что обобщенной методики такого исследования в литературе обнаружено не было, хотя в [11] рассматривается метод оценки помехоустойчивости АП к воздействию широкополосных помех, а в [6] приведена методика оценки помехозащищенности от узкополосных помех при работе по сигналам С/А GPS.

Исследователи [6, 10] отмечают, что различные типы помех имеют различный механизм воздействия на АП СНС. Широкополосные помехи повышают уровень шума во всей полосе приема, что ведет к снижению отношения сигнал/шум. Узкополосные помехи опасны, если, находясь вблизи несущей частоты, проходят через схемы слежения (которые можно рассматривать как узкополосные фильтры). В таком случае, даже обладая небольшой мощностью, они приводят к неработоспособности АП СНС. Импульсные помехи могут перегружать входные каскады по мощности. Различные механизмы действия помех приводят к необходимости использования разных средств для борьбы с разными типами помех.

Обзор методов, применяемых для помехозащиты АП СНС, приводится в [9, 20]. Классификация этих методов представлена на рис. 2.

Целью методов пространственно-временной обработки является минимизация влияния помех путем управления диаграммой направленности (ДН) многоэлементной антенной решетки. Система управления такими антенными решетками, по сути, представляет собой адаптивный фильтр, настраивающийся по параметрам помехи. Использование данного подхода эффективно для борьбы с широкополосными помехами.

В методе управления поляризацией антенны, разработанном и запатентованном специалистами фирмы Electro-Radiation [21, 22], учитывается, что сигналы СНС имеют правую круговую поляризацию и в антенне осуществляется управление ортогональными компонентами этого сигнала. Детали алгоритма управления в литературе не раскрываются.

При формировании нулей [23, 24] ДН в направлении на помеху центральной задачей является формирование матрицы весов, описывающей, какой вклад в результирующий сигнал дает выход каждого элемента антенны. Данная матрица должна формироваться динамически с учетом текущих параметров помехового воздействия. Согласно [6], для этого используются: метод наименьших квадратов, алгоритм Аплебаума или алгоритм "инверсии" энергии. Последний наиболее широко применяется на практике и подробно рассмотрен в [11]. В частности, показывается, что вычисление коэффициентов синтезируемого фильтра связано с операциями формирования корреляционной матрицы входного процесса (шум + + помеха + сигналы НКА). Однако для формирования данной матрицы необходимо решение системы линейных алгебраических уравнений высокого порядка, что связано с большими вычислительными затратами, поэтому имеется потребность в разработке альтернативных методов вычисления матрицы весов.

В отличие от методов пространственно-временной обработки, методы фильтрации в частотной и временной областях позволяют бороться с гармоническими и узкополосными помехами.

Для фильтрации сигнала во временной области применяются следящие и неследящие алгоритмы компенсационного типа.

Следящие алгоритмы компенсационного типа рассмотрены в [25]. К их недостаткам следует отнести то, что на каждую входную помеху требуется своя следящая система, которая должна проходить этап захвата слежения, в процессе чего может быть подвержена срывам слежения.

Рис. 2. Подходы к улучшению работы АП СНС в условиях действия помех

Неследящие алгоритмы компенсационного типа приводят к появлению линейного фильтра с конечной импульсной характеристикой (трансверсального фильтра). Поскольку априорные сведения о параметрах помехи неизвестны, встает проблема разработки алгоритмов адаптации коэффициентов фильтра. Применение итеративных методов адаптации коэффициентов [26] приводит к длительному времени адаптации. Использование прямых методов адаптации позволяет добиться хорошей степени подавления помех при не слишком высоком порядке трансверсального фильтра. Задача эффективного вычисления коэффициентов такого фильтра решена в [27].

Использование алгоритмов фильтрации в частотной области [28—31] подразумевает анализ спектра входного процесса с последующим обнаружением и "вырезанием" помех. Недостатком данного метода являются высокие аппаратные затраты при его использовании в чистом виде. Таким образом, необходимо провести поиск путей снижения ресурсов на его реализацию. Кроме того, необходимо найти способ снижения влияния эффектов Гиббса, возникающих при возврате во временную область.

При невозможности использования специальных средств помехозащиты (например, из-за массогабаритных ограничений или по соображениям минимизации стоимости) повышение помехоустойчивости АП СНС достигается в основном за счет мер, позволяющих снизить ширину полос пропускания схем слежения за параметрами сигналов НКА.

При интеграции АП СНС с инерциальными навигационными системами (ИНС) сужение полос сопровождения достигается за счет использования оценок изменений задержек и частот сигналов НКА, вычисленных на основе данных, получаемых от ИНС. Стандартный способ осуществления такой интеграции базируется на построении многомерного фильтра Калмана [32], для оптимальной работы которого необходимы модель погрешностей входных измерений и модель динамики системы. Поэтому основным вопросом является получение математической модели, описывающей погрешности ИНС. Модель динамики системы зачастую известна не полностью, что может приводить к накоплению ошибок при использовании фильтра Калмана. Для устранения этого в [33] предполагается использовать нейронную сеть, которая способна адаптироваться в условиях отсутствия точной модели динамики системы. Основной проблемой при этом является разработка методики обучения такой нейронной сети.

Векторное управление каналами коррелятора предполагает наличие одного векторного дискриминатора задержки и фазы сигнала для всех каналов коррелятора, а не для каждого (как на рис. 2). В результате вычисление приращений задержки и фазы происходит с использованием энергии всех сигналов НКА. Сложность заключается в получении решений уравнений Рикатти, описывающих такую многомерную следящую систему.

Использование адаптивных квантователей и адаптивной регулировки усиления подробно рассмотрено в [6] и дальнейшего исследования не требует.

Методы борьбы с имитационными помехами на сегодняшний день хорошо изучены и обобщены в [34]. Основная задача — проведение аутентификации НКА.

В табл. 4 обобщены результаты анализа эффективности различных методов повышения помехоустойчивости АП СНС при действии помех разных типов.

Таблица 4

Выигрыш в помехозащищенности АП СНС при использовании различных методов для борьбы с разными типами помех

Метод помехозащиты	Тип помехи		
	Гармонический	Узкополосный	Шумоподобный широкополосный
Переключение антенн	20...25 дБ	20...25 дБ	20...25 дБ
Управление поляризацией антенны	25...30 дБ	25...30 дБ	20...25 дБ
Формирование нулей ДН в направлении на помеху	—	—	35...40 дБ
Формирование лепестков ДН в направлении на НКА	—	—	40...45 дБ
Фильтрация в частотной области	До 60 дБ	25...35 дБ	—
Фильтрация во временной области	До 60 дБ	30...40 дБ	—
Интеграция с другими навигационными датчиками	—	—	До 12 дБ
Векторное управление каналами коррелятора	До 7 дБ	До 7 дБ	До 7 дБ
Адаптивные квантователи и адаптивное АРУ	До 3 дБ	До 3 дБ	До 3 дБ

Неисследованным является вопрос обеспечения быстрого поиска широкобазовых сигналов (например, ГЛОНАСС ВТ), доступных специальным потребителям и призванных обеспечить высокую помехоустойчивость за счет более широкого спектра (например, спектр сигнала ГЛОНАСС ВТ шире спектра сигнала ГЛОНАСС ПТ в 10 раз).

В существующей АП СНС переход на работу по коду ГЛОНАСС ВТ осуществляется после обнаружения сигналов ГЛОНАСС ПТ, потенциально обеспечивающих более низкую помехоустойчивость. Таким образом, стоит проблема разработки алгоритмов для прямого выхода АП СНС на работу по коду ГЛОНАСС ВТ или, в более общем случае, проблема быстрого обнаружения широкобазовых сигналов.

По вопросам ускорения поиска сигналов СНС существует большое число публикаций, которые можно разбить на несколько групп:

- оптимизация процедуры накопления сигнала и принятия решения;
- построение обнаружителя в форме согласованного фильтра;
- использование преобразования Фурье для поиска по частоте;
- поиск по задержке с использованием дискретной свертки, вычисляемой в спектральной области.

Первая группа публикаций сводится к уменьшению требуемого времени накопления сигнала, а остальные — к оптимизации процедуры вычисления корреляционных интегралов.

Первый подход — оптимизация процедуры обнаружения — заключается, как правило, в применении того или иного многоэтапного алгоритма принятия решения.

Так, например, в [35] рассматривается трехэтапная процедура обнаружения, позволяющая получать грубую оценку параметров на первом шаге с последующим их уточнением на других шагах. Другой подход известен под названием "последовательный наблюдатель" [36]. Идея таких алгоритмов заключается в разбиении процесса накопления на несколько этапов с тем, чтобы была возможность отсекать неправильные варианты до того, как будет осуществлено полное вычисление корреляционного интеграла. Применение подобного подхода к сигналам СНС можно найти, например, в [37]. Данные методы позволяют несколько снизить среднюю длительность процесса поиска по сравнению с традиционной структурой, но незначительно — до 15 %, что не достаточно.

Построение обнаружителя в форме согласованного фильтра является вариантом распараллеливания процесса вычислений. Данный метод подробно изучен. Удобством трансверсального фильтра является то, что последовательно идущие отсчеты на его выходе представляют собой значения корреляционных интегралов, вычисленные при соответствующих последовательных значениях задержки. Примеры построения блока быстрого поиска на основе согласованного фильтра можно найти в [38—40].

Третий подход — наиболее очевидный, заключается в использовании БПФ для поиска сигнала по частоте [28, 41]. Обычно число частотных каналов не очень велико. Так, например, в [41] рассматривается поиск в 64-частотных каналах. Выигрыш от использования подобных алгоритмов заключается лишь в эффективности алгоритма БПФ по сравнению с обычным алгоритмом дискретного преобразования Фурье (ДПФ). Известно, что выигрыш от использования БПФ увеличивается с увеличением объема анализируемой выборки. Так, при 64-канальной обработке выигрыш составляет около 10 раз. Алгоритмы, базирующиеся на дискретной свертке в спектральной области, используют БПФ объемом несколько тысяч точек для поиска по задержке, поэтому выигрыш от их использования выше [41, 46].

В литературе отсутствует информации по вопросам:

- стыковки средств борьбы с разными типами помех;
- влияния помех на точность навигационных определений.

Кроме того, применительно к АП СНС GPS появились публикации [44, 45] о повышении помехозащищенности за счет использования свойств псевдослучайных последовательностей (разностный алгоритм обнаружения), входящих в ее структуру. Целесообразно попытаться обобщить этот подход и для сигналов с частотным разделением, используемым в СНС ГЛОНАСС.

Задачами дальнейшего исследования могут являться:

- получение обобщенной методики оценки с выполнением численного моделирования помехоустойчивости АП СНС при работе по сигналам с различной структурой (в том числе — перспективных) для многосистемного приемоиндикатора и выявление факторов, влияющих на нее;
- синтез алгоритма подавления узкополосных помех с использованием спектральных методов, разработка математической модели и проведение численного моделирования такого подавителя помех;
- синтез алгоритма подавления широкополосных помех, разработка математической модели и проведение численного моделирования такого подавителя помех;
- разработка алгоритма быстрого поиска широкобазовых сигналов (например, сигналов ГЛОНАСС ВТ, минуя стадию обнаружения сигналов ГЛОНАСС ПТ), разработка математической модели и поведение численного моделирования блока быстрого поиска.

Решение поставленных задач позволит повысить надежность работы аппаратуры потребителей (в том числе, использующей перспективные сигналы СНС) в условиях действия помех, что особенно важно в таких приложениях, как навигация транспортных средств, наведение боеприпасов и др.

Список литературы

1. **Ballenger W. A.** GPS Status update // Матер. конф. ION CNSS 2005, сентябрь 2005, г. Лонг-Бич, США.
2. **Revnivykh S.** GLONASS Status, Perfomance and Perspectives // Матер. конф. ION GNSS 2005, сентябрь 2005, г. Лонг-Бич, США.
3. **Ruiz I.** GALILEO Overall Programme Status // Матер. конф. ION GNSS 2005, сентябрь 2005, г. Лонг-Бич, США.
4. **Лисов И.** Внедрение "ГЛОНАСС": когда и как? // Новости космонавтики, 2007.
5. **James C.** Vulnerability Assessment of the U. S. Transportation Infrastructure that Relies on GPS // Матер. конф. ION NTM 2001, январь 2001, г. Лонг-Бич, США.
6. **Spilker J., Natali F.** Global Positioning System: Theory and Applications. 1996. Vol. 1. Chapter 20. Interference Effects and Mitigation Techniques. American Institute of Aeronautics and Astronautics, 68 с.
7. **Deshpande S.** Modulated Signal Interference in GPS Acquisition // Матер. конф. ION GNSS 2004, сентябрь 2004, г. Лонг-Бич, США, 11 с.
8. **Gershanoff H.** Russian GPS Jammer Introduced // Journal of Electronic Defense, август 1999.
9. **Casabona M., Rosen M.** Discussion of GPS Anti-jam Technology. Electro-Radiation Inc. 1999.
10. **Cannon M., Deshpande S.** Interference Effects on the GPS Signal Acquisition // Матер. конф. ION NTM 2004, 26—28 января 2004.
11. **Перов А. И., Харисов В. Н.** ГЛОНАСС. Принципы построения и функционирования. Изд. 3-е, перераб. М.: Радиотехника, 2005. 688 с.
12. **Болденков Е. Н.** Разработка и исследование оптимальных алгоритмов обработки сигналов в аппаратуре спутниковой навигации. Автореферат дисс. на соискание степени канд. техн. наук. Москва, 2007. 21 с.
13. **Rowe D., Weger J.** Integrated GPS Anti-Jam Systems // Матер. конф. ION GNSS 2005, сентябрь 2005, г. Лонг-Бич, США. 7 с.
14. **http:**//www.krd.ru/www/prom.nsf/webdocs/40949F087838980BC325706E002F46CF.html Текст Федеральной Целевой Программы "Глобальная навигационная система".
15. **http:**//www.mkbkompas.ru/.files/mkb_kompas_pr.ppt. Каталог продукции ОАО МКБ "Компас".
16. **http:**//www.navis.ru/wmc/ru/catalog/potreb/avia/?id = 1096961568 Описание прибора СН-3700 КБ "Навис".
17. **www.**rirt.ru/product/at-302.htm Описание прибора АТ-302 Российского института радио и времени.
18. **Hegarty C.** Analytical Derivation of Maximum Tolerable In-Band Interference Levels for Aviation Applications of GNSS // Journal of The Institute of Navigation. 1997. Vol. 44. 21 p.
19. **Ярлыков М. С.** Меандровые радиосигналы (BOC-сигналы) в спутниковых радионавигационных системах нового поколения // Новости навигации. 2007. № 3.
20. **Antijamming** and GPS for Critical Military Applications // Grosslink. The Aerospace Corporation Magazine of advances in aerospace technology. Июль 2002. 9 p.
21. **Falcone K., Dimos G.** Small affordable anti-jam GPS antenna (SAACA) development // Матер. конф. ION GPS-99, г. Нэшвилл, США, сентябрь 1999. 7 p.
22. **Rosen M., Braasch M.** Low-Cost GPS Interference Mitigation Using Single Aperture Cancellation Techniques // Матер. конф. ION NTM-98. США. 1994. 14 p.

23. **Kim S., Iltis R.** GPS C/A Code Tracking with Adaptive Beamforming and Jammer Nulling // Матер. 36-й конф. IEEE. Секция "Signals, Systems and Computers". 2002. Vol. 2. P. 975—979.

24. **Robust** Adaptive Beamforming Using Worst-Case Perfomance Optimization: A Solution to the Signal Mismatch Problem // IEEE Transaction on Signal Processing. 2002. Vol. 51. N 2. 11 p.

25. **Перов А. И.** Синтез оптимального алгоритма обработки сигналов в приемнике спутниковой навигации при воздействии гармонической помехи // Радиотехника. 2005. № 7. 6 с.

26. **Уидроу Б., Стирнс С.** Адаптивная обработка сигналов. М.: Радио и связь, 1989.

27. **Болденков Е. Н.** Разработка и исследование оптимальных алгоритмов обработки сигналов в аппаратуре спутниковой навигации. Дисс. ... канд. техн. наук. М.: МЭИ. 2007. 226 с.

28. **Огнев В. А., Санников М. Г., Сурков Д. М.** Применение спектральных методов для подавления узкополосных помех в авиационных приемоиндикаторах СРНС // Матер. конф. "Гражданская авиация на современном этапе развития науки, техники и общества". МГТУ ГА, 18—19 мая 2006 г.

29. **Gunawardena S., Soloviev A.** Real Time Block Processing Engine for Software GNSS Receiver // Доклад на конф. Международного института навигации, 26—28 января 2004 г., Сан-Диего, США.

30. **Шилов А. И., Бакитько Р. В.** Предварительная обработка шумоподобных сигналов при наличии сильных интерференционных помех // Радиотехника. 2005. № 7.

31. **Бакитько Р.** Использование весовых функций для предварительной обработки шумоподобных сигналов при наличии сильных интерференционных помех // Радиотехника. 2006. № 6. С. 4.

32. **Petovello M., Lashapelle C.** Ultra-Tight GPS/INS for Carrier Phase Positioning In Weak-Signal Environment // Матер. конф. NATO RTO Set-104. Symposium on Military Capabilities "Enabled by Advances in Navigation Sensors". Анталия, Турция. 2007. 18 p.

33. **Wang G., Sinclair D.** A Neutral Network and Kalman Filter Hybrid Approach for GPS/INS Integration. 6 p. http://www.gmat.unsw.edu.au/snapl publications/wangja etal 2006 c. pdf.

34. **Wen H., Yih-Ru P.** Countermeasures for GPS signal spoofing, 2005. 5 p.

35. **Tsui J., Stockmaster M.** Block adjustment of synchronized signal (BASS) for global positioning system (GPS) receiver signal processing // Матер. конф. ION GPS-97. Kansas city, Missoury. 1997. 6 с.

36. **Тихонов В. И., Харисов В. Н.** Статистический анализ и синтез радиотехнических устройств и систем. 2-е изд. М.: Радио и связь, 2004. 608 с.

37. **Eerola V.** Rapid parallel GPS signal acquisition // Матер. конф. ION GPS-2000. Salt Lake City, 2000. 6 p.

38. **Liusin S., Khazarov I.** Fast acquisition by matched filter technique for GPS/GLONASS receivers // Матер. конф. ION GPS-98. Нэшвилл, США, 8 p.

39. **Wang M., Chen S.** Joint code acquisition and frequency offset estimation for the GPS L5 receiver // Матер. конф. ION GNSS-04. г. Логн-Бич, 2004, США. 6 p.

40. **Akopian D., Agaian S.** Fast and parallel matched filters in time domain // Матер. конф. ION GNSS-04. г. Логн-Бич, США, 2004. 9 p.

41. **Fu X., Arai T.** Error probabilities for determining carrier presence or absence by FFT algorithm // Матер. конф. ION GPS-97. Kansas city, США, 1997. 8 p.

42. **Rounds S.** A low cost, unclassified, direct Y-code fast acquisition SAASM // Матер. конф. ION GPS-98. Нэшвилл, США, 1998. 5 p.

43. **Wolfert R., Chen S.** Rapid direct P(Y) acquisition in a hostile environment // Матер. конф. ION GPS-98. Нэшвилл, США, 1998. 7 p.

44. **Shanmugam S.** Narrowband Interference Suppression Performance of Multi-Correlation Differential Detection // ENC-GNSS 2007. Женева, Швейцария, 29—31 мая 2007. 12 p.

45. **Shanmugam S., Lachapelle G.** Pre-Correlation Noise and Interference Suppression for Use in Direct-Sequence Spread Spectrum Systems with Periodic. PRN Codes // Матер. конф. ION GNSS 2006, г. Форт-Ворс, США, 2006. 11 p.

46. **Огнев В. А., Санников М. Г., Сурков Д. М.** Применение спектральных методов для быстрого поиска сигналов СРНС // Матер. конф. "Гражданская авиация на современном этапе развития науки, техники и общества", МГТУ ГА, 18—19 мая 2006 г.

УДК 621.395; 519.872

А. К. Гечис, студент,
Новосибирский государственный университет,
О. Д. Соколова, канд. техн. наук, науч. сотр.,
Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск,
Н. А. Соколов, д-р техн. наук, проф.,
Государственный университет телекоммуникаций, г. Санкт-Петербург

Входящий поток заявок для голосового трафика в сетях следующего поколения

Рассматривается задача определения статистических характеристик потока IP-пакетов в сетях связи следующего поколения. Задача решена за счет построения имитационной модели, адекватной реальному процессу обмена IP-пакетами. С помощью моделирования показано, что для реальных значений интенсивности потока вызовов в таких сетях порождается пуассоновский поток IP-пакетов.

Ключевые слова: *сети следующего поколения, голосовой трафик, статистические характеристики IP-пакетов.*

Введение

Обмен информацией любого вида (речь, данные, видео) в сетях следующего поколения (*NGN*) осуществляется в форме *IP*-пакетов [1]. Известно, что голосовой трафик очень чувствителен к задержкам передачи информации, и поэтому анализ вероятностно-временных характеристик такого трафика в *NGN* является актуальной задачей. Есть все основания полагать, что характер поступления *IP*-пакетов на коммутирующий узел имеет Пуассоновское распределение. Такое предположение

вполне оправдано, так как характер потока вызовов в сети с коммутацией каналов является пуассоновским, что подтверждено большим числом измерений [2, 3]. Цель данной работы — определение функции распределения потока *IP*-пакетов в сети *NGN*.

Постановка задачи

Будем использовать следующие известные понятия:

- *вызов в сети с коммутацией каналов* — заявка на соединение абонентов;
- *вызов в сети с коммутацией пакетов* — первый *IP*-пакет, поступающий пользователю после соединения абонентов;
- *заявка в сети с коммутацией пакетов* — любой *IP*-пакет в сети.

Рассмотрим следующую ситуацию. В сети телефонной связи в моменты времени T_j поступают вызовы. Так как поток вызовов подчиняется распределению Пуассона, то это означает, что существует функция $A(t)$ распределения длительности интервалов между вызовами, и она экспоненциальна с интенсивностью λ. В силу принятых выше понятий, с формальной точки зрения поток вызовов в сети с коммутацией пакетов и поток заявок в сети с коммутацией каналов эквивалентны. Как следствие, функции распределения временных интервалов между вызовами одинаковы. Математическое ожидание длительности интервала между вызовами равно λ^{-1}. Величины λ и n (число абонентов) связывает следующая формула: $3600\lambda = ni$, где i — число вызовов, совершаемых абонентом в час. Например, если один абонент делает 5 вызовов в час наибольшей нагрузки (это среднестатистические данные для телефонной сети общего пользования), то это означает, что 200 абонентов сделают 1000 вызовов, и, следовательно, $\lambda = 0{,}2778$ с$^{-1}$.

В сети *NGN* каждый успешный вызов порождает поток *IP*-пакетов (заявок), число пакетов прямо пропорционально времени разговора. Пакеты передаются только в то время, когда абонент говорит. Обычно время активности оценивается коэффициентом α, который считается равным коэффициенту β — доле времени, когда говорит другой абонент. Коэффициент $\gamma = 1 - (\alpha + \beta)$ определяет период времени, когда молчат оба абонента. Типичное распределение, таким образом, характеризуется параметрами: $\alpha = 0{,}45$, $\beta = 0{,}45$, $\gamma = 0{,}1$. Ситуации, когда одновременно говорят оба абонента, встречаются редко, и такие события в модели не учитываются.

Рассматриваемая авторами задача состоит в том, чтобы определить статистические характеристики потока *IP*-пакетов, *учитывая произвольный характер активности абонентов* (рис. 1). Расчеты проводятся для часа наибольшей нагрузки.

Рассмотрим модель, состоящую из одного коммутирующего узла и группы абонентов, которые соединяются друг с другом только через него. Поток вызовов в таком случае представляет собой суммарный поток, генерируемый всеми абонентами. В телефонной сети потери составляют 1—2 %, а для одного узла — около 0,5 %, и при моделировании этими потерями можно пренебречь. Частота передачи пакетов в модели предполагается равной 0,02 с (среднее значение, в большинстве случаев используемое в *IP*-телефонии) [2, 3]. Длительность самого периода активности определяется временем произнесения нескольких фраз (считается, что это время не может быть меньше 3 с).

Важное свойство некоторых классов потоков вызовов — отсутствие последействия. Если поток вызовов рассматривается после какого-то момента времени t_u, и его характеристики не зависят от поведения потока для $t < t_u$, то принято считать, что последействия отсутствуют.

Еще одним важным атрибутом потока вызовов следует считать стационарность. Рассмотрим конечную совокупность непересекающихся интервалов времени. Если вероятность π_k поступления k вызовов не меняется при сдвиге этой совокупности интервалов на любой отрезок времени, то поток считается стационарным. Для стационарного потока вероятность $\pi_k(\alpha, \beta)$ для интервала времени $[\alpha, \beta]$ зависит не от значений величин α и β, а только от их разности.

Детерминированный поток вызовов может быть представлен последовательностью моментов времени t_n ($n \geqslant 1$), в которые поступают вызовы. В простейшем случае в любой момент времени может поступить не более одного вызова (такой поток называется ординарным). Для потока вызовов можно определить вероятность поступления хотя бы k вызовов — $\phi_k(\alpha, \beta)$. Параметром потока $\lambda(t)$ называется следующий предел (если, конечно, он существует):

$$\lambda(t) = \lim_{\tau \to 0} \frac{\phi_1(t, t+\tau)}{\tau}.$$

В телефонии определяют *простейший* поток. Такой поток характеризуется тремя важными свойствами: он стационарен, ординарен и не имеет последействия. Это означает, что $\lambda \neq f(t)$ (параметром потока не является функция от времени) [2, 3]. Распределение длин проме-

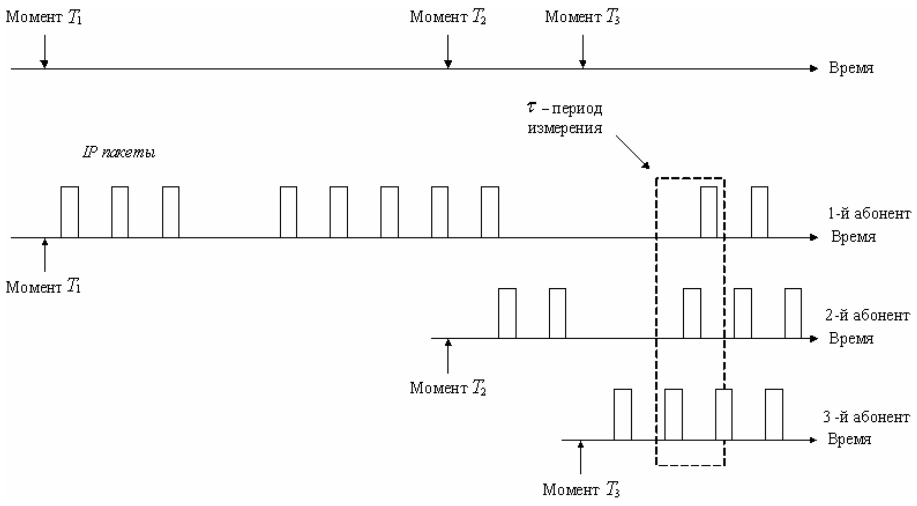


Рис. 1. Временная диаграмма сеанса связи для трех абонентов в *IP*-сети

жутков между вызовами в сети с коммутацией каналов для простейшего потока подчиняется экспоненциальному закону: $A(t) = 1 - \mathbf{e}^{-\lambda t}$. Вероятность поступления k вызовов за период длительностью t определяется распределением Пуассона: $\pi_k(a, a + t) = \frac{(\lambda t)^k}{k!}\mathbf{e}^{-\lambda t}$. Среднее число заявок, поступающих за время t, составляет λt. Математическое ожидание числа заявок, поступающих за единицу времени, называется интенсивностью потока ν, для простейшего потока $\nu = \lambda$.

Во многих случаях сумма большого числа малых стационарных потоков близка к простейшему потоку. Это положение часто используется в теории телетрафика. Оно помогает существенно упростить анализ систем массового обслуживания. Если показать, что поток заявок в сети с коммутацией пакетов пуассоновский, то методика расчета *IP*-сетей может быть разработана на основе уже имеющихся результатов для сети с коммутацией каналов. В противном случае возникает необходимость в разработке новой методики расчета.

Для определения характера голосового трафика в сети *NGN* были решены две основные подзадачи, а именно:

- построена модель сети, исследованы значения загруженности узла *IP*-пакетами в каждую единицу времени для разных значений λ;
- проведена проверка того, являются ли данные, полученные в результате функционирования модели, выборкой из распределения Пуассона.

Описание алгоритма

Для решения первой подзадачи был разработан и реализован следующий алгоритм.

1. Прогнозируется поступление звонков на все время исследования.
   - 1.1 Заполняется массив $A(i)$, $A(i) = A(i - 1) + t$, где i — номер поступившего звонка; $A(i)$ — время, когда поступил этот звонок.
   - 1.2 Значения массива $A(i)$ накладываются на решетку с шагом 0,02 с следующим образом. Вводим массив F, $F(j)$ равно числу заявок, поступивших в период времени $[j, j + 0{,}02]$. Получаем функцию $F(j) = k$, где j — единица времени, $j \cdot 50 = 1$ с, k — число поступивших в эту единицу времени вызовов.
2. Запускается процесс моделирования разговоров (включаем счетчик времени j).
   - 2.1 Если $F(j) > 0$, то определяются характеристики для новых вызовов.
     - 2.1.1 Случайным образом определяются два абонента. Если один из них в момент выбора говорит, то автоматически выбирается другой абонент, пока не найдется не говорящий. Это действие правомерно, так как по исходным данным гипотеза о пуассоновском потоке вызовов подтверждена измерениями при условии, что ни один вызов не заканчивается соединением вследствие занятости вызываемого абонента или его отсутствия. Время разговора $t_1 \in [30, 300]$, равномерно распределенное. Это предположение также является правомерным, так как время исследования (время работы модели) больше среднего времени разговора примерно в 1500 раз. Промежуток времени между 30 и 300 с выбран на основе статистики для телефонной сети общего пользования, представленной в работе [3]. В качестве параметров принимаем $\alpha, \beta \in \in [0{,}3; 0{,}6]$, $\gamma \in [0{,}05; 0{,}15]$.
     - 2.1.2 Происходит определение: кто, когда и сколько будет говорить. Время задается как случайная величина с равномерным распределением из отрезка $[1, t]$, где t не превосходит длины всего разговора. В массив $B(m, n) = l$, где m — номер абонента, n — единица времени ($n \cdot 50 = 1$ с), записывается: $l = 1$ в случае, если абонент говорит (то есть передается *IP*-пакет); $l = -1$ в случае, если он принимает *IP*-пакет; $l = 0$ в случае, если пакеты не передаются, т. е. оба абонента молчат.
   - 2.2 Вычисляется сумма $|B(m, n) = l|$ по всем активным абонентам функционирующей модели в каждый момент и делится пополам.

Считается, что время задержки передачи *IP*-пакета равно нулю. Таким образом, в каждую единицу времени одновременно реализуются следующие действия: абонент A передает пакет Y абоненту C, узел принимает пакет Y, узел отдает пакет Y, абонент C принимает пакет Y. Как следствие, число принимаемых пакетов равняется числу отдаваемых пакетов.

В результате формируется функция (массив) $H(j) = p$, которая для каждого j ($j \cdot 50 = 1$ с) выдает число, равное числу *IP*-пакетов, находящихся (поступающих) в это время на узле. Согласно теории телетрафика, скорость передачи *IP*-пакетов по сети в данном случае можно не рассматривать, потому что происходит процесс моделирования загруженности одного узла. Действительно, время задержки *IP*-пакетов в линиях связи местной сети очень мало. При этом требуется посчитать загруженность именно узла, поэтому задержкой можно пренебречь.

Определение характера функции распределения потока заявок

Для решения второй подзадачи необходимо найти функцию распределения выборки из полученного потока. Для определения характера *IP*-трафика на основе полученных результатов используется критерий согласия [4].

Введем необходимые обозначения:

X — выборка из искомого распределения F, полученная в результате работы модели;

F_n^* — эмпирическая функция распределения, построенная по этой выборке;

F_1 — распределение с непрерывной функцией распределения $F_1(y)$.

Введем функционал $\rho(F_n^*, F_1)$, который измеряет расстояние между эмпирическим и теоретическим распределениями. Это расстояние выбирается из следующих далее соображений.

По заданному ε можно найти c, такое, что $\lim\limits_{n\to\infty} P(\rho(F_n^*, F) > c) = \varepsilon$.

При этом $\rho(F_n^*, F_2) \xrightarrow[n\to\infty]{P} \infty$, где F_2 — распределение, отличное от F_1.

Критерий согласия, основанный на таком функционале, строится следующим образом:

$$\delta = \begin{cases} 0, \rho(F_n^*, F_1) \leqslant c, \\ 1, \rho(F_n^*, F_1) > c. \end{cases}$$

Для критерия хи-квадрат выбирается функционал $\chi^{2*} = \sum\limits_{j=1}^{k} \frac{(\nu_j - np_j)^2}{np_j} = \sum\limits_{j=1}^{k} \frac{\left(\frac{\nu_j}{n} - p_j\right)^2 n}{p_j}$, где k — число непересекающихся интервалов $\Delta_1 ... \Delta_k$, покрывающих R; $p_j = F_1(\Delta_j)$ — вероятность попадания в эти интервалы для распределения F_1; ν_j — число элементов выборки, попавших в интервал Δ_j.

Таким образом, можно построить критерий согласия хи-квадрат, который будет использоваться для определения функции распределения, полученной в результате моделирования выборки, а именно:

$$\delta = \begin{cases} 0, \chi^{2*} < c_{1-\varepsilon}^{k-1} \\ 1, \chi^{2*} \geqslant c_{1-\varepsilon}^{k-1}, \end{cases}$$

где c — квантиль уровня $1 - \varepsilon$ распределения χ^2_{k-1}.

Применим известную теорему Пирсона:

если $F = F_1$, то $\chi^{2*} \underset{n\to\infty}{\Longrightarrow} \chi^2_{k-1}$.

Для проверки того, является ли функция распределения выборки пуассоновской, используем тот факт, что вероятность попадания k вызовов в промежуток времени t при пуассоновском распределении равна $\pi_k(a, a + t) = \frac{(\lambda t)^k}{k!} \mathbf{e}^{-\lambda t}$.

Рассмотрим эту функцию при произвольном a и фиксированном $t = 0{,}02$ с. Рассматривается выборка для $\lambda = 0{,}3$ и подсчитывается ее среднее значение. Можно предположить, что выборка имеет пуассоновское распределение с параметром $\lambda^* = 38$, т. е. $\lambda t = 38$, $n = 1\ 320\ 000$. В случае, если распределение с таким λ^* не является пуассоновским, то рассматриваются те $\lambda^* \pm 1$, для которых $\rho(F_n^*, F_1)$ будет меньше, чем для λ^*. Так повторяем до тех пор, пока таких $\lambda^* \pm 1$ не останется. В результате получим λ^*, для которого $\rho(F_n^*, F_1)$ является наименьшей. Если и в этом случае распределение будет не пуассоновским, то этот факт означает, что распределение потока заявок имеет другую функцию распределения.

По критерию хи-квадрат функция распределения полученной выборки является пуассоновской. Для наглядности полученных результатов на одном рисунке (рис. 2) построены график распределения Пуассона и график полученной выборки. Видно, что распределение исследуемой выборки лежит близко к распределению Пуассона.

Для рассматриваемого значения параметра λ получен график интенсивности потока заявок (рис. 3), где по оси y — число поступивших в данную единицу времени заявок на узел, по оси x — время в секундах.

Рис. 2. Распределения выборки Пуассона и полученной выборки, $\lambda = 0{,}3$

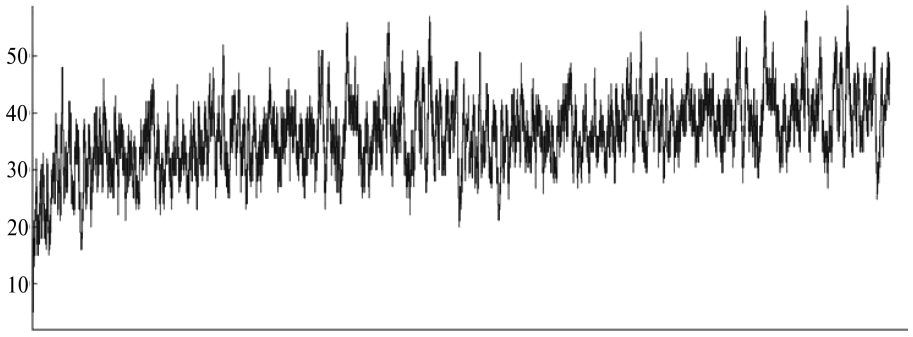


Рис. 3. График интенсивности потока заявок, $\lambda = 0{,}3$

Рис. 4. Распределения выборки Пуассона и полученной выборки, $\lambda = 1$

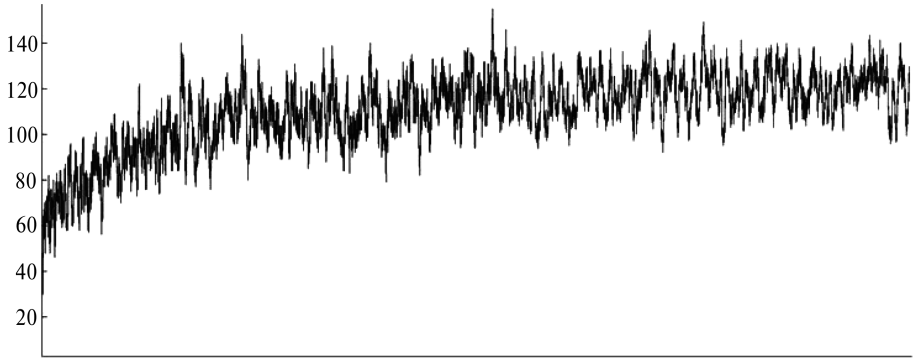


Рис. 5. График интенсивности потока заявок, λ = 1

Действительно, из графика видно, что поток является стационарным, это подтверждает корректность полученных результатов.

Исследуем выборку с другим параметром потока, например λ = 1, и также подсчитаем среднее значение. Выборка имеет распределение Пуассона с λ* = 129 (значение λ* определяем экспериментально, как и в предыдущем случае), т. е. $\lambda t = 129$, $n = 1\,320\,000$. Таким же образом, по критерию хи-квадрат получаем, что функция распределения полученной выборки является пуассоновской.

На одном рисунке (рис. 4) строим график распределения случайной величины с распределением Пуассона и график выборки. Видно, что распределение выборки лежит близко к распределению Пуассона.

Для этого λ получается следующий график интенсивности поступивших в данную единицу времени заявок на узел (рис. 5).

В представленном случае тоже легко заметить стационарность потока вызовов.

Исследования проводились для различных значений λ (в статье приведены только два из них). Во всех рассматриваемых случаях функция распределения потока заявок оказывается пуассоновской.

Выводы

Использовался известный факт, что выборка из функции распределения вызовов *в сети с коммутацией каналов* является пуассоновской. Очевидно, что в этом случае выборка из функции распределения вызовов с учетом равномерности распределения вероятности события — от кого поступит следующая заявка — является пуассоновской для каждого отдельного абонента, с параметром λ/n. При переходе к аналогичным задачам *для сетей с коммутацией пакетов* для достаточно большого времени проведения эксперимента и при фиксированном времени активности абонента (например 50 % от времени разговора, не считая пауз), выборка из функции распределения заявок (*IP*-пакетов) будет также пуассоновской, с параметром $\frac{\lambda}{n}k$, где $k = 100$. Как следствие, выборка из функции распределения заявок суммарно по всем абонентам будет также пуассоновской, с параметром $\lambda \cdot k$.

После добавления в задачу дополнительного условия, связанного с учетом произвольной активности абонентов, для определения функции распределения потока заявок в сетях с коммутацией пакетов потребовалось построить имитационную модель. В результате исследования было показано, что распределение потока заявок в этом случае также является пуассоновским. Наблюдается, что с ростом числа абонентов функция распределения, характеризующая *IP*-трафик, приближается к распределению Пуассона.

На графиках соотношения двух распределений видно небольшое отклонение распределения полученной выборки от распределения Пуассона. Оно возникает вследствие того, что фазы активности абонентов различны. Согласно критерию согласия, это отклонение не является существенным, и таким образом, можно предположить, что поток заявок является простейшим.

Список литературы

1. **Wilkinson N.** Next Generation Network Services. Technologies and Strategies. John Wiley & Sons, Ltd., 2002.

2. **Лившиц Б. С., Фидлин Я. В., Харкевич А. Д.** Теория телефонных и телеграфных сообщений. М.: Связь, 1971.

3. **Корнышев Ю. Н., Пшеничников А. П., Харкевич А. Д.** Теория телетрафика. М.: Радио и связь, 1996.

4. **Боровков А. А.** Математическая статистика: оценка параметров, проверка гипотез. М.: Наука, 1984. 472 с.

Новости IBM

Ученые IBM разработали самый быстрый в мире графеновый транзистор

Ученым IBM удалось создать графеновые полевые транзисторы на наноуровне и продемонстрировать работу графеновых транзисторов на гигагерцовых частотах. Графен — особая форма графита, состоящая из одного слоя атомов углерода, выстроенных в форме гексагональной решетки, аналогичной мелкой проволочной сетке атомарного масштаба.

Ключевое преимущество графена состоит в очень высокой скорости распространения электронов в этом материале, что является необходимым условием создания быстродействующих высокопроизводительных транзисторов. На данный момент рекордной для графенового транзистора является тактовая частота 26 ГГц; при этом длина затвора транзистора составляет 150 нм.

Исследователи IBM считают, что производительность графеновых транзисторов можно дополнительно увеличить за счет улучшения диэлектрических свойств затвора. По их мнению, оптимизация графенового транзистора и уменьшение длины его затвора до 50 нм позволит достичь рабочих частот уровня терагерц.

УДК 004.7

В. В. Наумова, д-р геол.-минер. наук, зав. лаб.,
Дальневосточный геологический
институт ДВО РАН,
А. А. Сорокин, канд. техн. наук, нач. Центра,
Институт геологии и природопользования
ДВО РАН,
И. Н. Горячев, мл. науч. сотр.,
Дальневосточный геологический
институт ДВО РАН

Видеоконференцсвязь — мультимедийный сервис корпоративной сети Дальневосточного отделения РАН

Территориальная разобщенность институтов Дальневосточного отделения РАН ставит задачи повышения оптимизации и эффективности управления научными исследованиями на Дальнем Востоке России, а также объединения территориально разрозненных научных сотрудников между собой для интеграции усилий при решении научных задач. Статья посвящена вопросам проектирования и разработки системы видеоконференцсвязи Дальневосточного отделения РАН.

Ключевые слова: *видеоконференцсвязь, корпоративные сети, сетевые мультимедийные сервисы, научный сервис в Интернет, виртуальные научные конференции, прямая трансляция в Интернет.*

Информация и процессы, связанные с ее использованием, занимают особое место в науке. Информационные технологии порождены наукой и направлены, в первую очередь, на создание информационной среды для науки, образования, наукоемких технологий, а также для промышленности и других сфер деятельности человека. Поэтому информатизация науки это, по существу, современная форма внедрения результатов науки в практику и ее взаимодействия с обществом.

Дальневосточное отделение РАН, в силу протяженности занимаемой территории и удаленности от центра России, нуждается в более тщательном подходе при проектировании и построении телекоммуникационной инфраструктуры, чем другие территориальные отделения Российской академии наук. Удаленность Отделения от научных и образовательных центров России создает проблемы с получением научной и специализированной информации, которая необходима для выполнения исследований на современном уровне.

Научные центры Отделения объединяют институты и организации, расположенные в городах Владивосток, Хабаровск, Благовещенск, Магадан, Петропавловск-Камчатский, Южно-Сахалинск. Отдельные институты ДВО РАН работают в городах Биробиджан, Комсомольск-на-Амуре; поселках Горнотаежное (Приморский край), Паратунка (Камчатская область). Филиалы ряда институтов расположены в г. Анадырь (Чукотский АО), пос. Стекольный (Магаданская область), пос. Мыс Шмидта (Чукотский АО), с. Забайкальское (Хабаровский край). Таким образом, институты и организации Дальневосточного отделения РАН расположены на огромной территории, равной 1/4 территории Российской Федерации, временна́я протяженность которой — четыре часовых пояса.

Корпоративная сеть ДВО РАН, развивающаяся с первой половины 90-х годов прошлого века и по настоящее время, является одним из важнейших факторов, определяющих эффективность работы Отделения. Ее структура ориентирована на интеграцию в информационной сфере всех территориально разрозненных научных подразделений Отделения [1]. Сеть эксплуатируется и развивается Дальневосточным отделением в рамках целевой программы ДВО РАН "Информационно-телекоммуникационные ресурсы ДВО РАН", утвержденной Президиумом ДВО РАН в 2004 году.

Корпоративная сеть ДВО РАН является региональной академической сетью, объединяющей большую часть научных институтов и организаций Дальневосточного отделения РАН.

Каркас Корпоративной сети и региональная инфраструктура построены на основе современных технологий передачи данных с использованием наземных и спутниковых каналов связи, волоконно-оптических линий, беспроводных оптических технологий (FSO), новейших стандартов передачи данных XDSL (G. Shdsl.bis) [2]. Внедряемые системы управления и контроля трафика позволяют обеспечить эффективное функционирование стандартных и корпоративных сервисов создаваемой сети.

В настоящее время в состав Корпоративной сети входят 34 института и организации Отделения, в том числе из научных центров: Приморского — 12; Хабаровского — 8; Амурского — 3; Северо-Восточного — 3; Камчатского — 4; Сахалинского — 4.

Корпоративная сеть ДВО РАН имеет два внешних канала: в г. Хабаровске — 16 Мбит/с (оператор связи — ОАО "Ростелеком") и в г. Владивостоке — 10 Мбит/с (оператор связи — ЗАО "Компания Трансгелеком"). Интеграция сегментов сетей научных центров и формирование единой транспортной инфраструктуры осуществляется на основе каналов связи, арендуемых у коммерческих операторов (ОАО "Ростелеком", ФГУП "РТРС", ОАО "Дальсвязь") (рис. 1).

Организованы прямые каналы между Корпоративной сетью ДВО РАН и сетями наиболее крупных учебных заведений в городах: Владивосток (ДВГУ, ДВГТУ, ВГУЭС); Хабаровск (ТОГУ), Благовещенск (АмГУ). Корпоративная сеть Отделения интегрирована с телекоммуникационной инфраструктурой Геофизической

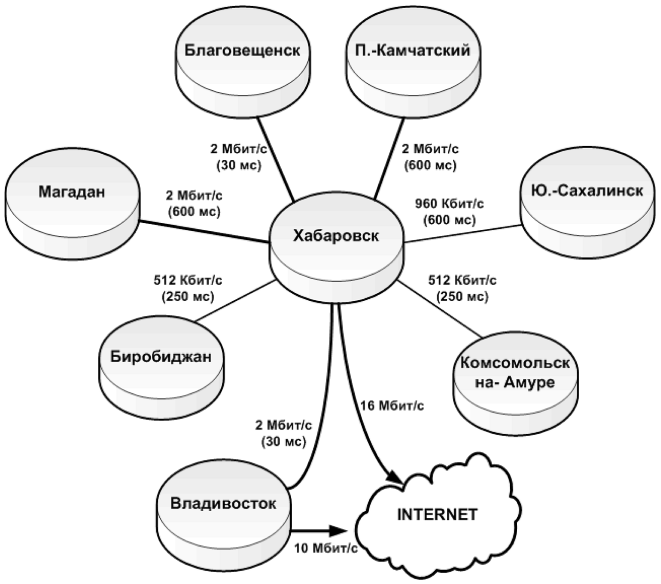


Рис. 1. Общая схема Корпоративной сети Дальневосточного отделения РАН

службы РАН в г. Петропавловск-Камчатский. Интеграция сети Отделения с Дальневосточными филиалами геофизической службы РАН, осуществляющими сейсмический мониторинг на территории Дальнего Востока России, реализована в целях повышения эффективности фундаментальных исследований в области наук о Земле для доступа сотрудников институтов ДВО РАН к данным сейсмических наблюдений.

На основе соглашения между ДВО РАН и Федеральным агентством по образованию РФ и Вузтелекомцентра на базе Амурского и Северо-Восточного научных центров в городах Благовещенск и Магадан организованы точки входа КС ДВО РАН в Федеральную университетскую компьютерную сеть России RUNNet с использованием спутниковых VSAT-станций.

Сеть ДВО РАН предоставляет своим пользователям все основные базовые сетевые сервисы и ресурсы, поддержка которых в настоящее время осуществляется на основе региональных узлов сети и институтов Отделения.

Территориальная разобщенность институтов Дальневосточного отделения РАН ставит задачи повышения оптимизации и эффективности управления научными исследованиями на Дальнем Востоке России, а также объединения территориально разрозненных научных сотрудников между собой для интеграции усилий при решении научных задач.

Для решения этих и других задач в 2006 году в Дальневосточном отделении РАН построена Система видеоконференцсвязи ДВО РАН (СВКС ДВО РАН), являющаяся мультимедийным сервисом Корпоративной сети ДВО РАН (рис. 2). Общение с помощью видеоконференцсвязи (ВКС), когда во время сеанса участники могут не только видеть и слышать друг друга, но и обмениваться данными и обрабатывать их в режиме реального времени, позволяет увеличить эффект восприятия информации до 90 %. По этой причине решения ВКС считаются одним из мощных инструментов повышения эффективности научных исследований и представляют собой качественно новый уровень коммуникаций, объединяя технологические достижения в компьютерной области, телефонии и телевидении.

При проектировании системы были поставлены следующие основные задачи:

- реализация как передачи и приема видео- и аудиосигналов, так и возможность качественного показа графических изображений и презентаций;
- проведение видеоконференций между институтами и организациями Дальневосточного отделения РАН и высшими учебными заведениями Дальнего Востока, а также другими научными и образовательными организациями России и мира (двухсторонние, коллективные);
- организация прямой трансляции в Интернет региональных, всероссийских и международных конференций и мероприятий, проводимых Дальневосточным отделением РАН;
- возможность записи сеансов видеоконференцсвязи для последующей трансляции в Интернет.

Важным принципиальным решением, которое было принято при проектировании СВКС ДВО РАН, было решение об оборудовании конференц-залов институтов оборудованием видеоконференцсвязи. Это решение связано с основной задачей, поставленной при создании СВКС ДВО РАН, — объединение территориально разрозненных научных сотрудников между собой для интеграции усилий при решении научных задач. И именно это решение делает СВКС ДВО РАН системой видеоконфе-

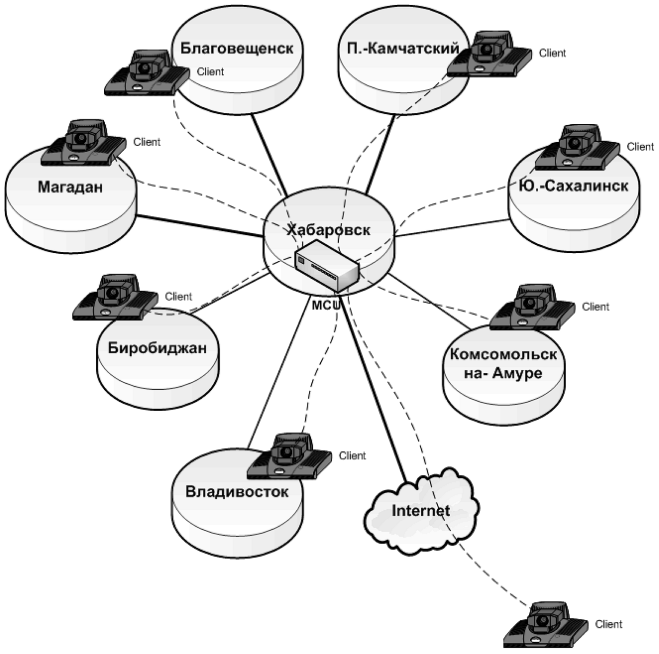


Рис. 2. Схема Системы видеоконференцсвязи ДВО РАН

ренцсвязи для увеличения эффективности научных исследований. При таком подходе также решается и задача повышения оптимизации и эффективности управления научными исследованиями на Дальнем Востоке России.

Принятые при построении СВКС ДВО РАН проектно-технические решения обеспечивают возможность дальнейшего развития, модернизации и масштабирования.

Сеть ВКС ДВО РАН — это программно-аппаратная инфраструктура, действующая поверх Корпоративной сети ДВО РАН. Она включает в себя терминалы ВКС, сервер многоточечной связи (MCU), шлюзы, серверы записи и хранения данных, различное мультимедийное оборудование, функционирующее на основе лицензионного программного обеспечения общего и специального назначения.

Техническое решение по организации СВКС учитывает территориально распределенный характер и топологию корпоративной сети ДВО РАН и пропускную способность ее каналов связи.

Важным этапом при создании СВКС ДВО РАН было формирование ядра системы, которое обеспечивает унифицированный подход к управлению и последующей модернизации системы.

В состав СВКС ДВО РАН входят следующие компоненты:

- устройство многоточечной связи MCU, установленное в техническом центре Корпоративной сети ДВО РАН (г. Хабаровск);
- программно-аппаратные комплексы видеоконференцсвязи, установленные в конференц-залах всех научных центров ДВО РАН в городах: Владивосток, Хабаровск, Магадан, Петропавловск-Камчатский, Благовещенск, Южно-Сахалинск;
- мобильный программно-аппаратный комплекс видеоконференцсвязи.

Параметры системы в целом, так и входящего в него оборудования полностью соответствуют стандартам и рекомендациям ITU-T (Международный союз электросвязи — Сектор стандартизации) H.323. Это обеспечивает бесконфликтную работу с сетевым и мультимедийным оборудованием других фирм-производителей.

Оборудование СВКС ДВО РАН совместимо со средствами защиты информации, передаваемой по *IP*-сетям общего пользования, и сертифицировано для использования в РФ. Оборудование содержит средства и технологии, позволяющие минимизировать проблемы качества при передаче и приеме видео- и аудиоданных.

Решения, использованные при реализации СВКС ДВО РАН, базируются на открытых документированных стандартах передачи видеоинформации и допускают подключение терминального оборудования видеоконференцсвязи других производителей.

В качестве устройства многоточечной связи (MCU) — выбран Codian MCU-4210. Сервер MCU СВКС ДВО РАН обеспечивает следующие возможности:

- подключение в одной конференции до 20 абонентов СВКС ДВО РАН по каналам *IP*-сети на возможной скорости от 56 Кбит/с до 2 Мбит/с для каждого канала;
- добавление и удаление участников без прерывания сеансов видеоконференции;
- в режиме "Непрерывного присутствия" CP обеспечивает одновременное выведение на экран до 16 изображений абонентов;
- подключение к сеансу СВКС абонентов возможно на разных скоростях подключения, и работающих на разных протоколах аудио/видео сжатия и т. д.

Сервер MCU поддерживает возможность удаленного управления всеми функциями через сеть передачи данных.

При необходимости система может работать по комбинированной топологии и имеет возможность работы с несколькими MCU внутри системы.

Для записи и трансляции используется специализированное функционально законченное устройство Codian IP VCR 2210. Данное устройство обрабатывает потоки мультимедийной информации с использованием специализированных процессоров, что обеспечивает максимально высокое качество записи и трансляции.

В качестве регионального терминального оборудования использованы Polycom VSX 8400 — видеотерминалы профессионального уровня, с возможностью интегрирования с разнообразными аудио- и видеосистемами.

Терминальное оборудование залов для организации СВКС ДВО РАН реализовано на базе технологической платформы, включающей в себя модуль кодека видеоконференцсвязи с подключенным к нему специализированным оборудованием.

Терминальные устройства имеют следующие характеристики по реализации видеоконференции:

- обеспечивают видеоконференцсвязь на скорости соединения от 384 Кбит/с до 2 Мбит/с;
- поддерживают передачу видеоизображения с частотой кадров до 25 кадров/с (PAL);
- осуществляют автоматическую синхронизацию звука с артикуляцией выступающего;
- поддерживают технологии трансляции сетевых адресов (NAT);
- поддерживают межсетевой экран с фиксированными портами TCP/IP;
- поддерживают функцию "картинка в картинке" (PiP).

Системы видеоконференций, звукоусиления, видеопроекций — основные компоненты оснащения конференц-залов. Решения направлены на создание сбалансированного комплекса видео- и аудиокомпонентов для оперативной и комфортной работы.

Конференц-залы следующих институтов и организаций Дальневосточного отделения РАН определены в качестве базовых узлов СВКС ДВО РАН:

- Президиума Дальневосточного отделения РАН, г. Владивосток (рис. 3);
- Президиума Хабаровского научного центра ДВО РАН, г. Хабаровск;

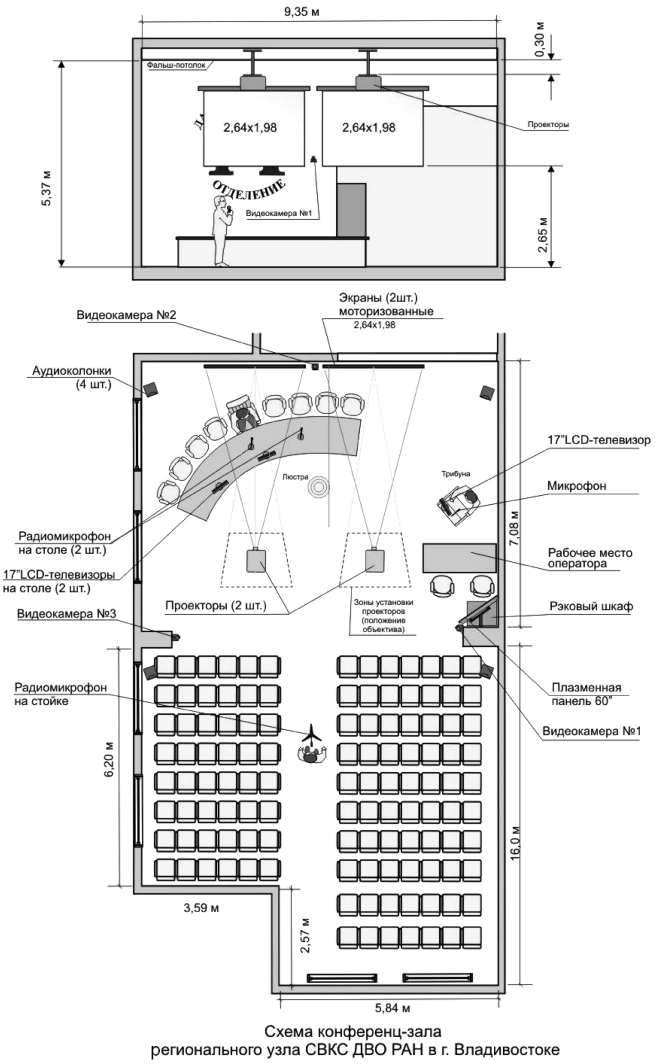


Схема конференц-зала
регионального узла СВКС ДВО РАН в г. Владивостоке

Рис. 3. Схема конференц-зала Президиума ДВО РАН в г. Владивостоке — регионального узла СВКС Дальневосточного отделения РАН

- Института геологии и природопользования ДВО РАН, г. Благовещенск;
- Северо-Восточного комплексного научно-исследовательского института ДВО РАН, г. Магадан;
- Института вулканологии и сейсмологии ДВО РАН, г.Петропавловск-Камчатский;
- Института морской геологии и геофизики ДВО РАН, г. Южно-Сахалинск.

Все они оснащены профессиональным оборудованием на постоянной основе.

Весь парк оборудования видеоконференцсвязи ДВО РАН допускает удаленное управление и администрирование через встроенный web-интерфейс с рабочего места оператора/администратора.

На 2008 год предусмотрено совершенствование координации региональных операторов и общего управления СВКС ДВО РАН. Для этого планируется решить следующие задачи:

- организация голосового взаимодействия операторов СВКС на базе шлюзов *IP*-телефонии с использованием AudioCodes MP-112;
- организация дистанционного управления сервером видеоконференций с использованием Codian Director;
- организация дистанционного управления терминалами с использованием Polycom GMS.

Ядро системы видеоконференцсвязи, помимо решения задачи развертывания самостоятельного централизованного информационного сервиса Дальневосточного отделения РАН, формирует среду прямого обращения к ее возможностям (программным, аппаратным) и их интеграции в другие службы, например системы документооборота, телефонии и т. д. Такие опции возможны за счет поддержки протокола XML-RPC.

Группа сопровождения СВКС ДВО РАН создана в 2006 году. В состав группы входят руководитель группы и по одному человеку из каждого научного центра ДВО РАН. Группа обеспечивает бесперебойную работу СВКС ДВО РАН.

Основные направления применения СВКС в Дальневосточном отделении РАН:

- оперативное проведение заседаний Президиумов и Общих собраний ДВО РАН, различных комиссий и редакционных коллегий журналов и т. д.;
- пресс-конференции руководителей ДВО РАН;
- научные конференции и семинары как внутри ДВО РАН, так и с научными организациями и университетами России и мира;
- лекции и семинары для молодых сотрудников и аспирантов, которые могут быть проведены учеными из различных регионов мира;
- совместная работа над общими проектами и программами;
- демонстрация возможностей новейшего аналитического оборудования центров коллективного пользования.

В настоящее время СВКС ДВО РАН является эффективным инструментом управления и взаимодействия.

Таким образом, при внедрении СВКС ДВО РАН в текущую деятельность Дальневосточного отделения РАН получены следующие основные результаты.

1. СВКС ДВО РАН ускоряет принятие решений по ключевым вопросам, требующим присутствия всего руководящего состава Дальневосточного отделения РАН, а также существенно сокращает финансовые затраты на их проезд в г. Владивосток, где расположен Президиум Дальневосточного отделения РАН.

2. Ежемесячные заседания редколлегий научных журналов Дальневосточного отделения РАН (например, журнала "Тихоокеанская геология") в режиме видеоконференцсвязи позволяют повысить эффективность обсуждения статей членами редакционной коллегии.

3. Повышается уровень научных конференций, проводимых в Дальневосточном отделении РАН, вследствие полученной возможности проводить в режиме видеоконференцсвязи включения ведущих российских и мировых научных центров и университетов. Например,

в рамках Международной конференции "Россия и Америка в Тихоокеанском регионе: проблемы и решения", которую в июне 2007 г. проводили Институт истории ДВО РАН и Генконсульство США в г. Владивостоке, прошла видеоконференция с университетом г. Аугуста (США).

4. В режиме видеоконференцсвязи решаются и обсуждаются многие вопросы совершенствования Корпоративной сети ДВО РАН и ее основных сервисов.

Высокое качество звука и полноэкранное видео, возможность оперативного обмена данными и документами делают видеоконференции в Дальневосточном отделении РАН мощным инструментом с широчайшим спектром практического применения.

Список литературы

1. **Ханчук А. И., Наумова В. В., Сорокин А. А.** Корпоративная сеть ДВО РАН — высокотехнологичная интеграция научных подразделений // Вестник РАН. 2008. № 4.

2. **Ханчук А. И., Сорокин А. А., Наумова В. В., Нурминский Е. А., Смагин С. И., Ворошин С. В., Казанцев В. А.** Корпоративная сеть Дальневосточного отделения РАН // Вестник ДВО РАН. 2007. № 1 (131). С. 3—20.

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ И ОБРАБОТКА СИГНАЛОВ

УДК 621.396.98(100):519.673

В. Е. Алексеев, мл. науч. сотр.,
Научно-исследовательский институт
вычислительных средств и систем управления
(НИИ ВС и СУ),
А. Н. Соловьев, д-р техн. наук, проф.,
гл. науч. сотр.,
Институт проблем проектирования
в микроэлектронике
Российской академии наук (ИППМ РАН),
e-mail: valerbas@mail.ru

Многоантенные GPS-системы с дециметровой точностью позиционирования*

Рассматривается один из наиболее эффективных подходов в работе многоантенных GPS-систем с дециметровой точностью позиционирования на основе калмановской фильтрации. Приводятся результаты экспериментальной оценки предлагаемого подхода при различных условиях работы: в статике и в динамике.

Ключевые слова: *многоантенные GPS-системы, фильтр Калмана, фазовая неоднозначность, плавающее решение, двойные фазовые разности.*

1. Многоантенные системы на основе GPS

В настоящее время в различных областях науки и техники все более активное применение находят многоантенные (сокращенно МА, от слова *Multi-Antenna*) системы на основе спутниковых сигналов GPS. Главным достоинством данных систем является высокий уровень точности определения относительных координат GPS-приемников. В настоящее время достигнут дециметровый и сантиметровый уровни точности при использовании фазовых измерений. Это позволяет успешно применять данные системы как для *высокоточного определения координат*, так и для *высокоточного определения угловой ориентации* вектора, связывающего фазовые центры антенн GPS-приемников.

Основной проблемой в использовании фазовых измерений является наличие *фазовой неоднозначности* (ФН) — неизвестного целого числа длин волн, укладывающихся по пути распространения сигнала от спутника к антенне приемника.

Проведенный анализ показывает, что в основе всех существующих алгоритмов устранения (разрешения) фазовых неоднозначностей лежит последовательное выполнение двух основных шагов:

- нахождение "плавающего" решения;
- нахождение "фиксированного" решения.

Под "плавающим" решением понимается набор *дробных значений* числа длин волн, максимально близких к истинным целочисленным значениям. Погрешность определения относительных координат по плавающему решению составляет 20—50 см.

Под "фиксированным" понимается набор целочисленных значений, образующих истинное решение, на основе которого осуществляется полное устранение неоднозначностей. Погрешность определения относительных координат по фиксированному решению составляет около 1 см.

Качество и сходимость "фиксированного" решения в значительной мере зависит от качества получения "плавающего" решения. Целью данной статьи являются:

* Данная статья написана в рамках программы фундаментальных исследований отделения информационных технологий и вычислительных систем РАН "Фундаментальные основы информационных технологий и систем".

- описание общего подхода к нахождению плавающего решения, на основе которого достигается определение координат с дециметровой точностью;
- описание конкретной реализации предлагаемого подхода с использованием калмановской фильтрации;
- сравнительная оценка эффективности предлагаемого подхода на основе реальных экспериментальных данных при различных условиях работы: в статике и в динамике.

2. Основная идея поиска плавающего решения

Базовым элементом математического аппарата, который используется при поиске плавающего решения, являются *двойные фазовые разности*, получаемые на основе *фазовых измерений*, а также *одинарных фазовых разностей*.

2.1. Структура фазового измерения

Фазовые измерения для двух приемников k и m от спутника p (рис. 1) можно представить в виде

$$\begin{aligned}\Phi_k^p(t) &= \varphi_k^p(t) - \varphi^p(t) + N_k^p + \\ &+ S_k + f\tau_p + f\tau_k + \beta_{iono} + \delta_{tropo};\\ \Phi_m^p(t) &= \varphi_m^p(t) - \varphi^p(t) + N_m^p + \\ &+ S_m + f\tau_p + f\tau_m + \beta_{iono} + \delta_{tropo},\end{aligned} \quad (1)$$

где p — индекс спутника, от которого получен сигнал; $\varphi^p(t)$ — фаза сигнала, излученного со спутника p как функция времени; $\varphi_k^p(t)$ и $\varphi_m^p(t)$ — фазы сигнала от спутника p, измеренные приемниками k и m; N_k^p, N_m^p — неизвестные целые числа циклов фазы сигнала от спутника p к приемнику k и от спутника p к приемнику m соответственно; S_k, S_m — шумы измерения фазы от различных источников (переотражение сигнала, случайные ошибки) для приемников k и m соответственно; f — несущая частота излучаемого сигнала; τ_p, τ_k, τ_m — уход часов спутника p, приемника k и приемника m соответственно; β_{iono} — ионосферная задержка; δ_{tropo} — тропосферная задержка.

Аналогично (1) могут быть получены выражения $\Phi_k^q(t)$ и $\Phi_m^q(t)$ для фазовых измерений двух приемников k и m от спутника q.

2.2. Одинарные фазовые разности

Геометрическое представление *одинарной фазовой разности* изображено на рис. 2, где присутствуют следующие элементы:

- вектор базы b, направленный от приемника m к приемнику k;
- единичные направляющие вектора e^p и e^q, указывающие на спутники p и q соответственно;
- линии визирования от приемников m и k на спутники p и q;
- одинарные фазовые разности SD_{km}^p и SD_{km}^q.

Одинарную разность между измерениями двух приемников k и m сигналов спутника p можно записать следующим образом:

$$SD_{km}^q = \Phi_k^q - \Phi_m^q = \varphi_{km}^q + N_{km}^q + S_{km}^q + f\tau_{km}. \quad (2)$$

Аналогичную одинарную разность можно записать для спутника q:

$$SD_{km}^p = \Phi_k^p - \Phi_m^p = \varphi_{km}^p + N_{km}^p + S_{km}^p + f\tau_{km}. \quad (3)$$

2.3. Двойные фазовые разности

Двойная разность, получаемая на основе двух одинарных разностей для спутников p и q, определяется следующим образом:

$$DD_{km}^{pq} = SD_{km}^p - SD_{km}^q = \varphi_{km}^{pq} + N_{km}^{pq} + S_{km}^{pq}. \quad (4)$$

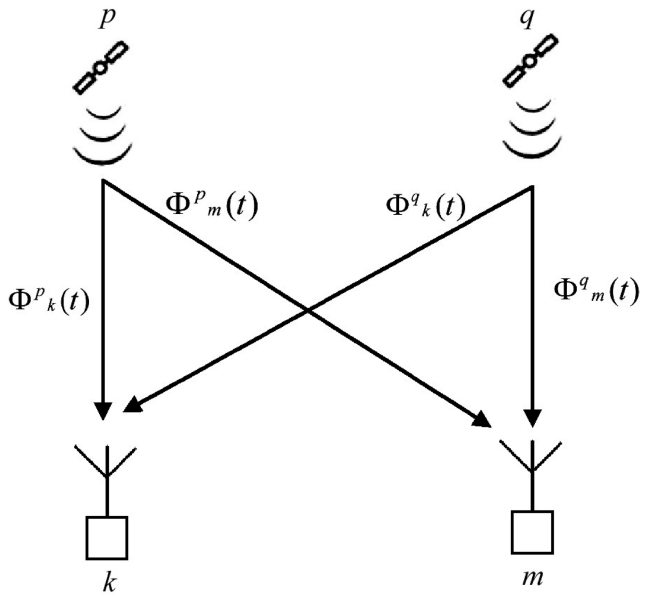


Рис. 1. Фазовые измерения для двух приемников (m, n) от двух спутников (p, q)

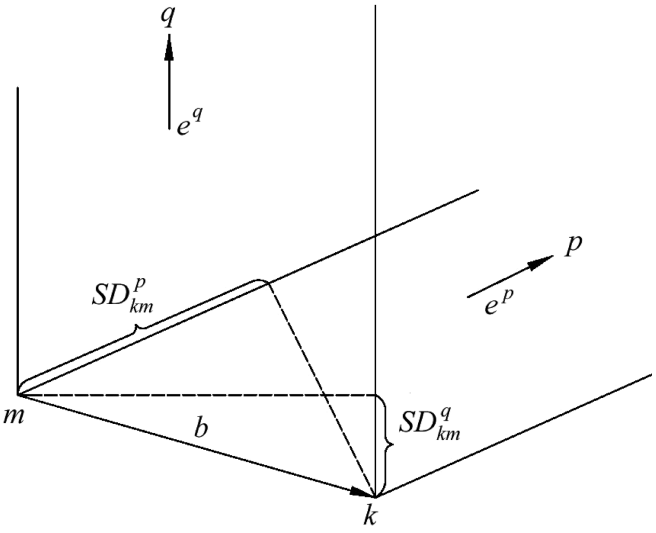


Рис. 2. Геометрическое представление одинарных фазовых разностей

Полученное выражение (4) показывает, что с помощью двойных разностей полностью компенсируются уходы часов приемников и спутников, а также в значительной мере подавляется ошибка, связанная с тропосферными и ионосферными задержками.

В целях составления основного уравнения для двойных разностей примем выражение (4) в качестве левой части уравнения, а для получения правой части используем геометрическое представление двойной разности, показанное на рис. 2. Из рисунка видно, что проекция вектора b на линию визирования от m к p может быть представлена в виде скалярного произведения b на единичный вектор e^p. Данная проекция есть не что иное, как SD^p_{km}. Аналогично, скалярное произведение вектора b на единичный вектор e^q равно SD^q_{km}. С учетом этого можно переписать (2) и (3) в следующем виде:

$$SD^p_{km} = (be^p)\lambda^{-1};$$

$$SD^q_{km} = (be^q)\lambda^{-1}.$$

Тогда геометрическое представление двойной разности будет иметь вид

$$DD^{pq}_{km} = (be^p)\lambda^{-1} - (be^q)\lambda^{-1} = (be^{pq})\lambda^{-1}. \qquad (5)$$

Объединив (4) и (5), получим **основное уравнение для двойных разностей:**

$$\varphi^{pq}_{km} + N^{pq}_{km} + S^{pq}_{km} = (be^{pq})\lambda^{-1}, \qquad (6)$$

где be^{pq} — скалярное произведение неизвестного вектора базы на разность между единичными направляющими векторами на спутники p и q.

Для вывода вычислительного алгоритма будем использовать только детерминированные составляющие из выражения (6), а случайную шумовую составляющую S^{pq}_{km} опустим. Для наглядности также опустим индексы $_{km}$, а индексы буквенного вида ^{pq} заменим на цифровые — вида $_{1,2}$, $_{1,3}$ и т. д., где 1, 2, 3... — номера спутников. Тогда выражение (6) для пяти наблюдаемых спутников будет иметь следующий вид:

$$\begin{bmatrix} DD_{cp1,2} \\ DD_{cp1,3} \\ DD_{cp1,4} \\ DD_{cp1,5} \end{bmatrix} = \underbrace{\begin{bmatrix} e_{1,2x} & e_{1,2y} & e_{1,2z} \\ e_{1,3x} & e_{1,3y} & e_{1,3z} \\ e_{1,4x} & e_{1,4y} & e_{1,4z} \\ e_{1,5x} & e_{1,5y} & e_{1,5z} \end{bmatrix}}_{H} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} \lambda. \qquad (7)$$

Как видно из (7), на основе пяти спутников образуются четыре пары и соответственно четыре двойные разности. Во всех парах присутствует один и тот же, так называемый *опорный* спутник, который обычно выбирается по наивысшему вознесению над горизонтом для обеспечения наилучшего геометрического фактора PDOP (*Position Dilution of Precision*). Индекс *cp* в записи DD_{cp} обозначает "carrier phase", т. е. фазовое измерение. В матричном виде выражение (7) представляется следующим образом:

$$\boldsymbol{DD}_{cp} = \mathbf{H}\boldsymbol{b} + \boldsymbol{N}\lambda. \qquad (8)$$

2.4. Нахождение плавающего решения

В основе нахождения плавающего решения для выражения (8) лежит поиск вещественных значений элементов вектора $\boldsymbol{N} = [N_1, N_2, N_3, N_4]^{\mathrm{T}}$ и вектора $\boldsymbol{b} = [b_x, b_y, b_z]^{\mathrm{T}}$, которые минимизируют функцию:

$$\mathbf{c} = \boldsymbol{DD}_{cp} - \mathbf{H}\boldsymbol{b} - \boldsymbol{N}\lambda \rightarrow \min.$$

При этом число неизвестных ($\boldsymbol{N}$, $\boldsymbol{b}$) больше, чем число уравнений. Увеличение числа наблюдаемых спутников не делает систему разрешаемой, так как число неизвестных также увеличивается. Для нахождения решения используется принцип усреднения, основанный на избыточности измерений. Избыточность достигается за счет проведения измерений в течение нескольких последовательных эпох. Однако для устранения вырожденности системы измерения должны быть независимыми. Независимость измерений достигается за счет изменения со временем геометрии спутников (взаимного расположения спутников на орбите относительно приемника сигналов) и, как следствие, изменения элементов матрицы **H**. В качестве процедуры усреднения возможно использование метода наименьших квадратов. Однако наиболее эффективной реализацией данного метода является использование калмановской фильтрации.

В следующем разделе подробно рассматривается реализация фильтра Калмана для получения плавающего решения.

3. Калмановская фильтрация — ключевая процедура поиска плавающего решения

3.1. Основа фильтра Калмана

Фильтр Калмана представляет собой рекурсивный фильтр, основанный на взвешенном усреднении прогноза и измерения. Данный фильтр позволяет оптимально использовать избыточность входных данных и придавать при этом больший вес тем входным величинам, которые характеризуются меньшей ошибкой. Важным преимуществом фильтра Калмана является то, что источниками входной информации для прогноза и измерения могут являться разные по типу величины, выбор и обоснование которых дается в разделе 3.3. Ниже в разделе 3.2 приводится основная вычислительная схема фильтра. Перед тем как переходить к описанию вычислительной схемы, необходимо отметить, что в предыдущих разделах говорилось о непрерывном случае представления фазового измерения, как о функции времени t. Приведенные выше выражения справедливы и для дискретного случая, когда вместо непрерывного времени t имеется некоторая выборка моментов времени $t_1, t_2, ..., t_n$ с интервалом Δt.

3.2. Вычислительная схема фильтра

На рис. 3 приведена общая вычислительная схема фильтра Калмана. Так как фильтр — рекурсивный и работает в итеративном режиме, на схеме представлено описание одной итерации. Каждая итерация состоит из двух процедур: предсказание (левый блок на схеме) и корректировка (правый блок на схеме). Индексам n и $n-1$ соответствуют два последовательных момента времени измерений: текущий (t_n) и предыдущий (t_{n-1}).

На схеме представлены следующие элементы:

$\boldsymbol{X}_n$ — вектор состояний, характеризующий выходные значения фильтра;

$\mathbf{P}_n$ — матрица ковариации ошибок, характеризующая корреляцию элементов вектора состояний;

$\mathbf{F}$ — матрица перехода от предыдущего шага к текущему;

$\mathbf{Q}_n$ — матрица ошибок, полученных на текущем шаге;

$\mathbf{R}$ — матрица ошибок измерения;

$\mathbf{K}_n$ — коэффициент доверия фильтра Калмана;

$\boldsymbol{Y}_n$ — измерение фильтра Калмана;

$\mathbf{HH}$ — матрица перевода прогноза в пространство измерения;

$\mathbf{I}$ — единичная матрица.

3.3. Выбор прогноза и измерения

Устойчивость фильтра Калмана и его сходимость в значительной степени зависят от выбора *прогноза* и *измерения*. Существуют различные подходы к выбору прогноза и измерения, одним из которых является сглаживание двойных кодовых разностей на основе двойных фазовых, и оценивание параметров $\boldsymbol{b}$ и $\boldsymbol{N}$ по сглаженным значениям. В связи с тем, что кодовые измерения достаточно "грубые", сходимость такого фильтра будет не очень быстрой. Более эффективным подходом является сглаживание двойных фазовых разностей на основе тройных разностей. Данный подход был выбран за основу.

В рассматриваемом варианте вектор состояний представлен следующими элементами:

$$\boldsymbol{X}_n = [b_x,\ b_y,\ b_z,\ N_1,\ N_2,\ ...,\ N_m]^{\mathrm{T}},$$

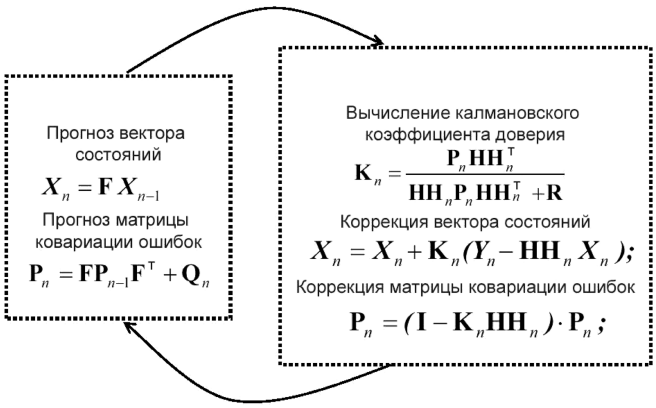


Рис. 3. Общая вычислительная схема фильтра Калмана

где b_x, b_y, b_z — элементы вектора базы $\boldsymbol{b}$; N_1, N_2, ..., N_m — элементы вектора неоднозначностей, а m — число пар спутников.

В качестве **прогноза** выступают двойные фазовые разности, полученные на основе вектора базы $\bar{\boldsymbol{b}}(t_i)$, вычисленного с использованием вектора $\boldsymbol{b}(t_{i-1})$ на предыдущем шаге и приращения $\Delta\boldsymbol{b}(t_i)$ на текущем шаге:

$$\bar{\boldsymbol{b}}(t_i) = \boldsymbol{b}(t_{i-1}) + \Delta\boldsymbol{b}(t_i).$$

Приращение $\Delta\boldsymbol{b}(t_i)$ определяется с использованием тройных фазовых разностей $\boldsymbol{TD}(t_i)$. Последние вычисляются путем вычитания двойных разностей для двух последовательных эпох измерений t_{i-1} и t_i. Конечная формула для определения приращения вектора $\Delta\boldsymbol{b}(t_i)$ выглядит следующим образом:

$$\Delta\boldsymbol{b} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}(\boldsymbol{TD}(t_i) - \Delta\mathbf{H}\boldsymbol{b}(t_{i-1})).$$

В качестве **измерения** фильтра используются двойные фазовые разности, построенные непосредственно по данным, полученным от GPS-приемников.

3.4. Взвешенная оценка прогноза и измерения, выходные значения фильтра

В основе работы фильтра Калмана лежит взвешенная оценка прогноза и измерения, что позволяет оптимально использовать избыточность входных данных и придавать при этом больший вес тем входным величинам, которые характеризуются меньшей ошибкой. Для того чтобы найти взвешенную оценку прогноза и измерения, необходимо перевести их в единую метрику (пространство). Для этого прогнозируемое значение вектора $\bar{\boldsymbol{b}}(t_i)$ проецируют на пространство двойных разностей с помощью специальной матрицы перехода $\mathbf{HH}$. Структура данной матрицы имеет вид

$$\mathbf{HH}_n = \begin{bmatrix} e_{1,2x} & e_{1,2y} & e_{1,2z} & 1 & 0 & 0 & 0 \\ e_{1,3x} & e_{1,3y} & e_{1,3z} & 0 & 1 & 0 & 0 \\ e_{1,4x} & e_{1,4y} & e_{1,4z} & 0 & 0 & 1 & 0 \\ e_{1,5x} & e_{1,5y} & e_{1,5z} & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (9)$$

Матрица (9) приведена для случая с пятью спутниками.

Произведение матрицы перехода $\mathbf{HH}_n$ на прогноз вектора $\bar{\boldsymbol{b}}(t_i)$ дает прогнозируемое значение двойных разностей $\bar{\boldsymbol{DD}}_{cp}$:

$$\bar{\boldsymbol{DD}}_{cp} = \mathbf{HH}_n\bar{\boldsymbol{b}}(t_i).$$

Соответственно, коррекцию вектора состояний можно записать в следующем виде:

$$\boldsymbol{X}_n = \boldsymbol{X}_n + \mathbf{K}_n(\boldsymbol{DD}_{cp} - \bar{\boldsymbol{DD}}_{cp}) =$$
$$= \boldsymbol{X}_n + \mathbf{K}_n(\boldsymbol{DD}_{cp} - \mathbf{HH}_n\bar{\boldsymbol{b}}(t_i)).$$

Выходные значения фильтра, содержащиеся в векторе состояний $\boldsymbol{X}_n$, образуют *плавающее решение*. Оценивая девиацию выходных значений вектора $\boldsymbol{X}_n$, можно судить о сходимости фильтра.

4. Оценка эффективности работы предлагаемого подхода на основе реальных экспериментальных данных

4.1. Методика оценки

Для оценки эффективности предложенного подхода по нахождению плавающего решения был проведен ряд тестов. Тесты проводились с использованием двух одночастотных GPS-приемников К-161, выпущенных ОАО "РИРВ" (http://www.rirt.ru). По результатам обработки собранных экспериментальных данных оценивались характеристики эффективности фильтра для двух разных состояний системы:

- статики (неподвижное состояние приемников);
- динамики (первый приемник — неподвижный, второй — подвижный).

В качестве оцениваемых характеристик эффективности были выбраны следующие величины:

- ошибка определения вектора базы в плавающем решении;
- максимальная девиация элементов неоднозначностей в плавающем решении;
- $t_{\text{схожд}}$ — время схождения фильтра, равное времени достижения максимальной девиации неоднозначностей порогового значения $\boldsymbol{PP}_{\text{порог}}$.

В качестве $\boldsymbol{PP}_{\text{порог}}$ выбрана величина, равная двум длинам волн. При достижении заданного порогового значения задачу нахождения плавающего решения можно считать выполненной, так как границы поиска фиксированного решения сужены до достаточно малых пределов. Ошибка вектора базы в плавающем решении определяется исходя из сравнения данного вектора со значениями, полученными на основе фиксированного решения, имеющего погрешность порядка 1 см.

4.2. Статика

В данном разделе приводятся графики оцениваемых характеристик эффективности фильтра для статичного состояния системы. На рис. 4 отображен график ошибки вектора базы, который представлен тремя составляющими: восток, север и высота.

На рис. 5 представлен график максимальной девиации элементов неоднозначностей, выраженной в длинах волн (λ), в зависимости от времени, а также пороговое значение $\boldsymbol{PP}_{\text{порог}}$ в виде константы.

4.3. Динамика

На рис. 6 представлен график трех составляющих вектора скорости подвижного приемника по трем осям: восток, север, высота. Перемещения приемника осуществлялись в горизонтальной плоскости, т. е. вдоль осей "восток" и "север".

На рис. 7 и 8 представлены графики ошибки вектора базы и максимальной девиации элементов неоднозначностей с пороговым значением, выраженным в виде константы.

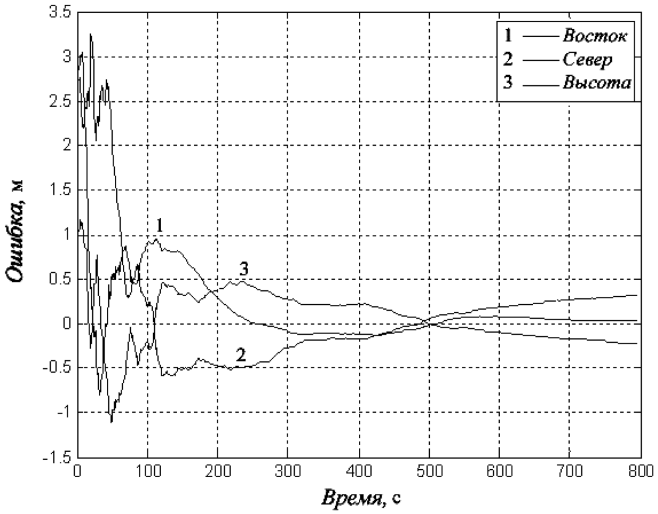


Рис. 4. График ошибки составляющих вектора базы (восток, север, высота) в зависимости от времени. Статика

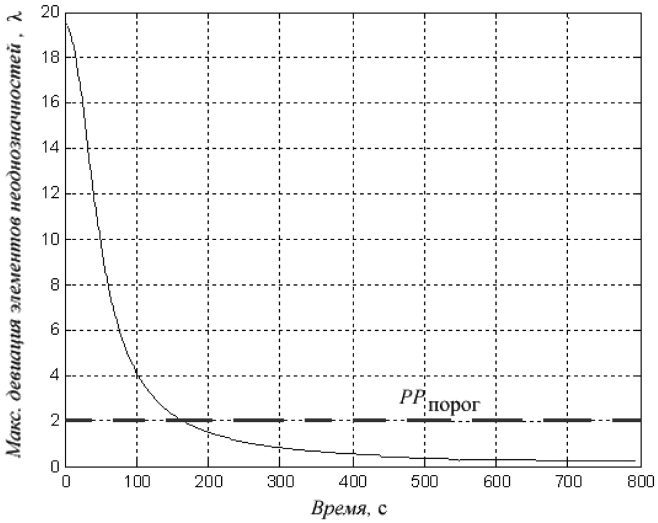


Рис. 5. График зависимости максимальной девиации элементов неоднозначностей от времени. Статика

Сравнение оценок эффективности фильтра для двух состояний системы: статика и динамика

Оценка	Статика	Динамика
Математическое ожидание ошибки вектора базы по модулю		
M ($\Delta b_{\text{восток}}$), м	0,293	0,389
M ($\Delta b_{\text{север}}$), м	0,002	0,012
M ($\Delta b_{\text{высота}}$), м	0,082	0,130
Сигма ошибки вектора базы		
σ ($\Delta b_{\text{восток}}$), м	0,665	0,753
σ ($\Delta b_{\text{север}}$), м	0,349	0,901
σ ($\Delta b_{\text{высота}}$), м	0,464	0,434
Время схождения		
$t_{\text{схожд}}$, с	166	586

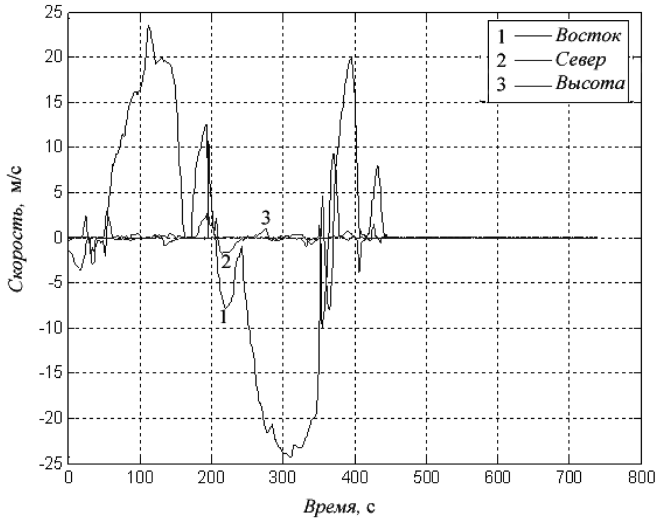


Рис. 6. График составляющих вектора скорости (восток, север, высота) подвижного приемника. Динамика

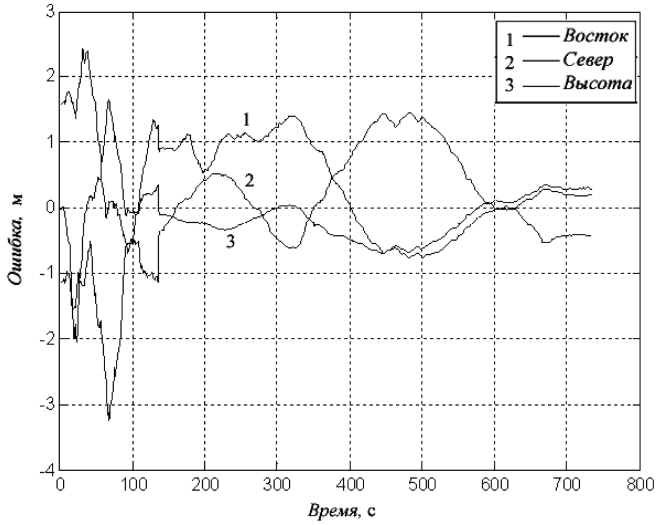


Рис. 7. График ошибки составляющих вектора базы (восток, север, высота) в зависимости от времени. Динамика

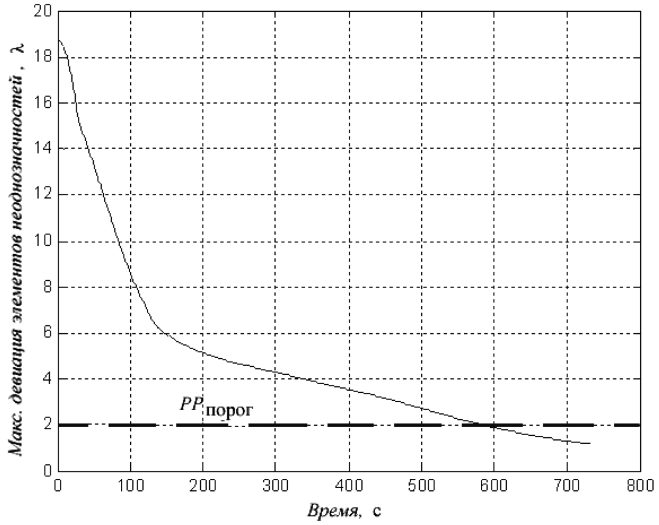


Рис. 8. График зависимости максимальной девиации элементов неоднозначностей от времени. Динамика

4.4. Анализ и сравнительная оценка полученных результатов

Из приведенных графиков видно, что ошибка вектора базы со временем стремится к нулю, а девиация элементов неоднозначностей плавающего решения становится незначительной. Это говорит о том, что фильтр сходится и предложенный алгоритм поиска плавающего решения работает корректно.

На основе проведенного анализа получены числовые оценки сравниваемых параметров для двух состояний системы. Данные оценки представлены в таблице.

Как видно из таблицы, фильтр сходится гораздо быстрее в статике, чем в динамике. Это связано с тем, что при работе фильтра выходные значения вектора базы и вектора элементов неоднозначностей, образующие вектор состояний, связаны и зависят друг от друга. Чем точнее вектор базы, тем точнее определены элементы неоднозначностей, и наоборот. В случае, когда до схождения фильтра возникают перемещения одного из приемников, три составляющие вектора состояний фильтра b_x, b_y, b_z начинают быстро меняться, и если до этого момента они были определены достаточно грубо, то возникающая ошибка перераспределится в остальные элементы вектора состояний, а именно — в элементы неоднозначности. В этом случае процесс приближения элементов вектора состояний к своим истинным значениям сильно замедлится. Для того чтобы фильтр работал так же эффективно в динамике, как и в статике, можно усовершенствовать его путем ввода дополнительных элементов вектора состояний, например вектора скорости.

Заключение

В статье рассмотрен общий подход к нахождению плавающего решения, являющегося одной из ключевых процедур при построении многоантенных GPS-систем с дециметровой точностью позиционирования. Приведены структура фазового измерения, а также принцип построения одинарных, двойных и тройных фазовых разностей, являющихся базовым элементом математического аппарата, лежащего в основе предлагаемого алгоритма. Рассмотрена конкретная реализация данного подхода на основе калмановской фильтрации. Приведены общие структурные элементы фильтра, их связь с решением, получаемым на выходе. Представлена оценка эффективности предлагаемого подхода на основе реальных экспериментальных данных, а также проведен сравнительный анализ работы алгоритма для двух различных состояний системы: в статике и в динамике. Полученные экспериментальные оценки подтвердили корректность и эффективность предложенного подхода.

Список литературы

1. **Kaplan E. D.** Understanding GPS Principles and Applications // Artech House Boston, London. 1996.

2. **Graas F. V., Braasch M.** GPS Interferometric Attitude and Heading Determination: Initial Flight Test Results // Navigation: Journal of The Institute of Navigation. 1991—1992. Vol. 38. N 4.

3. **Rybski P. E.** Mobile Robot Localization and Mapping using the Kalman Filter. http://www.cs.cmu.edu/~robosoccer/cmrobobits/

УДК 621.391

С. В. Дворников, канд. техн. наук,
доц., зам. нач. каф.,
А. Г. Жечев, преподаватель,
Военная академия связи, г. Санкт-Петербург

Демодуляция сигналов на основе обработки их модифицированных частотно-временных распределений

Предлагается метод демодуляции сигналов частотной манипуляции на основе обработки их частотно-временных распределений. Обосновывается целесообразность выбора в качестве базовых билинейных распределений. Предлагаются результаты практического эксперимента, подтверждающие продуктивность разработанного подхода.

Ключевые слова: *демодуляция, частотно-временное распределение, сигналы частотной манипуляции, модифицированная спектрограмма, билинейные распределения.*

Введение

Частотно-временной анализ находит все более широкое применение в технологиях цифровой обработки сигналов. Теоретические работы Коэна [1], получившие творческое развитие в трудах Классена и Мекленбрука [1], позволили по-новому подойти к решению таких задач радиомониторинга, как обнаружение, разделение сигналов, их распознавание и измерение параметров [3—5].

Очевидно, что возможности частотно-временного анализа не ограничиваются его приложением только в рамках указанных задач. Благодаря своим свойствам частотно-временные распределения являются уникальным инструментом, позволяющим обрабатывать тонкую структуру сигналов. А возможность получения на их основе модифицированных форм, с улучшенными свойствами помехоустойчивости, расширила сферу применения частотно-временных представлений. В частности, анализ результатов работы [6] позволил предположить целесообразность построения на их базе алгоритмов демодуляции.

В настоящей работе предлагаются результаты применения частотно-временного подхода к демодуляции сигналов частотной манипуляции.

Постановка задачи

Вопросам поиска универсальных алгоритмов демодуляции радиосигналов в комплексах мониторинга традиционно отводится особое место. В первую очередь, это связано с необходимостью их работы с передачами, использующими широкий спектр самых разнообразных модуляционных форматов. Очевидно, что использование в указанной ситуации классических алгоритмов, опирающихся на методы оптимальной фильтрации и корреляционной обработки, приведет к необходимости наличия соответствующего ассортимента аппаратуры, или же значительного перечня соответствующего программного обеспечения.

Таким образом, проблема поиска универсальных алгоритмов, позволяющих проводить демодуляцию широкого класса сигналов, остается актуальной и значимой задачей для радиомониторинга.

В связи с этим целью данной работы является *разработка метода демодуляции, позволяющего эффективно работать с широким классом сигналов частотной манипуляции в шумах высокой интенсивности.*

Метод демодуляции сигналов на основе их частотно-временных представлений

В соответствии с методологией, разработанной Коэном, спектрограмма, представляющая квадрат модуля кратковременного (оконного) преобразования Фурье, является простейшим частотно-временным представлением, хотя и не относится к классу билинейных распределений [1]. Поэтому рассмотрим возможность применения спектрограммы для демодуляции сигналов с частотной манипуляцией.

Пусть $z(t)$ — радиосигнал частотной манипуляции. Тогда его спектр $F(f)$ в комплексном базисе функций Фурье будет представлен следующим аналитическим выражением [7]:

$$F(f) = \int_{-\infty}^{\infty} z(t)\mathbf{e}^{-j2\pi ft}dt. \qquad (1)$$

Для получения совместного частотно-временного представления необходимо в выражение (1) ввести функцию, ограничивающую процесс вычисления по времени $h(t)$. Функцию $h(t)$ часто называют временным окном анализа. В результате получим частотно-временное распределение, представляющее оконное преобразование Фурье [1],

$$\rho_{\text{ОПФ}}(f, t) = \int_{-\infty}^{\infty} z(\tau)h(\tau - t)\mathbf{e}^{-j2\pi f\tau}d\tau. \qquad (2)$$

Тогда в соответствии с определением выражение для расчета спектрограммы сигнала $z(t)$ запишем следующим образом:

$$\rho_{\text{СП}}(f, t) = |\rho_{\text{ОПФ}}(f, t)|^2. \qquad (3)$$

На рис. 1 представлен спектр тестового радиосигнала $z(t)$, манипулированного меандром, и его спектрограмма.

Вертикальные штриховые линии на рис. 1, обозначенные как A и B, определяют границы, заключающие 98 % спектральной энергии сигнала.

Анализ спектрограммы на рис. 1, *б* показывает, что на частотно-временной матрице распределения сигнальные компоненты имеют вид функции первичного модулирующего сигнала. Следовательно, если выделить стро-

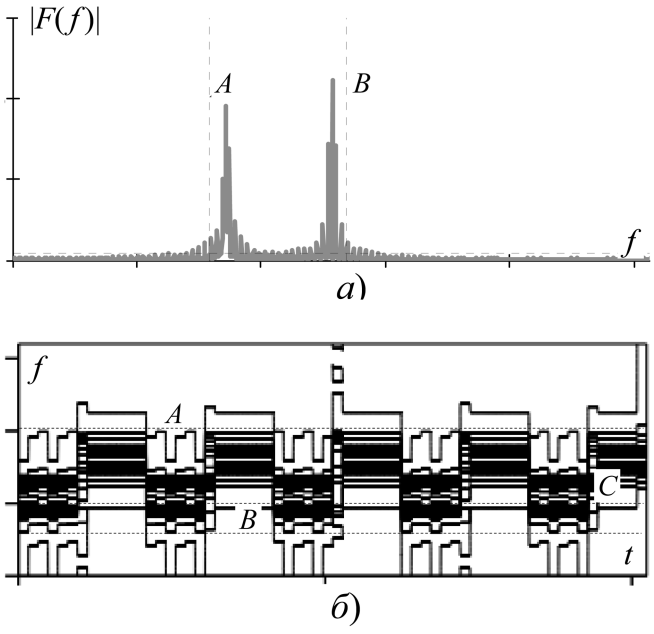


Рис. 1. Спектр тестового сигнала (*а*) и его спектрограмма (*б*)

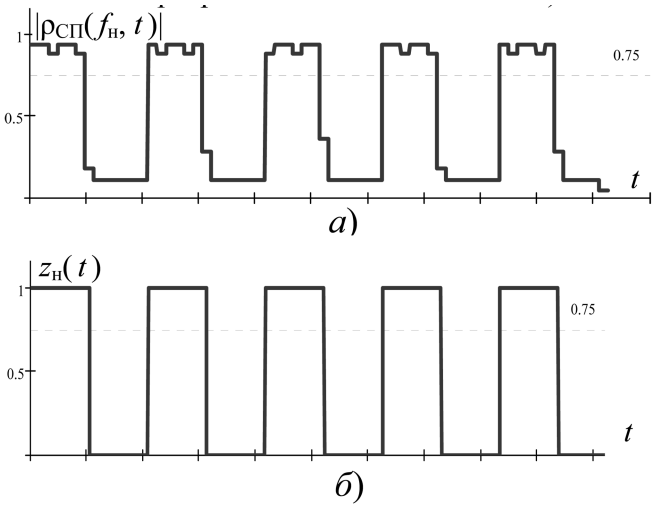


Рис. 2. Функция огибающей строки частотно-временной матрицы распределения тестового сигнала, соответствующей частоте нажатия (*а*) и первичный модулирующий сигнал на частоте нажатия (*б*)

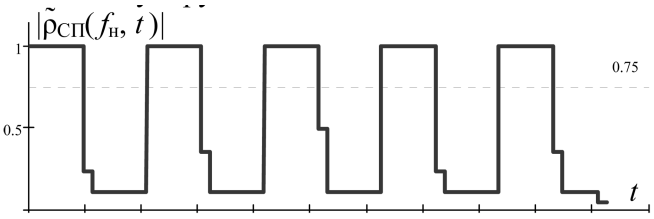


Рис. 3. Функция огибающей строки частотно-временной матрицы распределения тестового сигнала, соответствующей частоте нажатия (*а*), и первичный модулирующий сигнал на частоте нажатия (*б*)

ку матрицы, соответствующую максимальному уровню энергии, то по форме функции ее огибающей можно судить о модулирующем (первичном) сигнале. Так, на рис. 2, *а* представлена функция огибающей значений строки матрицы, соответствующей частоте нажатия $f_н$ (на рис. 1, *б* пунктирная линия *C*).

На рис. 2, совместно с функцией огибающей нанесена штриховая линия по уровню 0,75, представляющая порог принятия решения. Он позволяет нивелировать фронты функции огибающей, поскольку те не всегда могут быть достаточно гладкими. Более того, они, как правило, разнятся по уровню от одной битовой посылки к другой [6].

Анализ полученных результатов (рис. 2, *б*) позволяет сделать вывод о близости форм функции огибающей распределения пиковых значений энергии и первичного сигнала. Следовательно, выделенная из частотно-временной матрицы функция огибающая довольно полно характеризует модулирующий сигнал.

Таким образом, данные проведенных исследований позволяют определить первичную трактовку основных этапов, составляющих метод демодуляции частотно-манипулированных сигналов на основе обработки их частотно-временных представлений.

На первом этапе рассчитывается матрица частотно-временного представления демодулируемого сигнала.

На втором этапе определяются строки матрицы, в пределах которых локализуется энергия сигнальных компонент.

На третьем этапе определяется уровень порога принятия решения.

На четвертом собственно и осуществляется демодуляция как процедура сравнения значений функции огибающей со значением порога в каждый момент времени на основе дуального решения, т. е. есть пересечение или нет.

Учитывая, что функция огибающей тестового сигнала не является гладкой, можно предположить, что в условиях шумов значение неровностей увеличится. Указанные обстоятельства в значительной степени усложнят эффективную реализацию процедуры выбора порога принятия решения.

В целях повышения ее продуктивности необходимо модифицировать рассчитанную частотно-временную матрицу за счет ее временно́го нормирования. Это позволит компенсировать энергетические "провалы" на частотно-временной плоскости, вызванные не только аддитивными шумами, но и результатом мультипликативных замираний, например, обусловленных многолучевостью ионосферного распространения радиоволн, характерной для сигналов декаметрового диапазона.

На рис. 3 представлена функция огибающей тестового сигнала, полученной на основе его предварительно нормированной частотно-временной матрицы $\rho_{СП}(f, t)$.

Результаты экперимента по демодуляции частотно-манипулированных сигналов

Для оценки продуктивности предлагаемого метода были проведены исследования по демодуляции частотно-манипулированных сигналов в шумах различной интенсивности.

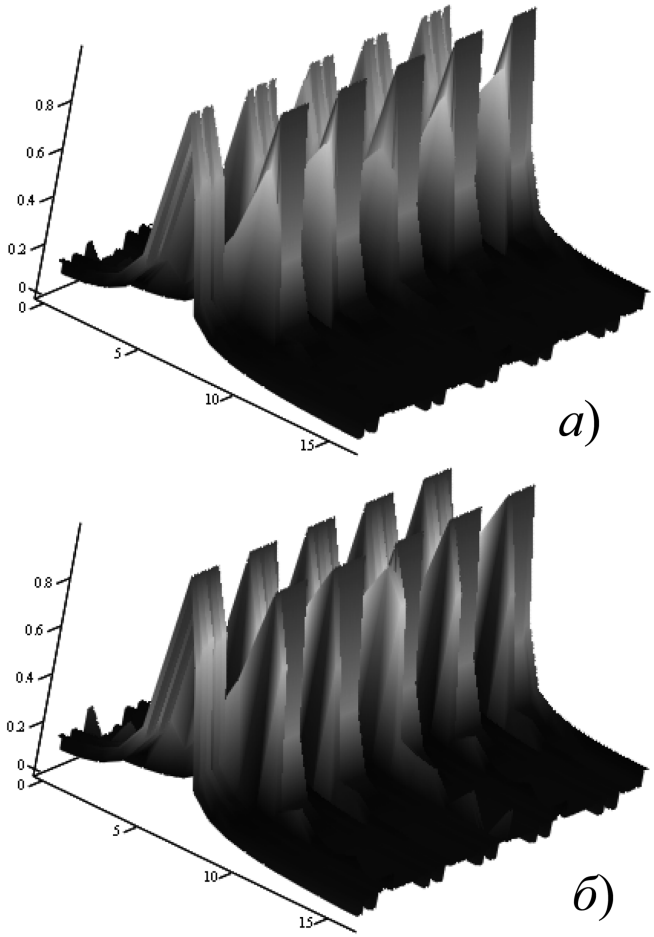


Рис. 4. Спектрограмма тестового сигнала без шумов:
а — классическая форма; *б* — модифицированная форма

В качестве тестового синтезировался радиосигнал длительностью в 2048 дискретных отсчета, модулированный меандром с первичным импульсом в 204 дискретных отчета, аналогичный изображенному на рис. 2, *б*.

При формировании спектрограммы в качестве функции-окна использовалась функция Хэмминга [7]

$$h(t) = 0{,}54 - 0{,}46\cos\left(2\pi\frac{t}{n-1}\right), \qquad (4)$$

где n — переменная, регулирующая длительность окна анализа, в пределах которого осуществляется обработка сигнала.

Эта же функция использовалась для синтеза фильтра, который согласовывался по спектру с модулирующим сигналом.

Затем с первичным сигналом $z_{\Pi}(t)$ сравнивались функции огибающих, выделенные из матрицы спектрограммы $\rho_{\text{ОПФ}}(f_{\text{н}}, t)$ и матрицы ее модифицированной формы $\rho^{\text{м}}_{\text{ОПФ}}(f_{\text{н}}, t)$ за счет процедур временного нормирования, а также результирующая функция $\rho_{\Phi}(t)$, полученная в результате свертки согласованного фильтра с тестовым сигналом

$$\rho_{\Phi}(t) = \int_{-\infty}^{\infty} h(\tau - t)z(\tau)d\tau. \qquad (5)$$

На рис. 4 представлены классическая и модифицированная формы спектрограммы тестового сигнала без шумов.

Качество демодуляции оценивалось по значению абсолютной ошибки между первичным сигналом и полученными функциями

$$\Delta = \frac{1}{N}\sum_{i=0}^{N} |z_{\Pi}(t) - \rho(t)|. \qquad (6)$$

Здесь для соответствующего значения ошибки $\Delta_{\text{ОПФ}}$, $\Delta^{\text{м}}_{\text{ОПФ}}$, Δ_{Φ} в качестве $\rho(t)$ выступали, соответственно, $\rho_{\text{ОПФ}}(f_{\text{н}}, t)$, $\rho^{\text{м}}_{\text{ОПФ}}(f_{\text{н}}, t)$ и $\rho_{\Phi}(t)$ (рис. 5).

Затем тестовый сигнал аддитивно смешивался с шумом и эксперимент повторялся. В ходе опытов отношение сигнал/шум (ОСШ) оценивалось как отношение средней мощности сигнала к спектральной плотности шума. Мощность сигнала рассматривалась в пределах пятых гармоник раздельно для каждой из частот нажатия и отжатия

$$\Delta F_0 = |(f_0 + f_0^5) - (f_0 - f_0^5)|,$$
$$\Delta F_{\text{н}} = |(f_{\text{н}} + f_{\text{н}}^5) - (f_{\text{н}} - f_{\text{н}}^5)|. \qquad (7)$$

На рис. 6 представлены матрицы классической спектрограммы и ее модифицированной формы тестового сигнала при ОСШ 10 дБ.

Следует отметить, что при указанных значениях ОСШ метод корреляционной фильтрации фактически не позволяет получать приемлемых результатов (рис. 7, *а*). В данном эксперименте амплитудно-частотную характеристику фильтра выбирали из соображения полного восстановления модулирующего колебания. Однако на практике фильтрующие системы настраивают только на первую гармонику, а для восстановления первичного сигнала дополнительно применяют корректирующие алгоритмы (рис. 8, *а*). В этом случае общая эффективность демодулятора существенно повышается.

Следует заметить, что аналогичные алгоритмы коррекции применимы и для предлагаемого метода, а так как в рамках эксперимента исследовалась эффективность фильтрации с позиций непосредственной демодуляции, то полученные результаты сравнивались с функцией $\rho_{\Phi}(t)$.

В табл. 1 представлены оценки значений ошибок $\Delta_{\text{ОПФ}}$, $\Delta^{\text{м}}_{\text{ОПФ}}$ и Δ_{Φ}, нормированных к максимальному значению.

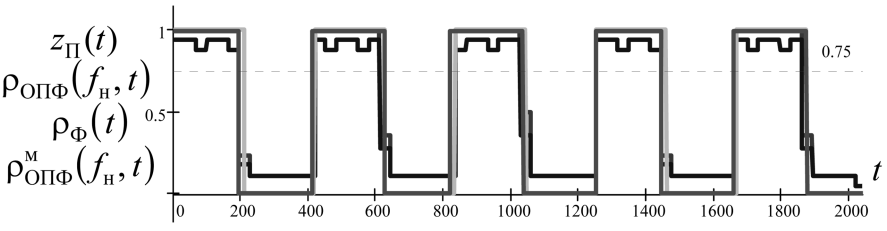


Рис. 5. Модулирующий сигнал, функция огибающей строки матрицы классической спектрограммы и ее модифицированной формы, результирующая функция корреляции

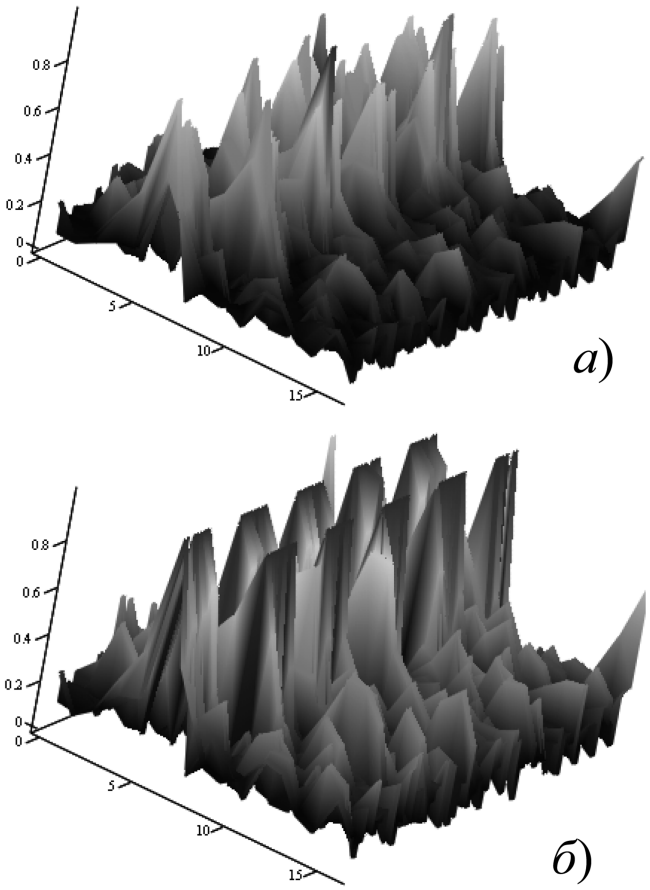


Рис. 6. Спектрограмма тестового сигнала при ОСШ 10 дБ: *а* — классическая форма; *б* — модифицированная форма

Результаты табл. 1 получены методом Монте-Карло по 100 выборкам, хотя рекомендуемое число выборок должно быть больше 200. Однако на практике часто ограничиваются меньшим числом, допуская при этом определенный проигрыш в точности вычисления статистических оценок [8].

В качестве другого показателя эффективности (табл. 2) рассматривалась относительная оценка, полученная как отношение функций $\Delta_{\text{ОПФ}}$, $\Delta^{\text{м}}_{\text{ОПФ}}$ и Δ_{Φ} к значению ошибки огибающей, выделенной из частотно-временной матрицы классической спектрограммы $\delta = \Delta/\Delta_{\text{ОПФ}}$ для каждого значения ОСШ.

Использование δ позволяет наглядно оценить пределы диапазона эффективного применения того или иного метода. В частности, при высоких значениях ОСШ применение демодуляции на основе корреляционной фильтрации видится более предпочтительным. В то время как при ОСШ ниже 18—15 дБ преимущество разработанного метода неоспоримо.

Анализ результатов, представленных на рис. 8, показывает, что даже изменение порога принятия решения в методе демодуляции, базирующемся на классической форме спектрограммы, не даст существенного улучшения.

Таким образом, основываясь на результатах проведенного эксперимента, целесообразно уточнить трактовку первого этапа разработанного метода. А именно, сформированную матрицу распределения модифицировать за счет применения к ней операций временного нормирования.

Распространение метода демодуляции на билинейные частотно-временные распределения

Для строгости перехода от спектрограмм к билинейным распределениям в предлагаемом методе целесообразно рассмотреть обобщенное распределение Коэна, свойства которого подробно исследованы в работе [9]:

$$\rho(f, t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp[j2\pi(\xi t - f\tau - \xi\nu)]\Phi(\tau, \xi)K(\nu, \tau)d\nu d\tau d\xi, \quad (8)$$

где $K(\nu, \tau) = z_a^*(\nu - \tau/2)z_a(\nu + \tau/2)$; $\Phi(\tau, \xi)$ — функциональное (порождающее) ядро преобразования, определяющее тип распределения.

Фундаментальность уравнения (8) определяется сохранением всех полезных свойств синтезируемых на его

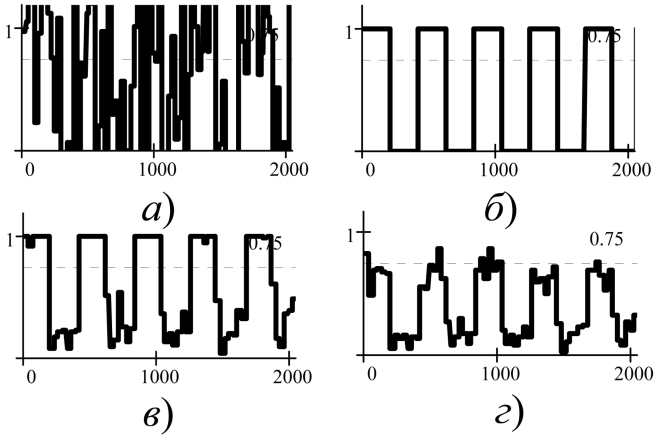


Рис. 7. Функция корреляции (*а*), модулирующий сигнал (*б*), функция огибающей строки матрицы модифицированной спектрограммы (*в*) и функция огибающей строки матрицы классической спектрограммы (*г*) при ОСШ 15 дБ

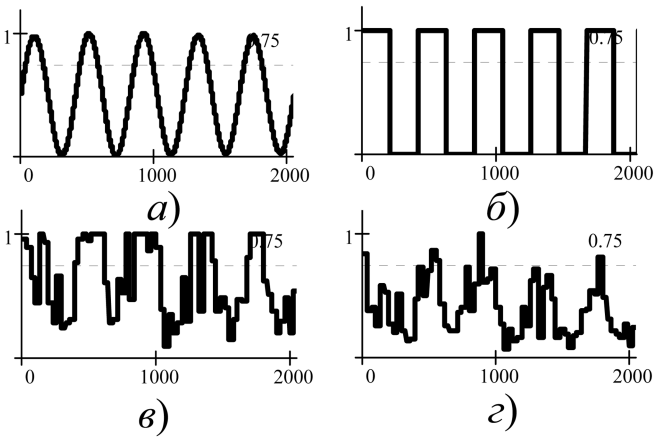


Рис. 8. Функция фильтра 1-й гармоники (*а*), модулирующий сигнал (*б*), функция огибающей строки матрицы модифицированной спектрограммы (*в*) и функция огибающей строки матрицы классической спектрограммы (*г*) при ОСШ 8 дБ

Таблица 1

Результаты ошибки демодуляции тестового сигнала

Значение ОСШ, дБ	Значение ошибки		
	Δ_{Φ}, %	$\Delta_{\text{ОПФ}}$, %	$\Delta^{\text{м}}_{\text{ОПФ}}$, %
0	2,37	7,72	5,34
34	7,53	8,69	5,41
30	12,33	9,17	5,47
25	17,13	9,84	5,59
22	20,90	10,09	5,72
20	24,00	11,30	6,07
19	27,64	12,70	6,80
18	34,99	13,55	7,29
16	38,82	14,34	7,72
15	40,93	14,82	8,63
14	48,91	15,31	8,99
13	61,30	15,86	10,09
12	66,16	16,40	10,45
11	70,47	17,31	11,00
10	76,67	18,65	14,22
9	85,24	20,60	14,40
8	100,0	21,63	16,46

Таблица 2

Результаты относительной оценки

Значение ОСШ, дБ	Значение ошибки для		
	δ_{Φ}	$\delta_{\text{ОПФ}}$	$\delta^{\text{м}}_{\text{ОПФ}}$
0	0,31	1,0	0,69
34	0,87	1,0	0,62
30	1,34	1,0	0,60
25	1,74	1,0	0,57
22	2,07	1,0	0,56
20	2,12	1,0	0,54
19	2,18	1,0	0,54
18	2,58	1,0	0,54
16	2,7	1,0	0,54
15	2,75	1,0	0,58
14	3,19	1,0	0,59
13	3,87	1,0	0,64
12	4,03	1,0	0,64
11	4,07	1,0	0,64
10	4,11	1,0	0,68
9	4,14	1,0	0,70
8	4,62	1,0	0,76

основе частотно-временных распределений по отношению к их классическим формам и согласованием их с теорией оценивания параметров сигналов.

Синтезу уравнения, определяющего обобщенную форму (8), предшествовал ряд работ Коэна, наиболее интересной из которых является работа [1]. В ней впервые определено понятие билинейности, сущность которой состоит в том, что при формировании распределений исходный сигнал используется дважды (в виде $K(\nu, \tau) = z_a^*(\nu - \tau/2)z_a(\nu + \tau/2)$), причем в формуле синтеза связь между описаниями сигнала линейна.

В целях исследования возможности применимости билинейных распределений, синтезированных на основе уравнения (8) в разработанном методе демодуляции, проанализируем, на сколько этому соответствуют получаемые частотные и временные оценки сигналов с использованием функции плотности распределения энергии.

Утверждение 1.

Результат интегрирования любого совместного распределения по частоте дает среднее значение квадрата огибающей при условии $\Phi(0, \xi) \equiv 1$.

Доказательство.

Проинтегрируем распределение сигнала $\rho(f, t)$ по частоте $\int_{-\infty}^{\infty} \rho(f, t)df$. Учитывая что $\int_{-\infty}^{\infty} \exp(-j2\pi f\tau)df = \delta(\tau)$, получим

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp[j2\pi\xi(t-\nu)]\Phi(0, \xi)\, z_a^*(\nu)z_a(\nu)d\nu d\xi =$$

$$= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} \exp[j2\pi\xi(t-\nu)]d\xi\right)|z_a(\nu)|^2 d\nu =$$

$$= \int_{-\infty}^{\infty} \delta(t-\nu)|z_a(\nu)|^2 d\nu = |z_a(t)|^2.$$

Утверждение 2.

Результат интегрирования любого совместного распределения по времени дает среднее значение энергии сигнала (квадрата его амплитудных значений) при условии $\Phi(\tau, 0) \equiv 1$.

Доказательство.

Проинтегрируем распределение сигнала $\rho(f, t)$ по времени $\int_{-\infty}^{\infty} \rho(f, t)dt$. Учитывая, что $\int_{-\infty}^{\infty} \exp(-j2\pi f\tau)d\tau = \delta(f)$, получим

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp[j2\pi(-\tau f)]\Phi(\tau, 0)\, z_a^*(\nu - \tau/2)z_a(\nu + \tau/2)d\nu d\tau =$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{e}^{-j2\pi\tau f}F_a^*(f)\mathbf{e}^{-j2\pi f(\nu-\tau/2)}F_a(f)\mathbf{e}^{-j2\pi f(\nu+\tau/2)}df d\nu d\tau =$$

$$= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp[j2\pi f(-\tau-2\nu)]d\tau d\nu\right)|F_a(f)|^2 df =$$

$$= \int_{-\infty}^{\infty} \delta(f)|F_a(f)|^2 df = |F_a(f)|^2.$$

Представленные аналитические доказательства подтвердили правомерность выбора билинейных распределений в качестве базовых.

Поскольку ядро обобщенного преобразования в уравнении (8) не зависит от f и t, то соответственно, любые сдвиги сигнала по времени или по частоте приводят к аналогичным частотно-временным сдвигам в их билинейных распределениях [9]. Указанное фундаментальное свойство делает его тонким инструментом анализа. Если у билинейных распределений частотное разрешение зависит от частоты дискретизации, то у спектрограмм оно определяется параметрами окна, в пределах которого происходит усреднение энергии. Это делает спектрограммы в большей степени размытыми частотно-временными описаниями по отношению к билинейным

представлениям. Следовательно, функция огибающей частотно-временной матрицы, полученная на основе билинейных распределений, будет в большей степени рельефна, что, в конечном счете, повысит эффективность демодуляции.

Заключение

Предложенный в работе метод демодуляции естественно в большей степени применим при решении задач мониторинга, когда сложно использовать методы оптимальной фильтрации. Между тем результаты практического эксперимента указывают на его высокую продуктивность в условиях шумов высокой интенсивности. Более того, можно предположить, что применение степенных преобразований к частотно-временным матрицам распределений [6] позволит повысить эффективность метода. Другой путь повышения качества демодуляции видится в использовании масштабно-временных распределений, поскольку те позволяют концентрировать шумы преимущественно в высокочастотной части матриц, которые затем легко убираются в результате обратной репродукции только непораженных фрагментов.

Результаты работы могут рассматриваться в качестве основы для разработки алгоритмов демодуляции частотно-манипулированных радиосигналов.

Список литературы

1. **Cohen L.** Generalized phase-space distribution function // J. of Mathematical Physics. 1966. Vol. 7. N 5. P. 781—786.

2. **Claasen T. A. C. M., Meclenbrauker W. F. G.** The Wigner distribution a tool for time-frequency signal analysis. Part 1, 2, 3 // Philips J. Res. 1980. Vol. 35. P. 217—250, 276—300, 372—389.

3. **Алексеев А. А., Дворников С. В., Железняк В. К.** и др. Применение методов частотно-временной обработки акустических сигналов для анализа параметров реверберации // Научное приборостроение. 2001. Т. 11. № 1. С. 65—76.

4. **Дворников С. В., Алексеева Т. Е.** Распределение Алексеева и его применение в задачах частотно-временной обработки сигналов // Информация и космос. 2006. № 3. С. 9—21.

5. **Дворников С. В., Комарович В. Ф., Железняк В. К.** и др. Метод обнаружения радиосигналов на основе обработки их частотно-временных распределений плотности энергии // Информация и космос. 2005. № 4. С. 13—17.

6. **Дворников С. В., Сауков А. М.** Модификация частотно-временных описаний нестационарных процессов на основе показательных и степенных функций // Научное приборостроение. Т. 14. 2004. № 2. С. 57—66.

7. **Гоноровский И. С.** Радиотехнические цепи и сигналы: Учебник для вузов. 4-е изд., перераб. и доп. М.: Радио и связь, 1986. 512 с.

8. **Математический** энциклопедический словарь. М.: Сов. Энциклопедия, 1988. 847 с.

9. **Коэн Л.** Времячастотные распределения: Обзор // ТИИЭР. 1989. Т. 77. № 10. С. 72—121.

ОБМЕН ОПЫТОМ

УДК 004.942:005.311.6:025.21

О. В. Федорец, ст. науч. сотр.,
Всероссийский институт научной и технической информации
Российской академии наук (ВИНИТИ РАН), e-mail: ovf@viniti.ru

Статистический подход к определению приоритета критериев для рейтингового оценивания научных журналов методом анализа иерархий

Рассматривается задача автоматизации рейтингового оценивания научных журналов. Ключевой проблемой многокритериального оценивания является обоснованный отбор частных критериев и определение их приоритета. Предлагается статистическая методика отбора и взвешивания критериев, основанная на использовании эталонного критерия и обучающей выборки. Фрагменты результатов эксперимента представлены в виде ранжированных списков российских журналов, тематика которых близка к информационным технологиям.

Ключевые слова: *рейтинг научных журналов, принятие решений, метод анализа иерархий, приоритет критериев, статистическая методика, обучающая выборка, база данных, язык SQL.*

Введение

Наиболее традиционным результатом многокритериального оценивания в различных областях деятельности является числовой показатель, именуемый *рейтингом*. Рейтинги используются для ранжирования банков, корпораций, ценных бумаг, университетов, спортсменов, политических деятелей, стран и т. д. При этом одному и тому же объекту может быть присвоено несколько рейтингов, характеризующих его с различных сторон.

Научный журнал, как один из важнейших компонентов системы научных коммуникаций, также может являться объектом для комплексного рейтингового оце-

нивания, по результатам которого может приниматься решение о включении журнала во входной поток информационного центра, библиотеки, реферативной службы.

Исходные данные и постановка задачи

В [1] представлен краткий обзор публикаций, посвященных комплексной оценке сериальных изданий по множеству критериев. В основу разработанной методики оценивания положен метод анализа иерархий (МАИ) [2], который находит практическое применение в различных сферах, в том числе рейтинговом оценивании [3] и оптимизации распределения ресурсов [4]. Описание МАИ и примеры его применения в экономической сфере можно также найти в [5].

Наиболее объективным показателем влияния, которое журнал оказывает на научное сообщество, является импакт-фактор, вычисляемый на основе частоты цитирования журнальных статей и публикуемый в указателе "*Journal Citation Reports*" (*JCR*), который издается в США фирмой *Thomson Scientific*.

В связи с этим правомерно использовать импакт-фактор в качестве эталонного критерия для определения весомости остальных критериев оценки сериальных изданий на основе обучающей выборки. Несмотря на изменчивость показателя цитирования во времени на больших статистических выборках можно получать вполне достоверные результаты.

Конечной целью рейтингового оценивания сериального издания (СИ) является получение векторной оценки в виде двух величин: рейтинга качества $R_{\text{кач}}$ и рейтинга спроса $R_{\text{спр}}$.

Общий рейтинг СИ было решено вычислять умножением рейтинга спроса на рейтинг качества:

$$R_{\text{общ}} = R_{\text{спр}} R_{\text{кач}}. \qquad (1)$$

Критерии оценки качества научного журнала были разбиты на три группы:

- экспертные критерии оценки;
- формальные критерии оценки;
- оценка цитирования.

Оценка сравнительной важности критериев

В связи с тем, что для отбора и взвешивания критерия выполняются различные действия над статистическими выборками, сами выборки и действия над ними удобно определять с помощью аппарата теории множеств.

Вначале определим множества J, E и K:

- J — множество индексов. Каждый индекс представляет собой целое положительное число, идентифицирующее объект. В нашем случае объектом является научный журнал;
- E — множество допустимых значений эталонного критерия. Допустимым значением будем считать неотрицательное рациональное число. Для научных журналов в качестве эталонного критерия был выбран индекс цитирования — "импакт-фактор";
- K — множество допустимых значений исследуемого частного критерия: $K = E$.

В данной модели возрастание значения эталонного критерия означает возрастание предпочтительности объекта или повышение его качества. Выдвигается гипотеза, что более предпочтительному (качественному) объекту соответствует большее значение частного критерия. Цель исследования критерия — подтверждение или опровержение этой гипотезы, а также определение приоритета критерия по отношению к другим критериям в случае подтверждения гипотезы. Если гипотеза принимается, то частный критерий включается в иерархию критериев, на основе которой строится обобщенный критерий — рейтинг объекта.

Элементы множества R являются тернарными кортежами, которые мы будем обозначать тройками вида (a, b, c). Строчные латинские буквы внутри круглых скобок — переменные величины, соответствующие элементам кортежа. Например, запись $(x, y, z) \in R$ означает, что $x \in J$, $y \in E$, $z \in K$.

Нам необходимо, чтобы первый элемент тернарного кортежа являлся уникальным идентификатором объекта, поэтому для множества R должно выполняться следующее условие:

$$\forall_{(j, e, k) \in R} \forall_{(i, x, y) \in R} (j, e, k) \neq (i, x, y) \Rightarrow (j \neq i).$$

Приведем пример, поясняющий данное условие.

Пусть отношение R — множество, состоящее из четырех кортежей: (1, 10, 20); (2, 30, 40); (3, 20, 50); (3, 40, 30).

Очевидно, что условие для R не выполняется, так как идентификатор 3 дублируется:

$$(3, 20, 50) \neq (3, 40, 30) \Rightarrow (3 \neq 3).$$

Следовательно, отношение R является недопустимым. Чтобы условие выполнялось для отношения R, нужно удалить из него последний или предпоследний кортеж, или оба этих кортежа.

Рассмотрим два случая: для булева и количественного значения исследуемого критерия k соответственно.

В первом случае исследуемый критерий k является булевой переменной.

Определим тернарные отношения R_0 и R_1, которые назовем соответственно "нулевой" и "единичной" выборкой:

$$K = \{0, 1\} \Rightarrow R_0 = \{(j, e, k) \mid (j, e, k) \in R \,\&\, k = 0\};$$

$$K = \{0, 1\} \Rightarrow R_1 = \{(j, e, k) \mid (j, e, k) \in R \,\&\, k = 1\}.$$

Во втором случае исследуемый критерий k является неотрицательной количественной переменной. В этом случае отношения R_0 и R_1 определяются по-другому:

$$K = \{k \mid k \geqslant 0\} \Rightarrow$$
$$\Rightarrow R_0 = \{(j, e, k) \mid (j, e, k) \in R \,\&\, k < \xi_{0,5}(K_R)\},$$
$$K = \{k \mid k \geqslant 0\} \Rightarrow$$
$$\Rightarrow R_1 = \{(j, e, k) \mid (j, e, k) \in R \,\&\, k \geqslant \xi_{0,5}(K_R)\},$$

где K_R — случайная величина, принимающая значение критерия k в упорядоченных парах $(j, e, k) \in R$, $\xi_{0,5}(K_E)$ — квантиль порядка 0,5 случайной величины, кратко именуемый медианой.

Введем дополнительные случайные величины. Обозначим E_0 случайную величину, принимающую значения критерия e в упорядоченных парах $(j, e, k) \in R_0$, т. е. в нулевой выборке. Аналогичным образом обозначим E_1 случайную величину, принимающую значения критерия e

в упорядоченных парах $(j, e, k) \in R_1$, т. е. в единичной выборке. Тогда $\bar{E}_0$ и $\bar{E}_1$ — средние арифметические значения случайных величин E_0 и E_1 соответственно, т. е. выборочные средние.

Величина E^* показывает соотношение выборочных средних эталонного критерия в единичной и нулевой выборке:

$$E^* = \frac{\bar{E}_1}{\bar{E}_0}. \qquad (2)$$

Чем слабее связь исследуемого критерия с эталонным критерием, тем ближе значение E^* к единице. Далее в следующем параграфе описана проверка статистической гипотезы $\bar{E}_1 = \bar{E}_0$.

Зададим отношения S_0 и S_1 следующим образом:

$$S_0 = \{(j, e, k) | (j, e, k) \in R \;\&\; e < \xi_{0,5}(E_R)\};$$

$$S_1 = \{(j, e, k) | (j, e, k) \in R \;\&\; e \geqslant \xi_{0,5}(E_R)\},$$

где E_R — случайная величина, принимающая значение критерия e в упорядоченных парах $(j, e, k) \in R$.

Для булева частного критерия "успехом" назовем событие $k = 1$.

Для количественного частного критерия "успехом" назовем событие $k \geqslant \xi_{0,5}(K_R)$.

Тогда количество успехов в выборке "эталонный критерий < медианы" равно мощности множества $|S_0 \cap R_1|$, а число успехов в выборке "эталонный критерий ⩾ медианы" равно мощности множества $|S_1 \cap R_1|$.

Величина S^* показывает отношение числа успехов в выборках S_1 и S_0:

$$S^* = \frac{|S_1 \cap R_1|}{|S_0 \cap R_1|}. \qquad (3)$$

Чем слабее связь исследуемого критерия с эталонным критерием, тем ближе значение S^* к единице. В следующем параграфе описана проверка статистической гипотезы $|S_0 \cap R_1| = |S_1 \cap R_1|$.

Числовое значение приоритета исследуемого критерия вычисляется как среднее геометрическое величин E^* и S^*, определяемых формулами (2) и (3) соответственно:

$$w = \sqrt{E^* S^*} = \sqrt{\frac{\bar{E}_1 |S_1 \cap R_1|}{\bar{E}_0 |S_0 \cap R_1|}}. \qquad (4)$$

Ранжирование критериев по убыванию величины w равносильно ранжированию критериев по убыванию приоритета.

Отбор критериев по результатам проверки статистических гипотез

В основе разработанного метода статистического взвешивания частных критериев лежат выражения $\bar{E}_1 / \bar{E}_0$ и $|S_1 \cap R_1| / |S_0 \cap R_1|$, которые используются в формуле (4). Проверяется статистическая гипотеза *"более предпочтительному (качественному) объекту соответствует большее значение исследуемого критерия*". Если эта гипотеза принимается, то критерий считается значимым для оценки объекта.

Предлагается для проверки гипотезы использовать два статистических теста. Если в обоих случаях подтверждается значимость частного критерия, то он используется для вычисления интегрального показателя (рейтинга) объекта.

Подробное описание обоих тестов можно найти в [6], где они названы следующим образом:

- сравнение двух средних произвольно распределенных генеральных совокупностей (большие независимые выборки);
- сравнение двух вероятностей биномиальных распределений.

Первый тест используется для проверки гипотезы $\bar{E}_1 = \bar{E}_0$, т. е. гипотезы о равенстве средних значений эталонного критерия (импакт-фактора) в нулевой и единичной выборках при конкурирующей гипотезе $\bar{E}_1 > \bar{E}_0$.

Второй тест используется для проверки гипотезы $|S_1 \cap R_1| : |S_1| = |S_0 \cap R_1| : |S_0|$, т. е. гипотезы о равенстве относительных частот успеха в выборках S_1 и S_0, при конкурирующей гипотезе $|S_1 \cap R_1| : |S_1| > |S_0 \cap R_1| : |S_0|$.

Если оба теста отвергают гипотезу о равенстве, критерий считается значимым для оценки объекта. В табл. 1 приведены реальные исходные данные по шести критериям для проверки статистических гипотез. По результатам проверки статистических гипотез все указанные критерии были признаны значимыми.

Критерий "Направляется академикам" выделяется из общего ряда, так как на самом деле он является не формальным, а экспертным показателем. Список журналов, оглавления которых направляются академикам РАН по их личному заказу, регулярно актуализируется. Таким образом, статистическая методика отбора и взвешивания критериев была успешно про-

Таблица 1

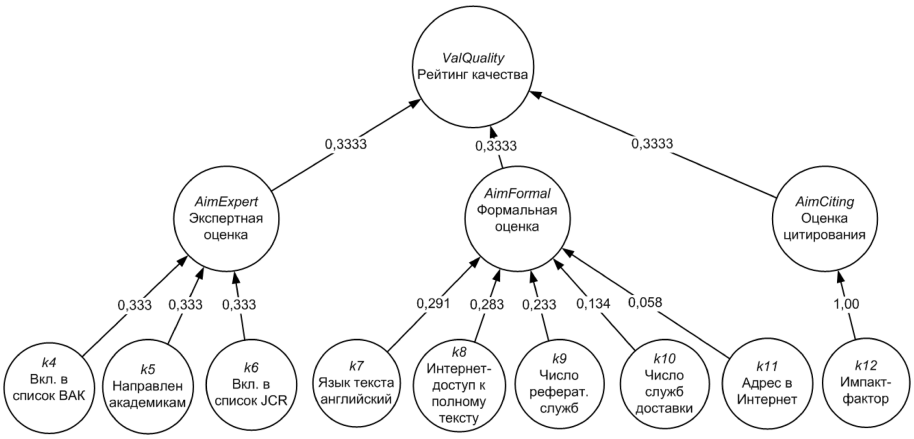
Исходные данные для проверки статистических гипотез о значимости критериев оценки

Название частного критерия	Нулевая выборка			Единичная выборка			Число успехов: $s_0 = \|S_0 \cap R_1\|$, $s_1 = \|S_1 \cap R_1\|$		Вес
	$\|R_0\|$	$\overline{E_0}$	$D(E_0)$	$\|R_1\|$	$\overline{E_1}$	$D(E_1)$	s_0	s_1	w
Язык текста английский (да, нет)	247	0,742	0,847	4090	1,843	2,648	1960	2130	1,643
Есть адрес в Интернет (да, нет)	735	1,423	2,965	3602	1,853	2,503	1675	1927	1,224
Есть Интернет-доступ к полному тексту (да, нет)	1581	1,225	2,177	2756	2,098	2,753	1082	1674	1,628
Направляется академикам (да, нет)	3047	1,588	2,309	1290	2,232	3,116	506	784	1,476
Число реферативных служб ⩾ 23 (да, нет)	2145	1,417	2,379	2192	2,135	2,739	853	1339	1,538
Число служб доставки ⩾ 6 (да, нет)	2251	1,505	2,524	2086	2,077	2,633	1220	1636	1,360
Примечание: $\xi_{0,5}(E_R) = 1{,}101$, $\|S_0\| = \|S_1\| = 2168$.									

верена на показателе, значимость которого изначально не вызывала сомнений.

Шкалы для оценки журналов по критериям

Значения веса, полученные ранее по формуле (4) и приведенные в табл. 1, позволяют получить ранжированный по важности список критериев, но являются ненормированными величинами. Их нельзя использовать в качестве весовых коэффициентов при вычислении интегрального показателя, так как отношение максимального значения к минимальному слишком мало.



Иерархия оценки качества научного журнала

Универсальный подход к определению весомости критериев в зависимости от предметной области — построение вербально-числовой шкалы для установления соответствия между "относительным приростом" и "относительной важностью". За основу для построения такой шкалы можно взять 9-балльную вербально-числовую шкалу отношений, которая наиболее часто используется в методе анализа иерархий при парном сравнении критериев и объектов относительно критерия [2, 5].

На рисунке представлена иерархия критериев оценки качества СИ в виде ориентированного графа древовидной структуры. Значение веса критерия указано на дуге графа, исходящей из вершины, соответствующей критерию.

Точно так же на исходящих дугах указаны значения веса для подцелей, т. е. для трех вершин, соответствующих трем видам оценки: экспертной, формальной и цитирования.

Вес формальных критериев, вносящих вклад в формальную оценку, определен статистически. Экспертным критериям и подцелям присвоен одинаковый вес в связи с тем, что на данном этапе работы привлечение экспертов для сравнения критериев и подцелей не планировалось.

В данной работе предлагается упрощенный способ установления соответствия между приростом и уровнем значимости прироста некоторого признака. Экспертам достаточно определить значимость превосходства максимального значения прироста над минимальным значением прироста по вербально-числовой шкале отношений.

Это позволяет вычислять значимость прироста автоматически для любых других значений w, лежащих в интервале $[\min(W); \max(W)]$, по формуле

$$w^{(1\ldots s)} = \left(\frac{w - \min(W)}{\max(W) - \min(W)} (s - 1) \right) + 1, \qquad (5)$$

где s — предпочтительность наиболее приоритетного критерия по сравнению с наименее приоритетным критерием по шкале отношений; w — ненормированный вес критерия.

Табл. 2 иллюстрирует получение нормированного вектора приоритета (в последней колонке) с помощью формулы (5) при $s = 5$, что соответствует "существенной или сильной значимости" по шкале отношений. Последняя колонка табл. 2 представляет собой нормированный вектор-столбец приоритета критериев, т. е. $\sum_{i=1}^{n} \hat{w}_i = 1$.

Рейтинг качества и рейтинг спроса

Характеристики издания, которые не относятся к показателям спроса, но могут оказывать влияние на оценку и отбор издания, можно отнести к качественным свойствам издания.

Согласно иерархии, изображенной на рисунке, оценка качества сериального издания вычисляется следующим образом:

$$ValQuality = AimExpert \cdot 0{,}333 + AimFormal \cdot 0{,}333 + AimCiting \cdot 0{,}333,$$

где значения трех подцелей второго уровня вычисляются по формулам:

$$AimExpert = k_4 \cdot 0{,}333 + k_5 \cdot 0{,}333 + k_6 \cdot 0{,}333,$$

$$AimFormal = k_7 \cdot 0{,}291 + k_8 \cdot 0{,}283 + k_9 \cdot 0{,}233 + k_{10} \cdot 0{,}134 + k_{11} \cdot 0{,}058,$$

$$AimCiting = k_{12} \cdot 1{,}00.$$

В отличие от критериев качества журнала критерии спроса — это в основном количественные переменные целого типа, показывающие частоту использования журнальных статей по различным каналам информационного обслуживания. В иерархии спроса все значения критериев имеют количественный тип и неопределенные значения отсутствуют. Следовательно, для вычисления рейтинга спроса можно использовать метод анализа иерархий без каких-либо изменений.

Таблица 2

Статистический приоритет критериев

Критерий	i	w_i	$w_i^{(1\ldots5)}$	$\hat{w}_i$
Язык текста английский	1	1,643	5,000	0,291
Интернет-доступ к полному тексту	2	1,628	4,857	0,283
Число реферативных служб	3	1,538	3,998	0,233
Число служб доставки	4	1,360	2,298	0,134
Наличие адреса издания в Интернет	5	1,224	1,000	0,058

Таблица 3

Тематический профиль журнала "Информационные технологии"

Код РЖ	Название РЖ	Рефераты	№ в ранж. списке
01В	Автоматика и вычислительная техника. Программное обеспечение: выпуск сводного тома	145	**2**
81	Техническая кибернетика: отдельный выпуск	131	**6**
01Б	Автоматика и вычислительная техника. Вычислительные машины и системы: выпуск сводного тома	66	**12**
59	Информатика: отдельный выпуск	51	**14**
01А	Автоматика и вычислительная техника. Автоматика и телемеханика: выпуск сводного тома	46	28
29А	Связь. Сети и системы связи: выпуск сводного тома	40	21
67Б	Организация управления. Экономические аспекты организации и техники систем управления: выпуск сводного тома	12	22
24В	Радиотехника. Радиолокация. Радионавигация. Радиоуправление. Телевизионная техника: выпуск сводного тома	11	78

Таблица 4

TOP-10 тематических ранжированных списков журналов

№	ISSN	Название
РЖ01В. Автоматика и вычислительная техника. Программное обеспечение: выпуск сводного тома		
1	1560-4640	САПР и графика
2	**1684-6400**	**Информационные технологии**
3	0132-3474	Программирование
4	0868-6157	Компьютер Пресс
5	1028-7493	Открытые системы
6		Известия Таганрогского государственного радиотехнического университета (ТРТУ)
7		BYTE/Россия
8	1684-2588	Телекоммуникации
9	0321-2653	Известия высших учебных заведений (вузов). Северо-Кавказский регион. Технические науки
10	0021-3454	Известия высших учебных заведений (вузов). Приборостроение
РЖ 81. Техническая кибернетика: отдельный выпуск		
1	0005-2310	Автоматика и телемеханика
2	0002-3388	Известия Российской академии наук (РАН). Теория и сстемы управления
3	0869-5350	Нейрокомпьютеры: разработка, применение
4	0321-2653	Известия высших учебных заведений (вузов). Северо-Кавказский регион. Технические науки
5	0021-3454	Известия высших учебных заведений (вузов). Приборостроение
6	**1684-6400**	**Информационные технологии**
7	1561-1531	Приборы и системы: Управление, контроль, диагностика
8		Известия Таганрогского государственного радиотехнического университета (ТРТУ)
9	0869-4931	Автоматизация и современные технологии
10	1684-6427	Мехатроника, автоматизация, управление

Фрагмент результатов экспериментов

Разработанная математическая модель вычисления рейтинга была опробована на реальных данных в 2007 г. В качестве выборки использовалось множество научных журналов, приходивших в ВИНИТИ РАН в 2001—2005 гг. и отражавшихся в реферативных журналах в 2001—2006 гг. В качестве эталонного критерия использовался импакт-фактор, опубликованный в *Journal Citation Report* в 2005 г.

Эксперименты проводились с использованием реляционной базы данных "Автоматизированной системы регистрации и комплектования ВИНИТИ РАН", работающей под управлением СУБД *Microsoft SQL Server 2000.* Все необходимые вычисления были реализованы с помощью процедур и функций, разработанных на языке *Transact-SQL.*

Рассмотрим пример ранжирования российских периодических изданий, близких по тематике журналу "Информационные технологии" (ISSN 1684-6400).

В течение 2001—2006 гг. около 2 тыс. российских периодических изданий регулярно приходили в ВИНИТИ РАН и отражались в 248 реферативных журналах (РЖ). Рефераты статей исследуемого журнала были рассеяны по 36 РЖ, однако из них можно выделить 8, в которых концентрировались 84,4 % рефератов (табл. 3). В четырех тематических списках, ранжированных по общему рейтингу, исследуемый журнал оказался в первой двадцатке (см. последнюю колонку табл. 3).

В табл. 4 представлены тематические ранжированные списки TOP-10, в которых журнал "Информационные технологии" оказался в первой десятке.

Заключение

Методика статистического взвешивания критериев разработана с учетом необходимости ее использования в условиях высокой размерности исходных данных, неполноты данных и различных типов значений (булевых и количественных).

Разработанная методика обладает следующими достоинствами:

- независимость от функций распределения критериев при использовании репрезентативной обучающей выборки;
- возможность взвешивать критерии на неполных массивах информации, когда для отдельных объектов значения некоторых критериев неизвестны.

Список литературы

1. **Федорец О. В.** Рейтинговая система оценивания научных журналов: математическая модель и результаты экспериментов. М., 2007. 75 с. Деп. в ВИНИТИ 28.12.07, № 1257-В2007.
2. **Саати Т.** Принятие решений. Метод анализа иерархий: Пер. с англ. М.: Радио и связь, 1989. 316 с.
3. **Николаева М. А., Юнцевич О. Ф.** Методы и алгоритмы построения рейтингов // Информационные технологии. 2003. № 12. С. 7—18.
4. **Федоров Ю. В.** Решение многокритериальной задачи оптимизации в нечеткой постановке // Информационные технологии. 2005. № 7. С. 55—60.
5. **Андрейчиков А. В., Андрейчикова О. Н.** Анализ, синтез, планирование решений в экономике: Учебник. 2-е изд., доп. и перераб. М.: Финансы и статистика, 2004. 464 с.
6. **Гмурман В. Е.** Теория вероятностей и математическая статистика: Учеб. пособие для вузов. 6-е изд., стер. М.: Высшая школа, 1998. Гл. 19. С. 281—348.

CONTENTS

Адрес редакции:
107076, Москва, Стромынский пер., 4
Телефон редакции журнала **(499) 269-5510**
E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *О. А. Ефремова*.
Корректор *Е. В. Комиссарова*
Сдано в набор 09.02.2009. Подписано в печать 16.03.2009. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л.10,78. Уч.-изд. л. 12,64. Заказ 276. 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