

ОБ ОДНОМ МЕТОДЕ ПОСТРОЕНИЯ МОДЕЛЕЙ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМ

А. Мамян

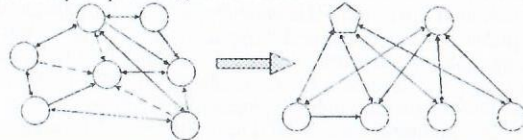
Для решения задачи прогноза состояния объектов социально-экономического характера в современных экспертных системах широко используется принцип имитационного моделирования [1]. В известных системах такого класса [2] эффективность решения поставленной задачи достигается применением микросимуляции агентов и их взаимоотношений, входящих в объект исследования. По причине того, что число взаимоотношений квадратично растет от мощности множества агентов, а также учитывая необходимость учета воздействия третьих агентов на парные взаимоотношения, требуются большие вычислительные ресурсы. В данной работе предлагается методы структурного преобразования модели объектов СЭС, приводящие ее к форме, эффективной для реализации современными инструментальными средствами моделирования. В числе последних можно отметить язык UML [3].

Рассмотрим отдельные аспекты предлагаемых методов структурного преобразования.

Введем классификацию экзогенных и эндогенных переменных агентов модели СЭС. В первую группу включим параметры товарно-денежных отношений. Связи, входящие в эту группу, условно назовем "рыночными". В следующем подмножестве структурируем параметры "социально-управленческих" отношений, которые представляют собой отношения с так называемыми властными структурами.

Согласно предлагаемому методу, в модель СЭС необходимо для групп с интенсивными парными взаимоотношениями ввести новые виртуальные звенья, в задачу которых будет входить имитация обеспечения баланса отношений между подчиненными ему узлами, что позволит основную часть парных взаимоотношений заменить на радиальные, проходящие через вышестоящий виртуальный узел. Обоснованием данного предложения служит однотипность большинства связей в СЭС: они имеют экономический "рыночный" характер [4]. Введение виртуального балансирующего узла вынуждает для подчиненных узлов определить новый список характеризующих их параметров $R\{j\}$, передача которых вышестоящему позволит заменить парные взаимоотношения на отношения один ко многим.

Описанный процесс реструктуризации модели заканчивается внесением корневого виртуального узла, условно называемого "рынком", с охватом всех "рыночных" взаимоотношений предметной области. В функцию модели "рынка" входит разрешение дисбаланса между спросом и предложением, возникшего по причине автономной работы агентов. Искомыми элементами множества $R\{j\}$ служат подаваемые на вход "рынка" аккумулярованные спрос и предложение моделей агентов. Выходом же модели являются установившиеся в результате торгов цены и объем продажи товаров. Торги для каждого товара производятся автономно.



Второй метод реструктуризации заключается в введении новых виртуальных узлов, которые представляют агрегацию группы агентов. Параметры нового узла можно разделить на две группы:

- $X_1\{x_{1j}\}, X_2\{x_{2j}\}$ - параметры, соответствующие внутренним параметрам агентов:

$$x_{1i} = \frac{\sum_{j=1}^k x_{1j}}{k} \quad (1), \text{ где } k - \text{число агрегируемых агентов,}$$

$$x_{2i} = \frac{\sum_{j=1}^k x_{2j} - x_{1i}^2}{k}; \quad (2)$$

- $Y_1\{y_{1j}\}, Y_2\{y_{2j}\}, Y_3\{y_{3j}\}$ - параметры, с помощью которых новый виртуальный групповой узел взаимодействует с остальными узлами модели СЭС. $Y_1\{y_{1j}\}, Y_2\{y_{2j}\}$ представляют собой характеристики типового члена агрегации и рассчитываются аналогично формулам (1) и (2) соответственно. $Y_3\{y_{3j}\}$ представляет собой куммулятивный эффект

воздействия агрегации агентов на остальные компоненты модели СЭС: $y_{3i} = \sum_{j=1}^k y_{3j}$.

Для принятия решения о включении отдельного предприятия в агрегацию вводится коэффициент значимости (влияния) K_z , который определяется исходя из размера, стратегической значимости, уникальности производимой продукции:

$$K_z = \sum_{i=1, k} \alpha_i \left(\frac{\sum_{j=1, l} \beta_j \delta_{ij}}{\delta_{zi}} \right), \text{ где } \alpha_i, \beta_j - \text{весовые коэффициенты; } \delta_{zi} - \text{вне-$$

сенная вариация i -й переменной состояния объекта при численном эксперименте; δ_{zi} - зарегистрированное изменение j -й характеристики модели СЭС.

Таким образом, предложенные методы структурного преобразования дают возможность уменьшить количество узлов в модели СЭС за счет агрегации и уменьшить число моделируемых отношений путем введения виртуальных балансирующих узлов. Реструктурированная модель также обеспечивает простое внесение новых агентов, так как последние связываются с имеющимися компонентами в основном посредством радиальных отношений через виртуальный балансирующий узел. Предложенная методика была апробирована в экспертной системе прогноза состояния СЭС ISMA, разработанной по гранту фонда Евразия.

Литература

1. Hanneman R. and Patrick S. On the Uses of Computer Assisted Simulation Modeling in the Social Sciences. Microsimulation and Public Policy, Edited by Ann Harding
2. Буч Г., Рамбо А., Джекобсон А. Язык UML. Руководство пользователя.
3. Экономическая теория /Под ред. А. Добрынина.

ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ УПРАВЛЕНИЯ XML ДАННЫМИ

А. Палян, А. Палян

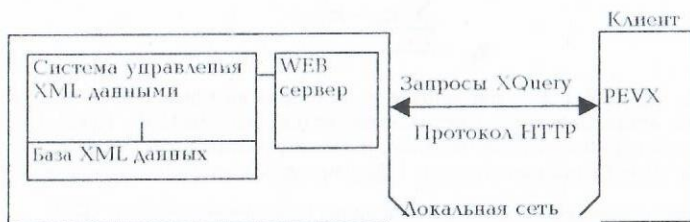
В современных информационных технологиях, в частности в Internet и в области электронной коммерции, широкое распространение получило представление данных в формате XML. Для хранения и обработки большого объема XML данных в настоящее время используется два типа программных систем [1]:

- родовые (native) системы управления XML данными;
- реляционные/объектно-реляционные системы управления базами данных (СУБД), расширенные средствами хранения и обработки XML данных.

Наличие множества систем обоих классов делает актуальной задачу создания средств оценки производительности систем управления XML данными. Эти средства станут необходимым инструментом при выборе надлежащей системы управления XML данными пользователем. При этом пользователю должна быть предоставлена возможность задания параметров его типовых XML данных (документов) и рабочей нагрузки (смесей операций над данными).

Работа посвящена принципам построения проектируемой системы оценки производительности систем управления XML данными, названной PEVX (Performance Evaluation of XML DBMS).

PEVX функционирует на следующей конфигурации:



Каждый XML документ имеет уникальный адрес URL. Система управления XML данными взаимодействует с программным обеспечением Web (HTTP) сервера, а также с оковечной базой данных. Рабочая нагрузка для оцениваемой системы создается в виде смесей запросов языка XQuery [2].

На клиентском компьютере функционирует PEVX, выполняющий следующие основные действия:

- генерацию документов и формирование начальной базы XML данных;
- формирование рабочих смесей операций над базой XML и их исполнение на сервере;
- имитацию многопользовательского режима доступа к базе XML данных;
- реализация сценария оценки производительности, сбор и выдачу итоговых данных;
- диалог с пользователем для получения ряда параметров (характеристики начальной базы XML данных и смесей операций, число пользователей и т.д.).

Генерация документов и входящих в него элементов основывается на генераторе случайных чисел, распределенных по равномерному закону. Текстовые документы имеют следующий примерный вид:

```
<document (название, автор, doc_id)>
  <глава (chapter)>
    <раздел (section)>
      <параграф (paragraph)> Текст </параграф (paragraph)>
    </раздел (section)>
  </глава (chapter)>
</document>
```

Документы генерируются по указанной древовидной схеме с переменной асимметричной глубиной и размером. Генерация основывается на вероятностных параметрах документов, предоставляемых пользователем. В частности, задаются возможные числа глав в документе, разделов в главах, параграфов в разделах, предложений в параграфах, вероятность наличия автора, ссылки (URL) в параграфе и т.д. Автор документа (имя, фамилия) выбирается случайно из большого списка возможных имен (фамилий), причем число авторов коррелируется с общим числом документов в базе.

Текст в документе должен быть аналогичен естественному. В PEVX используется перечень из 10 тыс. наиболее употребительных слов английского языка, из которых осуществляется случайный выбор с учетом частот использования слов в английском языке. При формировании предложений в него с заданной вероятностью включается поисковая фраза из предварительно сгенерированного списка фраз.

Рабочая смесь операций имитирует действия над сгенерированной базой XML данных. Смесь состоит из двух типов операций: "запрос" и "обновление". Операции типа "запрос" не изменяют базу. Основной операцией является "Включить в базу документ с данным URL (Включение 1)". Параметры операций, такие как URL, автор, doc_id, случайно выбираются из значений, существующих в базе.

Перечень всех операций обоих типов предлагается пользователю для выбора. При выборе операции пользователь должен задать также долю этой операции в рабочей смеси.

PEVX имитирует многопользовательскую работу, причем работа одного пользователя состоит из исполнения рабочих смесей операций после некоторого случайного времени "обдумывания" (2-15 с). Время исполнения всего сценария задается пользователем (0.5-2 ч).

Производительность измеряется количеством операций обоих типов в единицу времени. Наряду с производительностью в результирующем отчете работы системы выводятся:

- относительная частота выполнения операций над базой данных;
- гистограммы времени выполнения каждой операции;
- начальный размер базы данных и изменение ее во времени.

Таким образом, проектируемая система PEVX позволит всесторонне оценить производительность систем управления XML данными.

Литература

1. Грейвс М. Проектирование баз данных на основе XML. Вильямс, 2002.
2. World Wide Web Consortium (W3C): XML Query. <http://www.w3.org/XML/Query>, 2000.

АНАЛИТИЧЕСКИЙ ОБЗОР СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ, ОСНОВАННЫХ НА МНОГОАГЕНТНОЙ АРХИТЕКТУРЕ ПОСТРОЕНИЯ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ

А. Джавадян

В докладе представлен аналитический обзор систем обнаружения вторжений в компьютерных сетях на основе использования системы комплексной защиты информации, архитектура которой строится на основе многоагентного подхода [1-6].

Защита информации в компьютерных сетях становится первоочередной проблемой, и возможность ее успешной реализации связана с необходимостью решения ряда сложных теоретических и прикладных задач. Значимость этих проблем обусловлена возрастанием ценности информации и возможными последствиями атак на компьютерные сети, целью которых является либо нарушение их конфиденциальности и/или целостности, либо их недоступность в некоторые периоды времени. Сложность решения проблемы вызвана рядом причин, среди которых следует, прежде всего, отметить возрастание масштабов корпоративных компьютерных сетей, относительно высокую динамику их реконфигурации, рост количества возможных объектов и целей атак (сетевых протоколов, операционных систем, прикладных программ), постоянное появление новых изобретенных и ранее неизвестных типов атак. [6]

В настоящее время, проблема защиты информации, решается посредством использования стандартных методов защиты информации (криптографическое закрытие информации, разграничения доступа, идентификация и аутентификация, обнаружение известных типов атак и устранение их возможных последствий и т.д.), входящих в состав операционных систем, приложений и информационных систем (ИС) [7]. Этот подход становится неэффективным для защиты современных распределенных ИС [2, 8]. Такие механизмы защиты становятся слишком "тяжелыми" для ИС с тысячами соединений в сутки, сотнями различных серверов и миллионами всевозможных типов атак. Как результат, существует большое количество способов для проникновения в такие системы, которые зачастую неизвестны администратору системы защиты. Одним из главных недостатков такого подхода к построению систем защиты является неспособность последних адекватно реагировать на сетевые атаки (вторжения).

Современные взгляды на проблему защиты информации характеризуются тенденцией динамического противодействия атакам, используя многоагентный подход при построении систем защиты информации [3, 4].

В работе предложен способ разработки многоагентной системы защиты информации. В настоящее время проектируется программный прототип системы. Аппаратная часть включает локальную компьютерную сеть. Программная часть базируется на операционной системе Windows 2000/XP, которая разрабатывается на Visual C++ . На первом этапе реализации строится среда обнаружения компьютерных атак, а также агентов динамического противодействия вторжениям.

Литература

1. Лукацкий А.В. Обнаружение атак. – С. Пет.:БХВ-Петербург. – 2001.
2. Russell Stuart, Norvig Peter. Artificial Intelligence: A Modern Approach: Prentice Hall. 1995.
3. Ягджян В.Г., Джавадян А. Ю. Методика построения интеллектуальной информационно поисковой системы основанной на модульно-распределительной архитектуре, журнал //научно-техническая информация (НТИ), сер. 2.- 2002. -N2. С. 20-23.
4. Джавадян А.Ю., Наабандян О.З. Разработка единой информационно-поисковой системы для административно-организационной работы: Сборник статей молодых ученых (естественные науки). – Ереван, 2000. – С. 30-33.
5. Gorodetski V.I., Kotenko I.V., Skormin V.A., Popyack L.J. Agent-based model of Information Security Systems: Architecture and Framework for Behavior Coordination //In Proceedings of the 1 International Workshop of Central and Eastern Europe on Multi-agent Systems.- St. Petersburg, 1999. -P.113-123.
6. Gorodetski V.I., Kotenko I.V., Skormin V.A., Popyack L.J. Ontology-based Multi-agent Model of Information Security System. In Lecture Notes in Artificial Intelligence, Vol.1711. New Directions in Rough Sets, Data Mining, and Granular-Soft Computing. 7th International Workshop, RSFDGrC'99. Yamaguchi, Japan. November 1999. Proceedings. -P.528-532.
7. Gorodetski V., Karsaev O. Algorithm of Rule Extraction from Learning Data. In Proceedings of the 8th International Conference (joint Europe-USA) "Expert Systems Application & Artificial Intelligence" (EXPERTSYS-96). IPT International, Paris, France, 1996. -P.133-138.
8. Городецкий В.И., Котенко И.В., Карсаев О.В. Многоагентная система защиты информации в компьютерных сетях: Механизмы обучения и формирования решений для обнаружения вторжений // Проблемы информатизации. -2000. № 2. С.67-73.

СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ: МЕХАНИЗМЫ ФОРМИРОВАНИЯ РЕШЕНИЙ

Э. Погосян, А. Джавадян

В статье рассматриваются вопросы, связанные с механизмами формирования решений в системах обнаружения вторжений компьютерных сетей.

С точки зрения задачи обнаружения вторжений основой используемой модели данных является обобщенный трафик событий (состояний). События ассоциируются с элементарными действиями (командами), связываемыми с различными активными субъектами - пользователями, программными средствами и реализуемыми в сети процессами. Как правило, до недавнего времени каждое событие, происходящее в сети или операционной системе (ОС), рассматривалось как локальное, не связанное с другими происшествие. В работе [4] описаны механизмы формирования решений об обнаружении вторжений, в которых явно просматриваются механизмы, основанные не только на обработке отдельных событий, которые происходят в сети или ОС, но также и комплексный подход объединения некоторых событий воедино, который можно представить в виде последовательности отдельных событий, происходящих в защищаемой компьютерной сети.

Большинство машинно-ориентированных методов решения задач основаны на создании стратегии планов. Уже существуют передовые интерактивные инструменты, помогающие в планировании и тестировании соответствующих стратегий через состязание команд в имитируемых средах. Однако законченные модели, включающие генерацию стратегии планов, трансформацию планов в стратегии и динамическом тестировании стратегий в конкурентных средах, разработаны недостаточно.

Нами была создана модель Общего Планирования и Динамического Тестирования (ОПДТ), где пространство поиска стратегий в дереве игры сужено при помощи

использования знаний и опыта в планировании стратегий, проведения соревнований в этой среде и комбинаторной сущности принятия решений. Модель обеспечивает регулярное улучшение стратегий, используя общие знания и достижения теории менеджмента, а также личный опыт экспертов. Это согласуется с широко известной моделью Портера [2] и рекомендациями по поиску стратегий методом Ботвинника [3].

Две основные компоненты модели содержат общие знания планирования стратегий – онтологию Планирования Стратегий (ПС) и метод поиска стратегий, способный обращаться к онтологии ПС, и, как результат, изменяется процедура поиска стратегий (рис 1, 2).

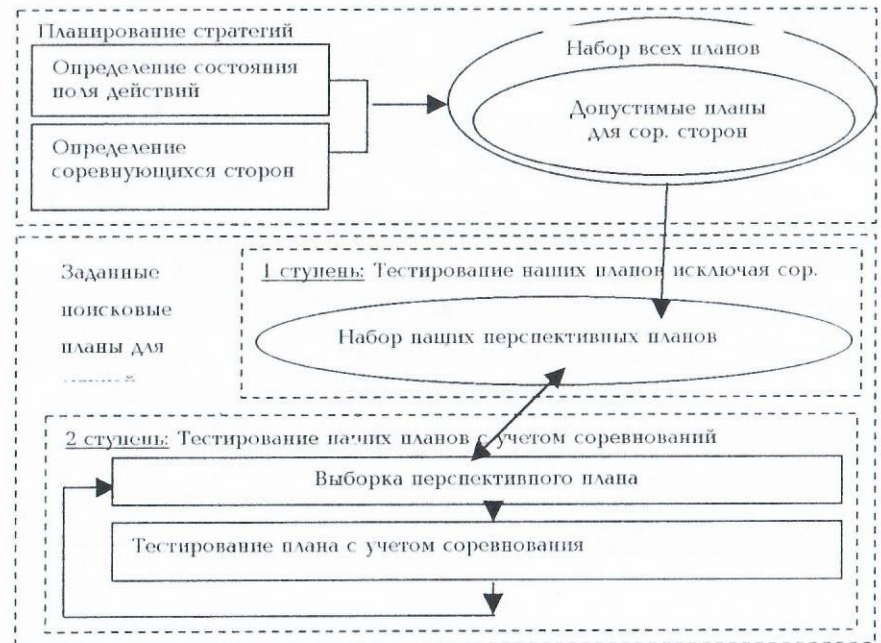


Рис. 1

Как результат, предоставляется возможность выбора наилучшей стратегии одним из методов, описанных в [1].

По сравнению с традиционными методами этот подход позволяет существенно повысить эффективность защиты информации, защиты системы, в том числе ее адекватность, отказоустойчивость, универсальность, гибкость и т.д.

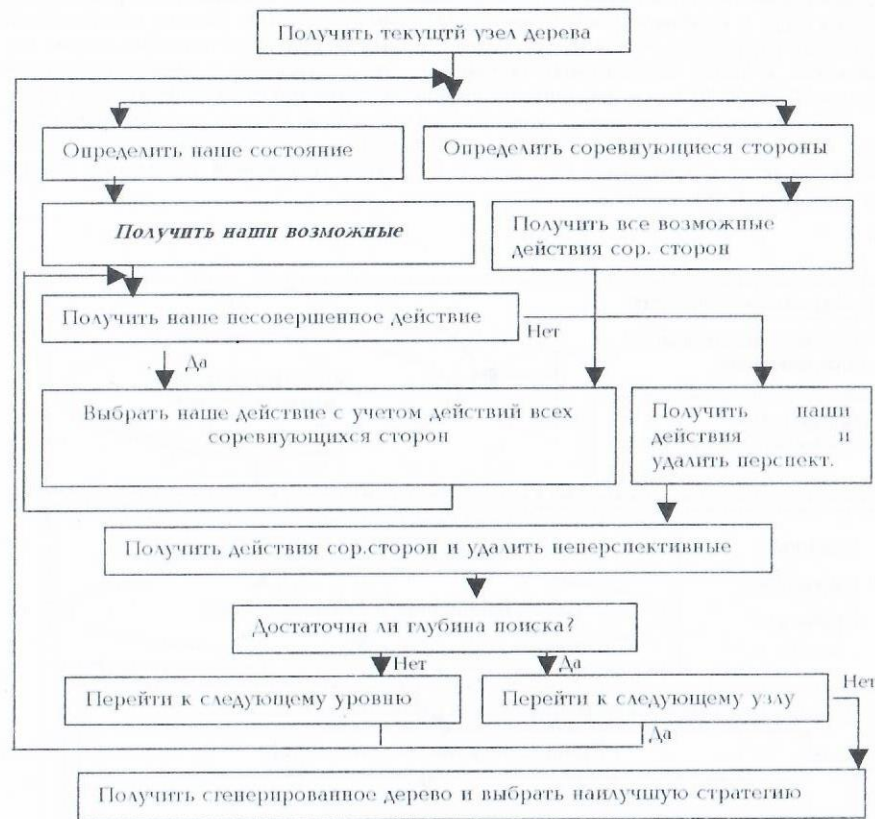


Рис. 2

Литература

1. Pogossian E. Management Strategy Search and Programming //Proc.of the 2nd Intern. - 1999.
2. Porter M. Competitive Strategy Techniques for Analysis Industries and Competitors //Free Press. -1980.
3. Botvinnik M.M. On Approximate Resolution of Search Problems. - М.: - Sov. Radio. - 1979.
4. Городецкий В.И., Котенко Н.В., Карсаев О.В. Многоагентная система защиты информации в компьютерных сетях: механизмы обучения и формирования решений для обнаружения вторжений // Проблемы информатизации. -2000. - № 2. - С.67-73.

ОБ ОРГАНИЗАЦИИ ПРОЦЕССА САМООБУЧЕНИЯ В СИСТЕМЕ ИНТЕЛЛЕКТУАЛЬНОГО МАШИННОГО ПЕРЕВОДА

Э. Манукян, С. Манукян, А. Манукян, А. Саядян

Требования, предъявляемые к повышению качества машинного перевода, служат предпосылкой для отказа от принципа пословного, фразеологического перевода [1,2] в пользу двухэтапного подхода, при котором входной текст (фрагменты текста) сначала преобразуется в семантическую сеть фактов, отображающих описанные в тексте объекты и процессы [3]. Далее на основе сгенерированной сети строятся предложения выходного текста, используя базу словарей и описаний конструкций выходного языка. При таком подходе определяющим является выбор типов от-

ношений между узлами семантической сети, привязка его фрагментов в единую систему, отражающую смысл входного текста, и выбор алгоритма построения сети согласно результатам анализа слов входного предложения. Достигнуть этого можно проведением морфологического, грамматического и семантического анализа текста с последующим формированием древовидного графа.

Ключевым в данной схеме перевода является этап семантического анализа, который невозможно реализовать, не имея развитой базы знаний о предметном мире, обсуждаемом в переводимом тексте. Необходимо отметить, что формирование БЗ, являющееся весьма трудоемким процессом, можно принципиально упростить, если следовать стратегии предложенного подхода.

Так, древовидный граф, полученный на промежуточном этапе перевода, можно использовать для доформирования БЗ, интерпретируя его как новую подсхему к уже имеющейся сети БЗ (рис.). Это делает его автоматически настраиваемой на предметную область, описанную в переводимом тексте.

Принцип наращивания (дополнения) БЗ все новыми знаниями путем перевода туда отношений, объектов и атрибутов из динамически поступаемых на вход переводчика текстов позволит использовать процесс самообучения для весьма развитых БЗ.



Рис. Организация процесса самообучения

Литература

1. Arnold Douglas, Balkan Lorna, Meijer Siety. MACHINE TRANSLATION An Introductory Guide, Linguistics, University of Essex, Colchester, CO4 3SQ, UK, doug@essex.ac.uk., March 6, 2001.
2. Morneau Rick. The Lexical Semantics of a Machine Translation Interlingua First Internet draft date: July 30, 1995, Current draft date: April 10, 2001.
3. Манукян Э. и др., Основы построения интеллектуальной системы машинного перевода текстов. Сб. материалов годичной научно-технической конференции ГИУА.Т.1. - Ереван. - 2001. - С. 306.

ОПРЕДЕЛЕНИЕ ЭКСПЛУАТАЦИОННЫХ ХАРАКТЕРИСТИК СЛОВАРНЫХ БАЗ СИСТЕМ МАШИННОГО ПЕРЕВОДА

Э. Манукян, А. Манукян

Составной частью автоматизированных систем машинного перевода (АСМП) является подсистема управления словарной базой. Эффективность использования уни-

версальных программных средств по их ведению будет крайне занижена ввиду специфичности режимов эксплуатации.

Поскольку в процессе разработки ряда версий АСМП выяснилось, что эффективность АСМП зависит от физической организации словарной базы, мы остановились на выборе так называемой "иерархической системы Гора" [1].

Целью данной работы является получение оценок объема памяти (Q) в зависимости от характеристик множества слов, при использовании схемы Гора. Наличие оценок позволит проектировщику АСМП выбрать оптимальный вариант модификации схемы Гора. В [1,2,3] получены оценки объема памяти при условии равновероятности появления символов в словах. Однако при проектировании АСМП известна более детальная информация о распределении символов в словах. В частности, можно получить двухмерную гистограмму частот появления букв в заданных позициях слов $\bar{G}(i, k)$, где i - номер позиции буквы в слове ($1 \leq i \leq s$, s - количество символов в максимально длинном слове словаря), k - номер буквы в алфавите данного языка ($1 \leq k \leq q$, q - число символов в алфавите).

Ниже предлагается схема получения оценок Q при условии наличия указанной гистограммы.

Предлагается схема физической организации словарной базы, представленной системой взаимосвязанных векторов $V_{ij}(k)$, где i ($1 \leq i \leq s$) - номер уровня, совпадающий с позицией в слове; j - порядковый номер вектора на данном i -ом уровне; k - номер буквы в алфавите. Элемент $V_{ij}(k) > 0$ указывает на вектор $(i+1)$ -го уровня, представляющий слова, число символов которых превышает i . Одновременно предполагается, что первые i символов этих слов находятся на пути, начинающемся от некоторого k -го элемента $V_{1,1}(k)$ до элемента $V_{ij}(k)$.

Предположим, что словарная база состоит из одного слова. Тогда вероятность появления нулевого значения элементов в векторе $V_{1,1}(k)$ будет определяться как

$$P(V_{1,1}(k) \neq 0) = G(1, k),$$

где $G(1, k)$ - вероятность появления k -го символа в первой позиции данного слова, определяемая из исходной гистограммы по формуле

$$G(i, k) = \bar{G}(i, k) / \sum_{k=1}^q \bar{G}(i, k) \quad \text{при } 1 \leq i \leq s.$$

Если в словаре имеется N_1 слов, то согласно правилу формирования векторов $V_{ij}(k)$, элемент $V_{1,1}(k)$ будет нулевым, если k -й символ будет отсутствовать в первой позиции у всех слов. Тогда для словарной базы, состоящей из N_1 слов, будем иметь

$$P(V_{1,1}(k) = 0) = (1 - G(1, k))^{N_1}, \quad 1 \leq k \leq q.$$

А оценка математического ожидания общего числа не нулевых элементов в векторе $V_{1,1}(k)$ определяется как

$$\sum_{k=1}^q P(V_{1,1}(k) \neq 0) = \sum_{k=1}^q (1 - (1 - G(1, k))^{N_1}), \quad 1 \leq k \leq q.$$

Заметим, что каждый k -ый, не нулевой элемент вектора $V_{1,1}(k)$ представляет слова, среднее количество которых определяется как $m_1(k) = N_1 G(1, k)$. Отсутствие указателя на следующий уровень задается отрицательным значением данного элемента в случае, если все эти слова заканчиваются на данном k -ом символе.

Вероятность такого события определяется как $p_1^{m_1(k)}$, где p_1 - вероятность появления слов с одним символом. Последнюю можно определить из исходной гистограммы по формуле

$$p_1 = \left(\sum_{k=1}^q G(2, k) - \sum_{k=1}^q G(1, k) \right) / \sum_{k=1}^q G(1, k).$$

Вероятность появления исходящей ветви, а следовательно, и вектора указателей на следующем уровне (связанного с данным элементом) можно определить как $1 - p_1^{m_1(k)}$.

Таким образом, из каждого k -го элемента $V_{1,1}(k)$ будет исходить ветвь, если он отличен от нуля и если имеются многосимвольные слова, начинающиеся с k -го символа. Оценка математического ожидания числа исходящих ветвей из вектора $V_{1,1}(k)$ определится из соотношения

$$R_1 = \sum_{k=1}^q ((1 - (1 - G(1, k))^{N_1}) (1 - p_1^{m_1(k)})) \quad \text{для } 1 \leq k \leq q.$$

Так как R_1 представляет собой среднее число векторов $V_{2,1}(k)$, то вероятность появления слов, имеющих два или более символов, можно определить из исходной гистограммы по формуле

$$\bar{p}_1 = \sum_{k=1}^q G(2, k) / \sum_{k=1}^q G(1, k).$$

Принимая условие равновероятности распределения слов между векторами $V_{2,1}(k)$, среднее число слов, представляемых каждым из векторов $V_{2,1}(k)$, определится как

$$N_2 = (N_1 \bar{p}_1) / R_1.$$

Обобщая многошаговую схему определения величин R_1 и N_2 последние можно представить как $R_1 = F_1(N_1, G(i, k))$, $N_2 = F_2(R_1, N_1, G(i, k))$. Тогда, повторяя расчеты для всех s уровней и используя функции F_1, F_2 получим последовательность R_{i-1} ($2 \leq i \leq s$) и N_i ($2 \leq i \leq s$). Требуемые при этом значения вероятностей появления слов с t символами (p_t) и слов с числом символов, превышающим t (\bar{p}_t), будут определяться как

$$p_t = \left(\sum_{k=1}^q G(t+1, k) - \sum_{k=1}^q G(t, k) \right) / \sum_{k=1}^q G(1, k)$$

$$\bar{p}_t = \sum_{k=1}^q G(t+1, k) / \sum_{k=1}^q G(1, k).$$

В итоге искомые затраты объема памяти для организации словарной базы определяются по соотношению

$$Q = q \sum_{t=1}^s \prod_{j=1}^{t-1} R_j.$$

Полученные оценки позволяют проектировщику определить эффективность применения выбранной схемы Гора в процессе разработки словарной базы АСМП.

Литература

1. Кнут Д. Е. Искусство программирования для ЭВМ. Т. 3: Сортировка и поиск. - М.: Мир, 1978.
2. Сиколенко В. В. Сервер Oracle: Текущее состояние.
3. Martin J. Computer Data-Base Organization. (2nd ed.) Englewood Cliffs. - N. J.: Prentice-Hall, 1977.

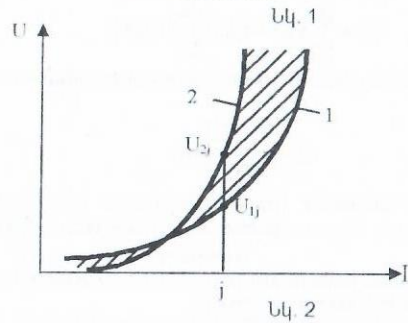
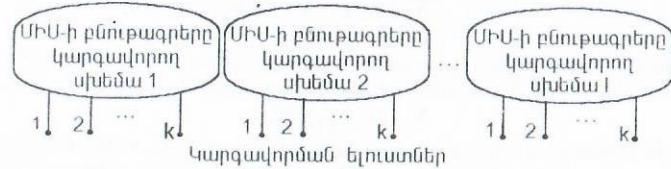
ԱՆԱԼՈՂԱՅԻՆ ՍԻՆՏԱՆՆԵՐԻ ԹԵՏԱՎՈՐՄԱՆ ՄԻՋՈՑՆԵՐ

Կ. Սկրոդյան

Ներկայումս մի շարք մեծ ինտերբալ սինտանների (ՄԻՍ) մեծ գնի և շահագործման ընթացքում ելքային բնութագրերի արագ փոփոխման պատճառով արդիական է դարձել նրանց՝ արտաքինից կարգավորող սինտայի ավելացումը, որի շնորհիվ երկարաձգվում է ՄԻՍ-ի շահագործման ժամանակահատվածը: Սակայն գրականության ուսումնասիրությունը ցույց է տալիս նաև, որ առայժմ մշակված չէ կարգավորող սինտայի, ինչպես նաև նրա պարամետրերի արժեքների ընտրության համար համապիտանի մեթոդիկա: Այսինքն, նմանատիպ համակարգերում սովորաբար, օրինակ՝ ՄԻՍ-ի համար, ընտրվում է նախապես հայտնի միակ կարգավորող սինտան, որը բնականաբար, բոլոր դեպքերում չի կարող լինել լավագույնը: Այդ նկատառումով գեկուցման մեջ նշված խնդրի լուծումը առաջարկվում է ավելի լայն տեսանկյունով:

Տրված է՝

1. ՄԻՍ-ը (նկ. 1), որն ունի n մուտք, m ելք և k ելուստներ, որոնցով ՄԻՍ-ի բնութագրերը կարգավորող սխեման միանում է արտաքինից: Ներկայումս նախագծվող ՄԻՍ-երի մի մասի համար հատուկ նախատեսվում են այդպիսի ելուստներ: Արանց բացակայության դեպքում, որպես այդպիսին կարող է ծառայել մուտքերի և ելքերի բազմության որոշակի ենթաբազմությունը:
2. ՄԻՍ-ի բնութագրերը կարգավորող սխեմաների բազմությունը (1, 2, ... l): Յուրաքանչյուր կարգավորող սխեմայի համար հայտնի է նրա կառուցվածքը (տրված որևէ սխեմատեխնիկական մոդելավորման ծրագրի մուտքային կամ ներքին նկարագրման լեզվով), էլեմենտների արժեքները, նրա արտաքին ելուստների քանակը:
3. ՄԻՍ-ի որևէ ելքային ընդհանրացված բնութագրի (օրինակ վոլտ-ամպերային բնութագրի) էտալոնային (նկ. 2, գիծ 1) և շահագործման հետևանքով փոփոխված (նկ. 2, գիծ 2) տեսքերը:
4. Էտալոնային և շահագործման հետևանքով փոփոխված (1 և 2) բնութագրերի շեղվածության բուլյատրեի չափը բացարձակ (Δ_{max}) կամ հարաբերական (ε_{max}) տեսքով:



Անհրաժեշտ է՝

1. ՄԻՍ-ի բնութագրերը կարգավորող սխեմաների բազմությունից ընտրել այն i -րդ սխեման և նրա էլեմենտների պարամետրերի այնպիսի արժեքներ, որոնց դեպքում էտալոնային և շահագործման հետևանքով փոփոխված կորերի շեղվածությունը (Δ կամ ε) փոքր լինի տրված Δ_{max} կամ ε_{max} չափից

$$\Delta < \Delta_{max} \\ \varepsilon < \varepsilon_{max}$$

$$\Delta = \sum_j |U_{2j} - U_{1j}|, \text{ իսկ } \varepsilon = \sum_j \left| \frac{U_{2j} - U_{1j}}{U_{2j}} \right|:$$

Չեկուցման մեջ նկարագրվում են՝

- Մշակված մեծ ինտեգրալ սխեմաների ընտրության մեթոդիկան:

- Ընտրված օպտիմալացման մեթոդների համալիրը, որը լավագույնն է դրված խնդրի համար:
- Մշակված մեծ ինտեգրալ սխեմաների արտաքինից միացվող կարգավորող սխեմաների ընտրության և նրանց տարրերի հաշվարկը իրականացնող ծրագրային համակարգը Delphi ծրագրային գործիքային միջավայրում, որն իր մեջ ներառում է նաև սխեմատեխնիկական մոդելավորման SPICE համակարգը:
- Որոշակի մեծ ինտեգրալ սխեմաների ընտրության ստացված արդյունքները մշակված ծրագրային համակարգի օգնությամբ և նրանց համեմատումը գրականությունից հայտնի տվյալների հետ:

Գրականություն

1. Tiemeijer L.F., Thijs P.J.A., Dongen T.V., Jansen E.J. and Binsma J.J.M. Direct electronic compensation of the amplification nonlinearity in semiconductor laser amplifiers. Received 18 October 1993, accepted for publication 26 January 1994. Philips Optoelectronics Center, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands.

МОДЕЛИ ДАННЫХ И ЯЗЫКИ ПРОГРАММИРОВАНИЯ (ПРОЕКТ ЯЗЫКА DBTDL)

Э. Иваниця, Р. Марганян

Каждая база данных создается в соответствии с той или иной моделью данных. Поэтому понятие модели данных является центральным в теории баз данных. В языках же программирования центральным является понятие типа данных. Согласно современной трактовке, тип данных определяет множество объектов посредством множества операций. Очевидная схожесть определений моделей и типов данных приводит к постановке задачи выработки общего взгляда и унификации этих двух понятий.

Понятия типов и моделей данных могут быть унифицированы в одном языке программирования с возможностью построения баз данных определенных моделей.

Соотношения между базой данных, схемой базы данных и моделью данных определяются следующим образом:

- модель данных - это набор параметризованных типов данных;
- схема базы данных - это набор типов данных, являющихся результатом конкретизации параметризованных типов данных;
- база данных - конкретизация типов, определенных в схеме базы данных.

Приведенные определения нуждаются в нескольких уточнениях. Во-первых, любая модель данных имеет дело не только с параметризованными, но и с конкретными типами данных. Во-вторых, модель содержит набор типов данных и языковые средства спецификации ограничений целостности и формирования запросов к базе данных, которые необходимо связать с понятиями типов данных. Итак, используя принятые в программировании термины, можем сказать, что модель данных есть система конкретных и параметризованных типов, которая задает и язык определения данных, и язык манипулирования данными, и набор ограничений целостности. Таким образом, в программировании наиболее близким аналогом понятия модели данных является понятие языка программирования.

Следующий вопрос - что соответствует в языках программирования понятию схемы базы данных? Если предположить, что программа выполняется неопределенно долго и допускается параллельная работа различных частей программы с одними и теми же данными, то аналогия между процессами работы системы управления базами данных и выполнения программы становится очевидной. В таком случае мы должны заключить, что схема базы данных - это совокупность определений типов, констант и переменных средствами модели данных. База данных - это совокупность значений переменных базы данных в настоящий момент времени.

Целью настоящей работы является разработка высокоуровневых языковых конструкций, предоставляющих следующие возможности:

- Построение различных моделей данных с последующей их конкретизацией.

- Инкапсуляция базы данных, т.е. полноекрытие реализационных операций.
- Параллельная работа с несколькими базами данных.
- Построение подсистем баз данных, обеспечивающих пользователям собственное видение баз данных.
- Построение собственных языков запросов пользователей, включающих необходимые сценарии ввода-вывода, процедуры обработки и т.п.
- Обработка аварийных ситуаций, возникающих при исполнении тех или иных операций.
- Построение объектно-ориентированных интерфейсов для реляционных, иерархических и сетевых баз данных.

Характерными свойствами предлагаемых языковых конструкций сформированных выше требований являются:

- типовая полнота;
- возможность статического контроля типов;
- поддержка смешанных вычислений;
- возможность создания баз данных.

Описываемые в настоящей работе языковые конструкции являются усовершенствованными версиями соответствующих конструкций специализированного языка описания задач TDL (Task Description Language), реализованного в ряде пакетов прикладных программ специального значения. Эти языковые конструкции можно рассматривать как отдельный язык программирования баз данных, который назовем DBTDL (DataBase Task Description Language).

DBTDL обладает широкими возможностями конструирования конкретных и родовых произвольных типов данных. Эти средства позволяют:

- представить модель данных в виде совокупности родовых и конкретных типов данных и затем конкретизировать ее для создания схемы конкретной базы данных;
- вывести на уровень использования только те операции, которые характеризуют рассматриваемую модель данных, и скрыть от пользователя реализационные операции;
- образовывать подсистемы пользователей как типы данных, базирующиеся на типах данных модели данных и обеспечивающие преобразование данных схемы базы данных в данные подсистемы.

Данная статья является кратким обзором разрабатываемого языка программирования DBTDL, в котором поставлена задача выработки общего взгляда для определения моделей данных и типов данных, а также унификации этих двух понятий в одном языке программирования.

Литература

2. Damers A.J., Donahue J.E. Type Completeness as a Language Principle.

ՔԱՆԱԿԱ-ՖԻՆԱՆՍԱԿԱՆ ՎԵՐԼՈՒԾՈՒԹՅԱՆ ԽՆԴԻՐՆԵՐԻ ԾՐԱԳՐԱՎՈՐՄԱՆ ԱՎՏՈՄԱՏԱՑՄԱՆ ՄԵԹՈԴԻ ՄԱՍԻՆ

Գ.Սոսիկյան

Տվյալ աշխատանքը նվիրված է ֆինանսական մաթեմատիկայի խնդիրների լուծման համար մշակված պրոբլեմային կոդձնորոշված լեզվին:

Լեզուն ընդգրկում է ֆինանսական հաշվարկներն իրականացնող օպերատորներ, որոնք մշակվել են ըստ հետևյալ երկու սկզբունքների:

Ըստ առաջին սկզբունքի ծրագրավորողը ստանում է մաթեմատիկական հաշվարկային մոդելը առավել դյուրին լուծելու հնարավորություն՝ նույնիսկ տեղյակ չլինելով հաշվարկվող մաթեմատիկական մոդելի հաշվարկման բանաձևի մանրամասներին:

Այն օպերատորը, որն օգտագործում է նշված սկզբունքը, այս համակարգում կոչվել է օպերատոր-բանաձև: Նրա աշխատանքի ելությունն այն է, որ գրվում է օպերատորը, որը ներկայացնում է հաշվարկվող մոդելի բանաձևը, ապա որպես առաջին օպերատոր նրան է փոխանցվում ֆունկցիայի արժեքը ներկայացնող փոփոխականը, դրան հաջորդում են բանաձևի օպերատորները, ըստ նախագծողի կողմից տրված հաջորդականության: Վերջում, հատուկ

տառանշանով, նշվում է օպերատոր-բանաձևի այն պարամետրը, որը պահանջվում է հաշվարկել: Այնուհետև ինտերպրետատորը սկսում է ծրագրի տեքստի վերլուծությունը, որի ընթացքում այն կարող է հայտնել հաշվարկի անհնարինություն մասին, եթե կատարվել է թերականական սխալ կամ պահանջվել է հաշվարկել անհաշվարկելի օպերատոր, որից հետո ինտերպրետատորը կատարում է ծրագիրը, ստանում արժեքը և ըստ պահանջի արտածում այն: Արդյունքում այնպիսի բանաձևի լուծումը, ինչպիսին է p.1-ը

$$S_v = \frac{P_v K_0 (1 + n * i)}{K_1} \quad \begin{array}{l} p.1 \text{ Արտարժույթի փոխանակում և տոկոսների աճեցում} \\ \text{տարրերակ } \$ > D > D \$ \end{array}$$

կարելի է ներկայացնել հետևյալ օպերատոր-բանաձևով

CONVERTVRRV !S_v, !P_v, !K₀, !n, !i, !K₁ օպ 1

Օպ 1-ում պարամետրների առջևում դրված '!' նշանը ցույց է տալիս, որ այդ պարամետրը կարող է լինել հաշվարկելի:

Երկրորդ սկզբունքն օգտագործվում է այն դեպքերում, երբ մինչև օպերատորի կիրառումը հնարավոր չէ նախատեսել օպերատորների նախնական կոնստանտ բանակ: ՊԿԾԼ-ում ստեղծվել են մասնագիտացված օպերատորներ, որոնք թույլ են տալիս մուտքագրել անհրաժեշտ բանակով արժեքներ:

Դետևյալ բանաձևի լուծումը, ինչպիսին է p.2-ը

$$Q = P * (1 + n * i) - \sum (R_j * (1 + i_j * i_j)) \quad \begin{array}{l} p.2 \text{ Պարտքի մարումը մասերով վաճառ-} \\ \text{կանի մեթոդ} \end{array}$$

կարելի է ներկայացնել հետևյալ օպերատոր-բանաձևով

PPAYMERCHRULE !Q, !P, !n, !i [, R_j, !i_j, !i_j] օպ 2

Օպ 2-ում 5, 6, 7 պարամետրերը ներառված են քառակուսի փակագծերի մեջ, որը նշանակում է, որ այդ խումբ փոփոխականները կարող են խնդրվել կրկնվել n անգամ (n >= 0): Զանի որ նրանք չեն նշված որպես հաշվարկվող, ապա նրանց հաշվարկի պահանջը կբերի սխալի: Մնացած 1-ից մինչև 4 պարամետրերը կարող են լինել հաշվարկելի:

Այսպիսով ծրագրավորման ժամանակ ծրագրավորողի համար հաշվարկվող բանաձևի մանրամասներին ծանոթանալու կարիքը վերանում է, քանի որ նա կարող է հաշվարկել մեկ օպերատոր-բանաձևով նույն բանաձևում պարունակող բոլոր հաշվարկելի օպերատորները գումարած բանաձևի արդյունքը:

Այս մեթոդը կիրառվել է քանակա-ֆինանսական վերլուծության խնդիրների ծրագրավորման ավտոմատացման միջոցների նախագծման համակարգում նպատակային ստեղծված ինտերպրետատորի աշխատանքի մեջ: Համատեղելով այս մեթոդը վիզուալ ծրագրավորման մեթոդի հետ՝ վերոհիշյալ համակարգում ստացվել է գործնականում կիրառելի և օգտակար արդյունք, որի իմաստն այն է, որ օգտագործող-ֆինանսիստը պետք եղած ժամանակ հնարավորություն ունի կատարելու ֆինանսական իրավիճակի շտապ նկարագրում ու լուծում տվյալ համակարգի միջոցով:

Գրականություն

1. Четыркин Е. Финансовая математика. - М.: Дело. - 2000. - 397с.
2. Шнайт Г. Теория и практика С+++, Санкт-Петербург: ВИВ. -2000. -416с.
3. Дембаускас А.П. Финансовая информатика. -М.: Финансы и статистика. -1987. -207с.
4. Рожнов В.С., Бегоцкая Г.К., Гекон С.В. Автоматизированные системы обработки финансово-кредитной информации. Учебник.

РЕАЛИЗАЦИЯ ОДНОГО АЛГОРИТМА ГЕОМЕТРИЧЕСКОГО МОДЕЛИРОВАНИЯ ПОВЕРХНОСТЕЙ ЧЕТЫРЕХМЕРНОГО ПРОСТРАНСТВА В СИСТЕМЕ AutoCAD

К. Согомонян, К. Туманян, Н. Казарян

Проектирование сложных криволинейных объектов технических форм, их расчет и воспроизведение требуют разработки математических моделей, основанных на тех или иных методах геометрического моделирования (конструирования поверхностей).

Алгоритмический принцип разработки конструктивных методов образования алгебраических форм заключается в следующем:

- выбирается некоторое множество исходных элементов (примитивы - элементы);
- выбирается некоторая совокупность геометрических операций (примитивы - операции);
- алгебраическая форма (результат) рассматривается как множество, полученное из некоторой совокупности исходных элементов с применением конечного числа операций из совокупности примитивов.

Особенностью предлагаемого метода является то, что среда, где осуществляется творческий процесс геометрического конструирования, не ограничивается рамками трехмерного пространства, а расширяется до четырехмерного пространства R^4 , где имеются более широкие возможности для творчества.

Задача.

Дано: в R^4 пара прямых $T'_1(I', 2')$, $T'_2(3', 4')$ и точка $A'(x_A, y_A, z_A, t_A)$. Необходимо методом параллельного проецирования найти отображение точки $A' \in R^4$ в R^2 $\varphi: \Lambda'(R^4) \rightarrow \Lambda_n(R^2)$, а затем $\psi: \Lambda_n(R^2) \rightarrow E(R^3)$.

Прежде всего рассмотрим процесс получения плоской бинарной модели четырехмерного пространства, используя исходные элементы рассматриваемой задачи.

Алгоритм решения:

1. В R^4 выберем двумерную поверхность H' .
2. Через прямую T'_1 и точку A' проведем проецирующую плоскость $\alpha(T'_1, A')$ до пересечения с плоскостью H' . В результате пересечения плоскостей α и H' получим точку $A_1 \in H'$ с координатами $(x_{A_1}, y_{A_1}, 0, 0)$, аналогично от пересечения плоскостей $\beta(T'_2, A')$ и H' получим точку $A_2 \in H'$ с координатами $(x_{A_2}, y_{A_2}, 0, 0)$.

Изменяя точку A' , получим два множества $\{A_{1i}\}_{i=1..n}$ и $\{A_{2i}\}_{i=1..n}$, каждое из которых можно представить как плоскую кривую - g_1 и g_2 , т.е. можно сказать, что множество $\{A'_{ij}\}_{i=1..n, j=1..n} \in R^4$, которое может представлять образ какой-нибудь поверхности в плоской бинарной модели пространства R^4 , интерпретируется в виде декартова произведения двух линий $(g_1 \times g_2) \in H'$.

Представим трехмерную модель четырехмерного пространства R^4 на основе исходных элементов задачи.

Для этого представим трехмерную прямоугольную декартову систему координат следующим образом:

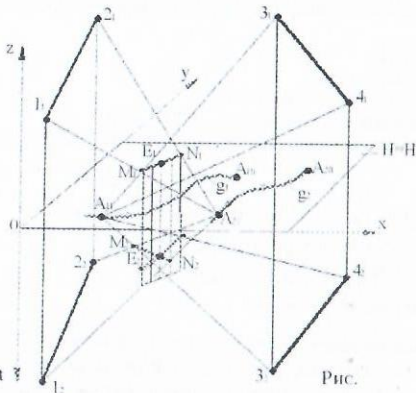


Рис.

1. Совместим плоскость $хоу$ с H' (см. рис.), тогда плоскость $хоz$ расположится перпендикулярно H' , а плоскость $х0t$ для облегчения дальнейших построений совместим с плоскостью $хоz$.

2. Проецируем прямые $T'_1(I', 2')$ и $T'_2(3', 4')$, где $T'_1(x'_1, y'_1, z'_1, t'_1)$, $T'_2(x'_2, y'_2, z'_2, t'_2)$, $T'_3(x'_3, y'_3, z'_3, t'_3)$, $T'_4(x'_4, y'_4, z'_4, t'_4)$, являются пары прямых со следующими параметрами;

$$\left\{ \begin{matrix} 1_1(x_1, y_1, z_1), 2_1(x_2, y_2, z_2) \\ 1_2(x_1, y_1, z_1), 2_2(x_2, y_2, z_2) \end{matrix} \right\} \left\{ \begin{matrix} 3_1(x_3, y_3, z_3), 4_1(x_4, y_4, z_4) \\ 3_2(x_3, y_3, z_3), 4_2(x_4, y_4, z_4) \end{matrix} \right\}$$

3. Создадим плоскости в виде треугольников $\Delta 1(1_1, 2_1, A_{21})$, $\Delta 2(3_1, 4_1, A_{21})$, $\Delta 3(1_2, 2_2, A_{11})$, $\Delta 4(3_2, 4_2, A_{11})$.

4. Определим линии пересечения соответствующих треугольников (если они непараллельны): $\Delta 1 \cap \Delta 2 \rightarrow (M_1 N_1)$, $\Delta 3 \cap \Delta 4 \rightarrow (M_2 N_2)$.

5. Определим общую точку прямых $(M_1 N_1)$ и $(M_2 N_2) \rightarrow E_1, E_2$, для чего:

- а) проведем через прямую $(M_1 N_1)$ плоскость γ , перпендикулярную плоскости $хоу$;
- б) $\gamma \cap (M_2 N_2) \rightarrow$ точка $E_2(x_E, y_E, z_E)$, а значит, определяется и точка $E_1(x_E, y_E, z_E)$ с помощью линии связи $(E_1 E_2)$, проведенной до пересечения с прямой $(M_1 N_1)$.

Повторяя этот алгоритм для всех точек $\{A_{1i}\} \in g_1$ и $\{A_{2i}\} \in g_2$, получим $\{E_{1j}\}_{j=1..n}$ и $\{E_{2j}\}_{j=1..n}$, которые являются трехмерной моделью поверхности, образ которой представлен множеством $\{A'_{ij}\}_{i=1..n, j=1..n}$.

Изменяя параметры прямых T'_1 и T'_2 , а также линий g_1 и g_2 , которые представляют множества $\{A_{1i}\}$ и $\{A_{2i}\}$, можно получить разнообразные поверхности R^3 .

Для автоматизации процесса реализации описанного алгоритма, что облегчит работы конструктора, разработано программное обеспечение в виде функции (команды) функционального языка AutoLISP, действующего в среде графической системы AutoCAD - функция (KPOV2).

Ввод необходимых параметров производится в интерактивном режиме.

ԺԱՄԱՆԱԿԱՅԻՆ ԱՌՈՒՄՈՎ ԶԿԱՆՏԱՑՎԱԾ ԵՎ ՄԻՋԻՆԱՑՎԱԾ ՎԻՃԱԿՆԵՐՈՎ ՄՈՂԵԼՆԵՐ

Ա. Կուլախյան, Վ. Սողոմոնյան, Ա. Սելիքյան

Խառը մոդելավորման համակարգերը ենթադրում են տրամաբանական մոդելների որոշակի սահմանափակումներ:

Նախ, այդ մոդելները պետք է լինեն ասինքրոն, այսինքն պետք է հաշվի առնեն փոխանջատման հասպարդումները: Դա բացատրվում է նրանով, որ դրանք աշխատում են ժամանակի մեջ անընդհատ մոդելների հետ, և համապատասխանաբար նրանք նույնպես պետք է արտադրվեն ժամանակային հարաբերություններ:

Երկրորդ տրամաբանական մոդելների վիճակները չնայած քվանտացված են, այնուամենայնիվ, պետք է հաշվի չառնեն այդ հանգամանքը և նրանց էլքերում պետք է ստացվեն ժամանակի մեջ անընդհատ արտապատկերումներ էլքային ազդանշանների բազմության վրա: Այս պահանջը վերաբերում է տարրերի այն տրամաբանական մոդելներին, որոնց էլքերը կապված են ժամանակի մեջ անընդհատ անալոգային մոդելներին, որոնց էլքերում տեղադրվում են ինտերֆեյսներ, որոնց էլքից պետք է դուրս գան անընդհատ ազդանշաններ, քանի որ ազդանշանի թռիչքները կարող են դժվարացնել անընդհատ անալոգային մոդելների հաշվարկային ալգորիթմների կայունությունը: Ներքին տարրերի մոդելներում ազդանշանների նմանատիպ ձևափոխումներ չեն պահանջվում:

Երրորդ ակնհայտ պահանջը ինտերֆեյսի առկայությունն է այն տրամաբանական մոդելների մուտքերում, որոնք կապված են անընդհատ մոդելների հետ: Այդ ինտերֆեյսները պետք է ձևափոխեն անընդհատ ազդանշանները տրամաբանական քվանտացված ազդանշանների: Այդ պահանջը հեշտ իրականացվում է շենային ֆունկցիաների օգնությամբ: