

066

ЕДИНЕН ЦЕНТЪР ПО МАТЕМАТИКА И МЕХАНИКА

Валентин Томов Томов

ИНТЕЛЕКТУАЛИЗИРАНИ
ПРИЛОЖНИ ПРОГРАМНИ
СИСТЕМИ

Научна специалност

01.01.08. Математическо осигуряване на изчислителни
машини и системи

ХАБИЛИТАЦИОНЕН ТРУД

за получаване на научното звание

"Старши научен сътрудник първа степен"

София, 1988

ХАРАКТЕРИСТИКА НА ХАБИЛИТАЦИОННИЯ ТРУД

а) Хабилитационният труд представлява научно-изследователски труд, отпечатан в негови съществени части. Част от включените в труда разработки са внедрени.

б) Хабилитационният труд е в областта на математическото осигуряване на изчислителни машини и системи (№ 01.01.08 от номенклатурата на научните специалности) и е посветен на въпросите за изграждане, изследване и разработване на Интелектуализирани Приложни Програмни Системи (ИППС).

Основни цели и задачи на труда са:

- да се изградят, разработят и обосноват концепциите, общите принципи и функционалната структура на ИППС;
- да се изследват и програмно да се експериментират такива системи;
- да се заговорят някои нужди на практиката от ИППС в определени важни предметни области.

Интелектуализирането на Приложните Програмни Системи (ППС) е нов, следващ автоматизацията етап в развитието на тези системи.

Интелектуализираните ППС са с повишени възможности, по-удобни за потребителя и по-ефективни в сравнение с традиционните автоматизирани ППС. Това очевидно води до повишаване на производителността на труда и до пестене на

човешко време и сили.

в) Основни методи на изследването са:

- методи за представяне и обработка на символна информация (работа с низове и списъци, търсене по образец, работа с дървовидни структури и др.);

- методи за създаване на ППС (специфични методи на съответните предметни области, системен подход и системен анализ и др.);

- методи на Изкуствения интелект (ИИ) (представяне и обработка на знанията, логически извод, търсене в пространството на състоянията и др.);

- метод на предметно-ориентираните празни ИППС за генериране на проблемно-ориентирани ИППС;

- други.

г) Основни приноси:

1. Изграждане, обосноваване и развитие на обща концепция на ИППС.

2. Дефиниране и изследване на понятието предметно-ориентирана празна ИППС.

Изследване на възможността за използване на такива системи за генериране на проблемно-ориентирани ИППС.

3. Изследване и програмно експериментирание на концепциите, принципите и идеите на ИППС (във връзка със следващите по-долу т. т. 5, 6, 7, 8 и 9).

4. Задоволяване на някои нужди на практиката от интелектуализирани ППС в определени важни предметни области (във връзка със следващите по-долу т.т. 5,6,7,8 и 9).

5. Водещо участие в разработката на общата концепция, основните принципи и функционалната структура на ИППС за

мембранната биофизика. Научно ръководство на реализацията на тази система.

6. Водещо участие в разработката на общата концепция, основните принципи и функционалната структура на ИППС на Експертната Система за Оценка на Кадри (ЕСОК). Общо научно ръководство на реализацията на тази система.

7. Водещо участие в разработката на общата концепция, основните принципи и функционалната структура на Интелектуализирани Въпросно-Ответни Системи (ИВОС), на предметно-ориентирана празна ИВОС за анализ на историко-географски текстове и нейната конкретизация за такива текстове за гревна Тракия.

8. Създаване и разработка на обща концепция за Интелектуализирани Аналитично Изчислителни Системи (ИАИС).

Водещо участие в изследване и разработка на обща схема на ИАИС и на функционалната структура на нейните модули.

Научно ръководство на създаването на ИАИС за работа с матрици и на система за работа с бази от математически знания.

9. Създаване, изследване и разработване на концепция за ИППС за научно-технически задачи.

Разработка на общ проект за интелектуализиране на приложната програмна система за вариантен анализ на вентилационни системи "ВЕНТ" и нейното развитие до ИППС за вариантен анализ на вентилационни системи.

10. Научно ръководство на усвояването, адаптирането и подобряването на транслаторите TU LISP, LISP F3, Standard LISP и на системите за аналитични преобразования REDUCE-2 и SAC-2.

g) Внедряване на приносите в обществената практика.

- Вече осъществени внедрявания:

1. Транслатор TU LISP (LISP 1.6) е внегрен в ЛИИЦ при СУ "Климент Охридски" и в ЦЛАНП при БАН (стр.185 и стр.186).

2. Транслатор LISP F3 е внегрен в БСНИПИ "ИНТЕРПРОГРАМА" и в Института по техническа кибернетика при Словашката академия на науките (стр.188 и стр.189).

3. Транслатор LISP-A F3 за ЕИМ СМ-4 е внегрен в ИТК при Словашката АН (стр. 190).

4. Транслатор Standard LISP е внегрен в ЛИИЦ при СУ "Климент Охридски" (стр.192).

5. Системата за аналитични преобразования REDUCE-2 за ДДС ЕС и за ОС ЕС е внегрена в ЛИИЦ при СУ "Климент Охридски" (стр.187 и стр.192).

6. Системата за аналитични преобразования SAC-2 е внегрена в ЛИИЦ при СУ "Климент Охридски" (стр.194).

7. Аналитично-изчислителната система за работа с матрици е внегрена в катедра "Техническа механика" при ВМЕИ-Варна (стр.191).

8. Експертната система за електросливане на протопласти (съставна част на ИППС за мембранната биофизика) е внегрена в Централната лаборатория по биофизика при БАН (стр.193)

Общо внедряванията са 10.

- Предложения за бъдещи внедрявания.

Всички транслатори, споменати в т.т. 1-4 могат да бъдат внегрени и в други изчислителни центрове и научно-изследователски лаборатории.

Системите за аналитични преобразования, цитирани в т.т.

5-7 могат да бъдат внедрени в други изчислителни центрове, проектантски и научно-изследователски институти във връзка с научни и приложни изследвания в областта на математиката, физиката, техниката, геодезията, астрономията, космическите изследвания и др.

Експертната система, цитирана в т. 8, може да бъде внедрена и в други научно-изследователски и приложни лаборатории и институти, работещи в областта на биофизиката и биотехнологиите.

След завършване, експертната система за оценка на кадри ЕСОК може да бъде внедрена за определяне на деловите, професионалните и политическите качества на личности от предприятия, учреждения, обществено-политически организации и др. ЕСОК може да бъде внедрена и за анализиране на творческите възможности на надарени деца.

Предметно-ориентираната празна ИВОС за анализ на историко-географски текстове и нейната конкретизация за древна Тракия могат да бъдат внедрени в научноизследователската и учебната работа по история в съответни институти, учебни заведения, организации и др.

След реализация, общият проект на ИППС за вариантен анализ на вентилационни системи може да бъде внедрен в съответните проектантски организации и рудници за проектиране и управление на рудничната вентилация.

Разработеният метод за създаване на предметно-ориентирани празни ИППС и на тяхното конкретизиране е използван, както при внедрените ИППС, така и в предлаганите за бъдещо внедряване ИППС. Този метод може да се внедри в организациите, които разработват програмни продукти с

изкуствен интелект.

е) Технически данни на хабилитационния труд

Хабилитационният труд включва увод, седем глави, списък на употребените съкращения, списък на цитираната литература (съдържащ 89 заглавия) и документи за внедряване и справочник за използваните съкращения.

Брой страници 199.

Брой фигури - 18.

Брой внедрявания 10.

ж) Основните резултати на хабилитационния труд са публикувани в следните 13 научни статии (в скоби се дават номерата им в общия списък на литературата):

1. (16) Геров, А., Капитонова, Ю., Спиридонова, М., Томов, В. Интеллектуальные системы аналитических преобразований. В кн. "Представление знаний в человеко-машинных и робототехнических системах", том С "Прикладные человеко-машинные системы, ориентированные на знания". Рабочая группа ИВ КНВВТ, М., ВЦ АН СССР, ВИНТИ, 1984, стр. 112-136.
2. (27) Dimitrov, D., Tomov, V., Sharkov, G., Angelova, M. Liposome Production and Protoplast Electrofusion Expert Systems. In "Artificial Intelligence II: Methodology, Systems, Applications", Ph. Jorrand, V. Sgurev (editors), North - Holland, Amsterdam, 1987.
3. (67) Томов, В. Математическо осигуряване за пресмятане и анализ на потокоразпределението във вентилационни системи. Автореферат, София, 1975.
4. (69) Томов, В., Геров, А. Интеллектуальные прикладные программные системы. Доклад по приглашению, Первый

- научный семинар с международным участием "Проблемы и применения искусственного интеллекта", Варна, 21-25 сентября 1987, с. 58-66.
5. (70) Томов, В., Геров, А. Экспертни системи и приложения. В сд. "Математика и математическо образование. Доклади на XVI пролетна конференция на СМБ, Слънчев дряг, 6-9 април 1987 г.", София, Издателство на БАН, 1987, стр. 103-112.
6. (71) Томов, В., Геров, А., Григоров, А., Колчев, К. Предметно-ориентированные пустые интеллектуальные прикладные программные системы. Межд. конф. "Методы искусственного интеллекта и их применения", Пхенян, 1-11 октомври 1988 (под печат).
7. (72) Томов, В., Геров, А., Спиридонова, М. ЛИСПоподобни езици за програмиране. Сборник доклади на Националния семинар по информатика към СМБ, т. II (под печат).
8. (73) Томов, В., Геров, А., Спиридонова, М. Системи за аналитични преобразования. В сд. "Математика и математическо образование. Доклади на XIV пролетна конференция на СМБ, Слънчев дряг, 6-9 април 1985 г.", София, Издателство на БАН, 1985, стр. 92-103.
9. (74) Томов, В. и др. Представяне и използване на знанията. В "Отчетни материали за работата по договор No 29/25.03.1987г. между Институт по математика с ИЦ и КН за периода април - декември 1987 г. ".
10. (75) Tomov, V., Sahnó, St. Experimental System for Recognition and Classification of Archaeological Ceramics. In "Artificial Intelligence II: Methodo-

- logy, Systems, Applications", W. Bibel, B. Petkoff (editor), North-Holland, Amsterdam, 1985, pp.
11. (76) Tomov, V., Spiridonova, M., Gerov, A. Intelligent Computer Algebra Systems. Intern. conf. "Artificial Intelligence II: Methodology, Systems, Applications", ЦИТИ, Мш 170, 1987.
 12. (77) Tomov, V., Tacheva, M., Grigorov, A. ESIT - An Expert System for Analysis of Antique Historical - Geographical Texts about Ancient Thrace, In "Artificial Intelligence II: Methodology, Systems, Applications", Ph. Jorrand, V. Sgurev (editors), North-Holland, Amsterdam, 1987, pp. 235 - 242.
 13. (78) Томов, В., Шарков, Г. Интеллектуальные прикладные программные системы для мембранной биофизики. Седьмой семинар "Проблемы информатики и ее применения в управлении, обучении и научных исследованиях", 5-11 юни 1988 (под печат).

УВОД

В развитието на математическото осигуряване за ЕИМ могат да се посочат поне три, последователно появили се крупни направления:

а) начално създаване и разработване на основни концепции, структури, методи и реализации на математическото осигуряване;

б) автоматизация на математическото осигуряване;

в) интелектуализация на математическото осигуряване.

В направления а) и б) бяха постигнати и се постигат забележителни научни и приложни резултати.

Третото направление, което сега започва да се развива, по същество представлява внасяне (вносяне) на концепциите, методите, подходите и средствата на изкуствения интелект в математическото осигуряване. Такова математическо осигуряване, в което е внесен изкуствен интелект, наричаме интелектуализирано математическо осигуряване (ИМО). Очевидно, при изпълнение на задачите за които е предназначено, ИМО може да извършва някои интелектуални дейности. При това, ясно е, че ИМО е с по-големи възможности (при формулиране и решаване на задачите както в ширина, така и в дълбочина), а освен това е по-удобно и в някои случаи по-ефективно от традиционното математическо осигуряване.

Ако този подход на интелектуализиране бъде приложен към

приложна програмна система (ППС), то в резултат се получава интелектуализирана приложна програмна система (ИППС).

Очевидно, ИППС има характерните свойства и качества на ИМО.

В настоящия хабилитационен труд е направен опит за:

- изграждане на обща концепция на ИППС;
- дефиниране и изследване на предметно-ориентирани празни ИППС и тяхното използване за генериране на проблемно-ориентирани ИППС;

- изследване на възможностите за създаване на някои видове ИППС за различни значими предметни области.

Хабилитационният труд се състои от човод и седем глави.

В глава I са разгледани основните концепции на ИППС. Дадено е определение и е разгледана общата структура на тези системи. Разгледани са основните видове ИППС.

Въведено е понятието предметно-ориентирана празна ИППС и е показано как от нея, чрез конкретизация могат да се генерират проблемно-ориентирани ИППС.

Глава II е посветена на един от фундаменталните въпроси на изкуствения интелект - представяне и обработка на знанията.

Разгледани са основните типове знания и основните схеми за представяне на знания - предикатна логика от I ред, представяне чрез процедури, семантични мрежи, системи продукции и фреймове.

Въведено е понятието семантичен елемент и е посочено неговото място и предназначение в общата структура на базата знания.

В глава III са разгледани най-важните инструментални

средства на изкуствения интелект (които са инструментални средства и на ИППС) – езикът ЛИСП и неговите диалекти, езикът ПРОЛОГ и езикът FRL за представяне на знанията чрез фреймове и работа с тях. Разгледани са и различни транслатори от ЛИСП, внедрени у нас.

Глави от IV до VII на хабилитационния труд са посветени на някои ИППС. При създаването на тези ИППС са преследвани следните две цели:

а) Изследване и програмно експериментиране на концепциите, принципите и идеите на ИППС.

б) Задоволяване на някои нужди на практиката от интелектуализирани ППС в определени важни предметни области.

В глава IV експертните системи се разглеждат като ИППС.

Представена е като ИППС завършената ЕС за мембранната биофизика.

Разгледана е като ИППС експертната система ЕСОК за оценка на кадри, която е в процес на разработка.

В глава V се разглежда друг вид ИППС, а именно интелектуализирани въпросно-ответни системи (ИВОС).

Представена е предметно-ориентирана празна ИВОС за анализ на историко-географски текстове.

Разгледана е също така и проблемно-ориентирана ИВОС за географски и логически анализ на историко-географски текстове за древна Тракия, която е в процес на завършване.

Глава VI е посветена на интелектуализираните аналитично-изчислителни системи (ИАИС). Представена и анализирана е общата схема на такива системи.

Разгледани и изследвани са специализираните Аналитично-Изчислителни системи (АИС) и са представени

редица специализирани за различни области АИС. Показан е ефективен подход за генериране на специализирани ИАИС, състоящ се в конкретизация за различни проблемни области на предметно-ориентирана празна ИАИС.

Представена е ИАИС за работа с матрици и система за работа с бази от математически знания.

Глава VII е посветена на интелектуализираните приложни програмни системи за научно-технически задачи (ИППС за НТЗ).

Тази глава съдържа някои идеи и разработки от кандидатската дисертация на автора и тяхното продължение в духа на ИППС.

Разработена и представена е концепция на ИППС за НТЗ.

Разработен и представен е общ проект за интелектуализиране на приложната програмна система за вариантен анализ на вентилационни системи "ВЕНТ" и нейното развитие до ИППС за вариантен анализ на вентилационни системи.

Представените в хабилитационния труд резултати са получени от автора самостоятелно и с участието на други специалисти (в продължение на 10 години) в сектор "Математическо осигуряване", а след това в сектор "Изкуствен интелест" на ЕЦММ при БАН във връзка с изпълнение на планови задачи на Институт по математика с ИЦ и на някои договори, свързани с тях.

Част от резултатите, представени в труда, са внедрени в различни организации в страната. Внедрени са:

- транслаторите от TU LISP (LISP 1.6), LISP F3, LISP-A F3 и Standard Lisp и системите за аналитични преобразования REDUCE 2 и SAC-2;

- ИППС за мембранната биофизика;
- ИАИС за работа с матрици.

Съществена част от разработените концепции, принципи и основни идеи са дело на автора.

Различните системи, представени в реабилитационния труд, са разработвани под научното ръководство на автора от сътрудниците на сектор "Изкуствен интелект" на ЕЦММ при БАН. Те са: н.с. кми Александър Геров, н.с. Маргарита Спиридонова, н.с. Александър Григоров, н.с. Мария Нишева, н.с. Георги Шарков, асп. Камен Колчев и др.

В разработките, имащи интердисциплинарен характер, са участвували специалисти от съответните предметни (проблемни) области. Те са: проф. дин Маргарита Тачева от Историческия факултет на СУ "Климент Охридски", проф. ктн Тодор Стефанов от ВМГИ-София, доц. кфн Паньо Данев - инструктор в отдел "Организационен" в ЦК на БКП, ст. н. с. дбн Димитър Димитров от Централната лаборатория по биофизика при БАН, ст.н.с. кми Тома Тонев от ЕЦММ при БАН и др.

Изложението е изградено въз основа на 13 научни публикации на автора, като са използвани и научни отчети по теми и документация на математическо осигуряване.

За да не се утежнява изложението, в текста са включени главно основни концепции, принципи, идеи и съображения за разглежданите разработки. Подробни описания на тези разработки могат да се намерят в съответната потребителска и системна документация, в отчетите на сектор "Изкуствен интелект" и "Математическо осигуряване" и в някои вътрешни за ИМ с ИЦ при БАН издания.

БЛАГОДАРНОСТИ

Създаването на настоящия хабилитационен труд стана възможно благодарение на съдействието и поддръжката на редица организации, колеги и приятели.

Авторът изказва голяма благодарност на Ръководството на ЕЦММ при БАН за оказаното съдействие, поддръжка и за осигурените прекрасни условия за работа. По същите причини съм благодарен и на секторите "Математическо осигуряване" и "Изкуствен интелект".

Не е възможно да спомена всички, които са ми оказвали благотворно влияние и подкрепа. Но има някои имена, които не мога да отмина!

На първо място трябва да изкажа голямата си благодарност към моя учител и ръководител академик Любомир Илиев за изключително ценните напътствия и непрекъснатото съдействие в цялостната ми научна, служебна и обществена дейност.

Много съм благодарен и на академик Благовест Сенгов за голямата подкрепа и доброжелателно отношение още от началото на работата ми по проблемите на математическото осигуряване и изкуствения интелект.

Изказвам сърдечна благодарност на проф. кфмн Петър Бърнев за голямата поддръжка и съдействие, за плодотворните научни обсъждания, за интересните и дълбоки идеи и за това,

че той пръв ми помогна да навляза в информатиката.

Авторът изказва голяма благодарност на всички членове на сектор "Изкуствен интелект", а именно: н.с. кми Ал. Геров, н.с. М. Спиридонова, н.с. М. Нишева, н.с. Ал. Григоров, н.с. Г. Шарков, асп. К. Колчев, кми В. Вълев, мат. Е. Дечева и В. Младенова, без самоотвержената и безкористна помощ на които не би било възможно създаването на настоящия труд.

Специална благодарност дължа на н.с. кми Ал. Геров за плодотворните научни обсъждания и за неговото научно и техническо съдействие по създаването на настоящия хабилитационен труд.

За техническото оформяне на труда голяма заслуга има В. Младенова, за което съм ѝ много благодарен.

Искам да отбележа с искрена благодарност и сътрудниците на сектор "Математическо осигуряване": ст.н.с. кми М. Бърнева, ст.н.с. кми П. Станчев, ст.н.с. кми Вл. Занев, ст.н.с. кми П. Азълков, н.с. кми П. Петров, н.с. М. Шишкова, Л. Здравкова и др.

Авторът се чувства приятно задължен да изрази своята голяма благодарност за активното и компетентно участие в интердисциплинарните разработки от хабилитационния труд на следните специалисти от различни предметни области и организации: проф. дин Маргарита Тачева от Историческия факултет на СУ "Климент Охридски", проф. кти Тодор Стефанов от ВМГИ-София, доц. кфи Паньо Данев - инструктор в отдел "Организационен" на ЦК на БКП, ст.н.с. дби Димитър Димитров от Централната лаборатория по биофизика при БАН, ст.н.с. кми Тома Тонев от ЕЦММ при БАН, доц. кфми Стоян Бъчваров от

ЛИИЦ при СУ "Климент Охридски" и др.

За оказаното съдействие авторът благодари и на ЛИИЦ при СУ "Климент Охридски", ВМГИ-София, БСНИПИ "ИНТЕРПРОГРАМА", ИТКР при БАН и др.

ГЛАВА I. ОСНОВНИ КОНЦЕПЦИИ НА ИНТЕЛЕКТУАЛИЗИРАНИТЕ ПРИЛОЖНИ ПРОГРАМНИ СИСТЕМИ

1. ОПРЕДЕЛЕНИЕ И ОБЩА СТРУКТУРА НА ИНТЕЛЕКТУАЛИЗИРАНИТЕ ПРИЛОЖНИ ПРОГРАМНИ СИСТЕМИ

Програмна система, предназначена за решаване на задачи от конкретна предметна област, обикновено се нарича приложна програмна система (ППС). Голямото значение на този вид системи обуславя интереса не само към тяхното автоматизиране, но също така и към тяхното интелектуализиране.

Към това трябва да се отбележи, че в последните 10-15 години интензивно се развива едно ново направление в информатиката, което се нарича "изкуствен интелект" (ИИ). То има за цел създаването на концепции, методи, алгоритми и съответни програмни и апаратни средства, чрез които би могло да се постигне автоматите (напр. ЕИМ, роботи и др.) да извършват интелектуална дейност или поне близка до нея.

Някои основни направления в ИИ са следните: търсене в пространството на състоянията, представяне и използване на знания, планиране на действията, експертни системи, разпознаване на образи, общуване с ЕИМ на език, близък до естествения.

Интелектуализираните приложни програмни системи (ИППС)

[Томов, Геров 87] представляват ППС, които с помощта на методите и средствата на изкуствения интелект са доразвити с цел подобряване на техните възможности и повишаване на удобствата за работа с тях. Тези системи в една или друга степен могат да осъществяват някои интелектуални дейности.

Ще отбележим някои основни характеристики на ИППС:

- наличие на интерфейс "човек-машина" на език, близък до естествения;
- използване на знания;
- логически извод;
- планиране на действията;
- обучение и самообучение;
- прилагане методите на изкуствения интелект в алгоритмите от проблемната област на съответната ИППС;
- визуален и звуков вход и изход.

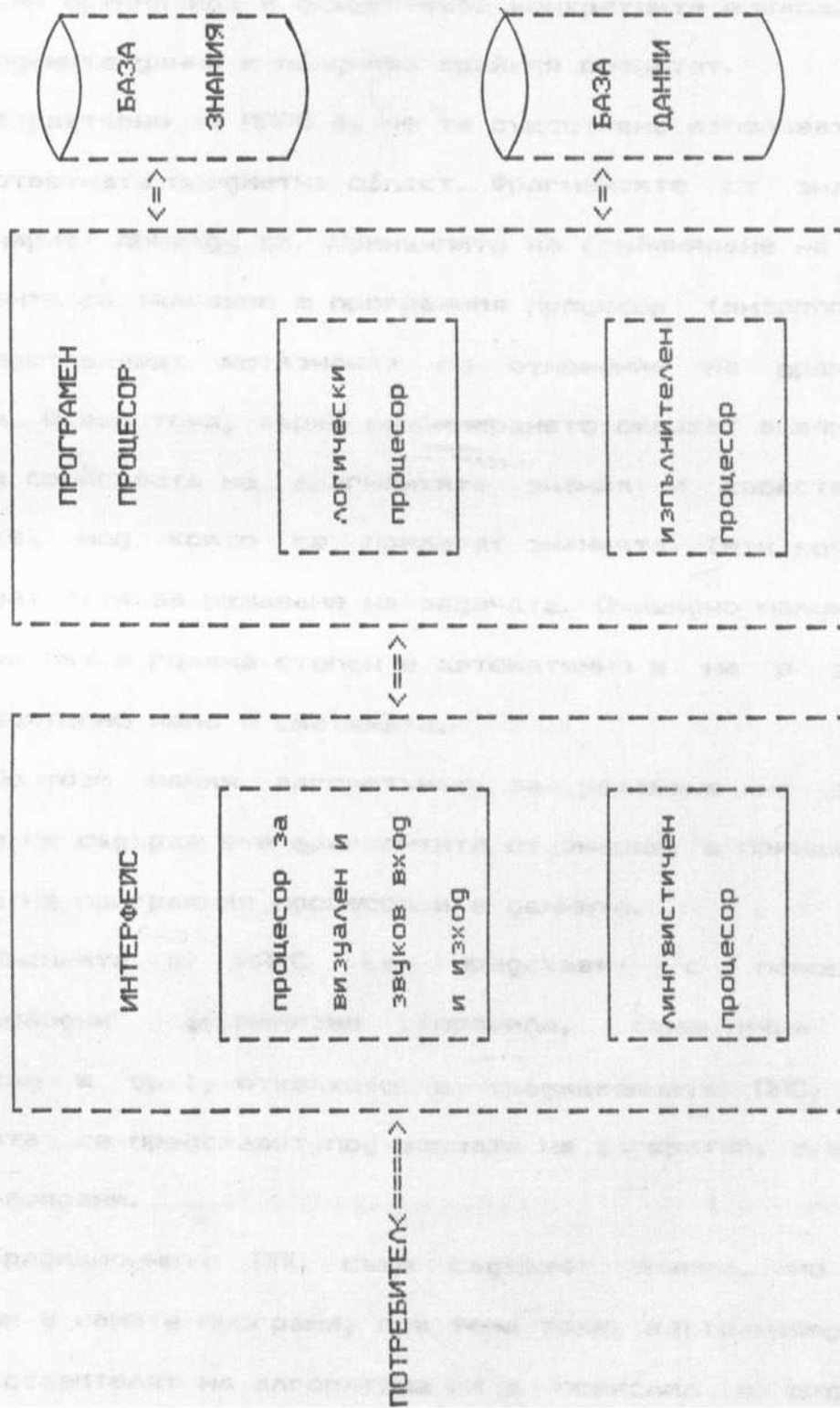
Разбира се, в различните конкретни ИППС в зависимост от тяхното естество, някои от тези характеристики могат да бъдат по-силно развити, отколкото други такива, които могат дори да отсъствуват.

На фиг. 1. е дадена общата схема на ИППС.

В блока "интерфейс" с помощта на съответен процесор се осъществяват визуалният и звуков вход и изход и интерпретацията на фрази на език, близък до естествения.

Програмният процесор осъществява логическите и изчислителните функции на системата. Той съществено използва базата знания (БЗ) и базата данни (БД).

Използвайки знанията на системата за проблемната област, логическият процесор реализира логически извод, самостоятелно избира път (метод) за решение на задачата



Фиг. 1. Обща схема на ИППС.

(когато това е възможно), осъществява самоусъвършенствуване на системата и т.н.

Използвайки наличните алгоритми и самостоятелно съставените от системата планове за действие, изпълнителният процесор организира и осъществява конкретните изчисления над необходимите данни и получава крайния резултат.

Характерно за ИППС е, че те съществено използват знания за съответната предметна област. Фрагментите от знания се комбинират помежду си. Принципите на комбиниране на отделни фрагменти са заложи в програмния процесор (интерпретатор) и представляват метазнания по отношение на фрагментите знания. Освен това, върху комбинирането оказват влияние също така и свойствата на фрагментите знания и свойствата на данните, над които се прилагат знанията. Тези комбинации трасират пътя за решаване на задачата. Очевидно получаването на този път в голяма степен е автоматично и не е заложило предварително явно в системата.

По този начин алгоритъмът за решаване на задачата неявно се съдържа във фрагментите от знания, в принципите на работа на програмния процесор и в данните.

Знанията в ИППС се представят с помощта на "по-свободни" формализми (правила, семантични мрежи, фреймове и др.), отколкото в традиционните ППС, където знанията се представят под формата на алгоритми, т.е. силно формализирани.

Традиционните ППС също съдържат знания, но те са вложени в самите програми, при това така, както програмистът или съставителят на алгоритъма ги е осмислил и отразил в програмата. С други думи, знанията се осмислят и се

използват в такава степен, в каквато програмистът е успял да ги запрограмира.

Фактът, че в ИППС знанията в голяма степен са отделени от програмите, позволява не само въвеждане на знания от специалист в предметната област, но също и лесна актуализация, коригиране и допълване на знанията без изменение на програмната част.

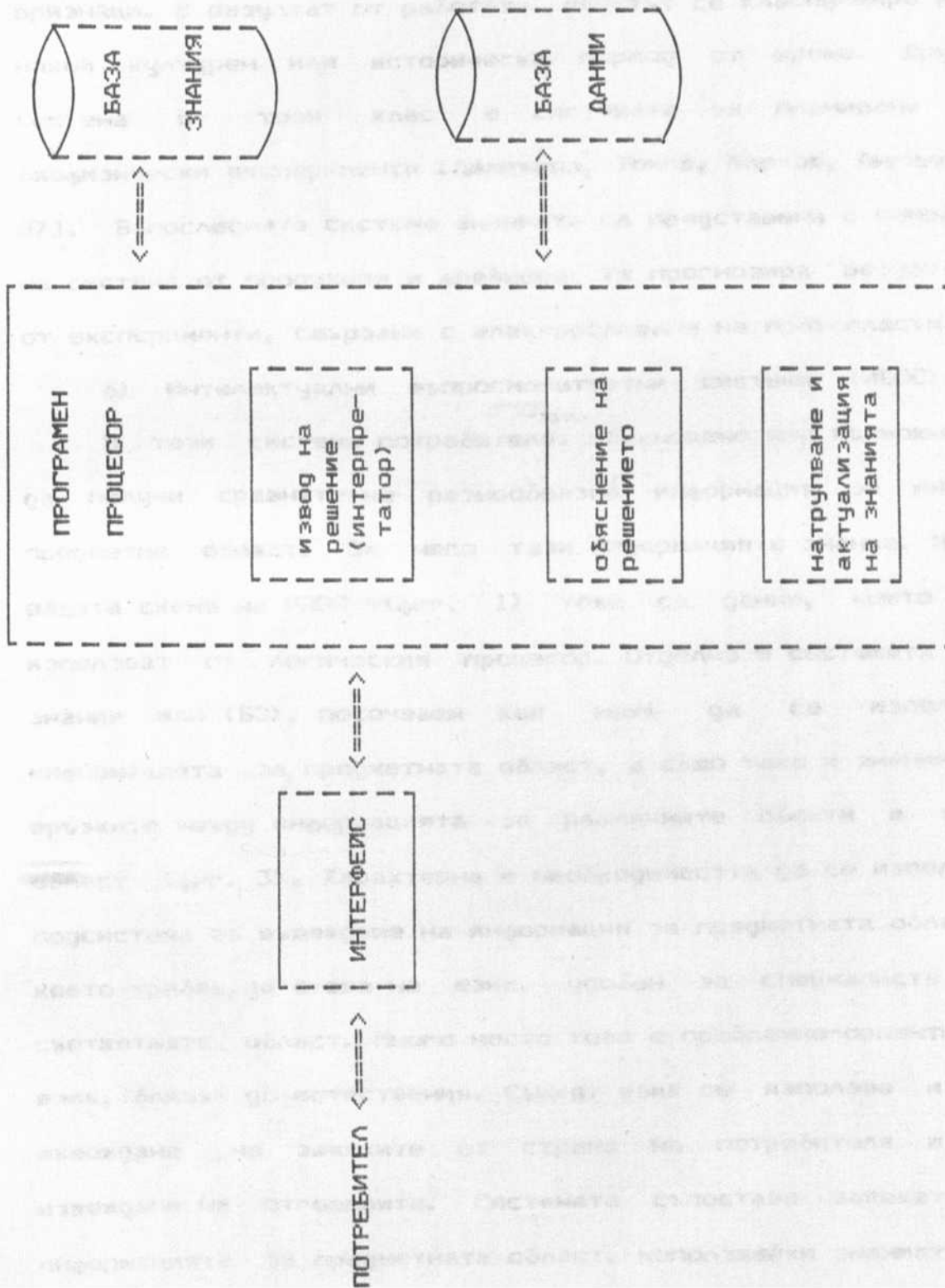
2. ОСНОВНИ ВИДОВЕ ИППС

Ще разгледаме някои основни видове ИППС и техни представители, разработени в ЕЦММ при БАН.

а) Експертни системи (ЕС).

Това е такъв клас ИППС, в който се използват експертни знания за определена проблемна област и които изпълняват ролята на експерт, съветник или даже преподавател в съответната област. Общата схема на ЕС е дадена на фиг. 2 [Томов, Геров 87а]. Характерно е това, че логическият и изпълнителният процесори са обединени в обща програмна част – интерпретатор или подсистема за извод на решение. От изброените характеристики задължителни се явяват използването на знания и логическият извод. Интерфейсът "човек-машина" трябва да включва поне някои елементи от естествения език. В развитите ЕС има и отделна подсистема за натрупване на знания, което позволява в известна степен да се говори за обучение и/или самообучение на системата.

Представител на този клас системи се явява ЕС за разпознаване на археологическа керамика [Томов, Сахно 85]. Системата използва археологически знания, представени с

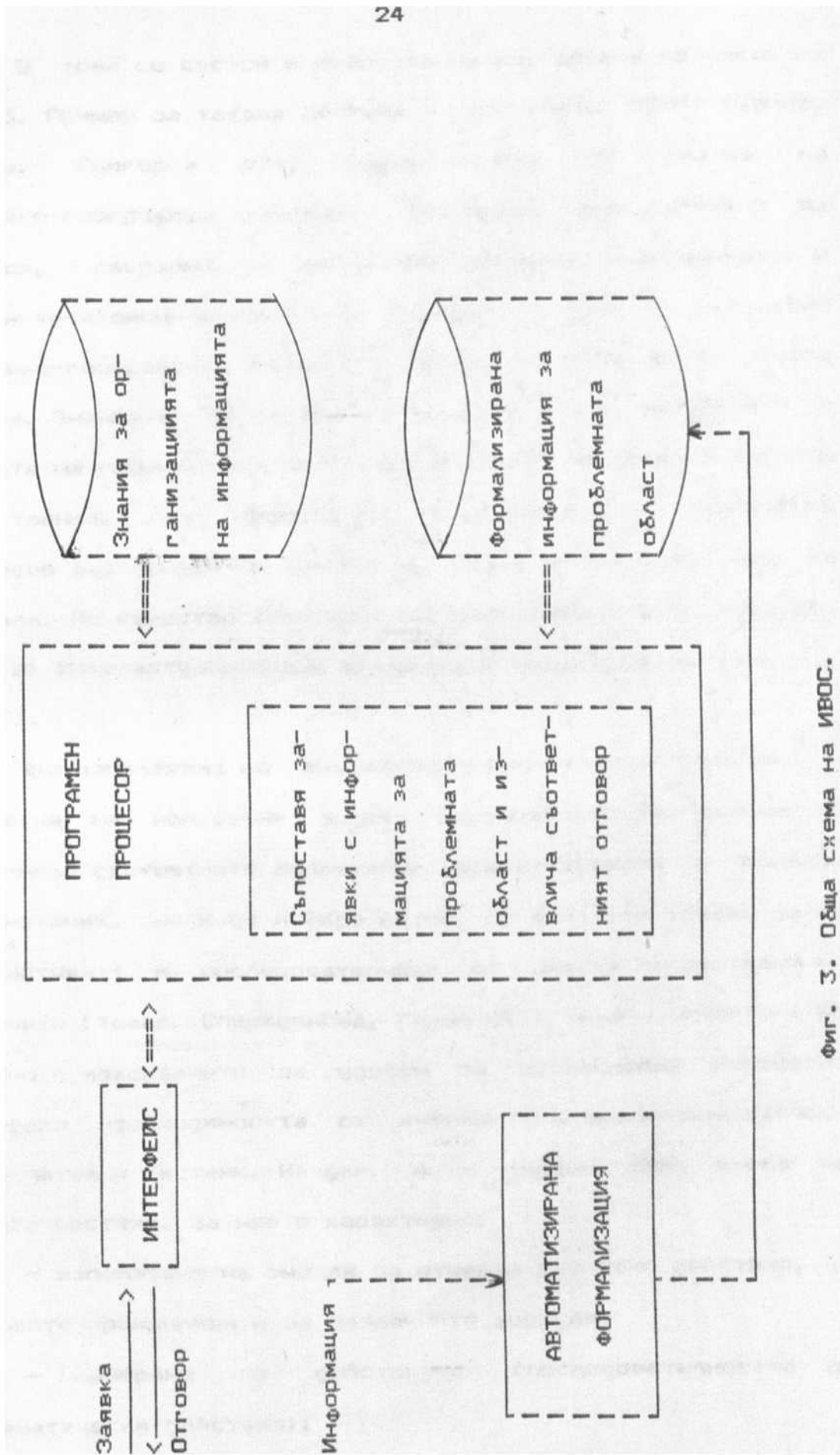


Фиг. 2. Обща схема на ЕС.

помощта на правила. Броят им е над 100. На основата на въведени признаци за конкретния обект се изгражда хипотеза, а за нейното потвърждаване при необходимост се задават въпроси към потребителя за стойностите на някои допълнителни признаци. В резултат от работата обектът се класифицира към някой културен или исторически период от време. Друга система от този клас е системата за планиране на биофизически експерименти [Димитров, Томов, Шарков, Ангелова 87]. В последната система знанията са представени с помощта на система от продукции и фреймове. Тя прогнозира резултата от експерименти, свързани с електросливане на протопласти.

δ) Интелектуални въпросно-ответни системи (ИВОС).

В тези системи потребителят обикновено има възможност да получи сравнително разнообразна информация за някоя предметна област. За него тази информация е знание. Но в общата схема на ИППС (фиг. 1) това са данни, които се използват от логическия процесор. Отделно в системата има знания или (БЗ), посочващи как може да се използва информацията за предметната област, а също така и знания за връзките между информацията за различните обекти в тази област (фиг. 3). Характерна е необходимостта да се използва подсистема за въвеждане на информация за предметната област, което трябва да става на език, удобен за специалиста от съответната област. Много често това е проблемно-ориентиран език, близък до естествения. Същият език се използва и за въвеждане на заявките от страна на потребителя и за извеждане на отговорите. Системата съпоставя заявката с информацията за предметната област, използвайки знанията за връзките и организацията на тази информация.



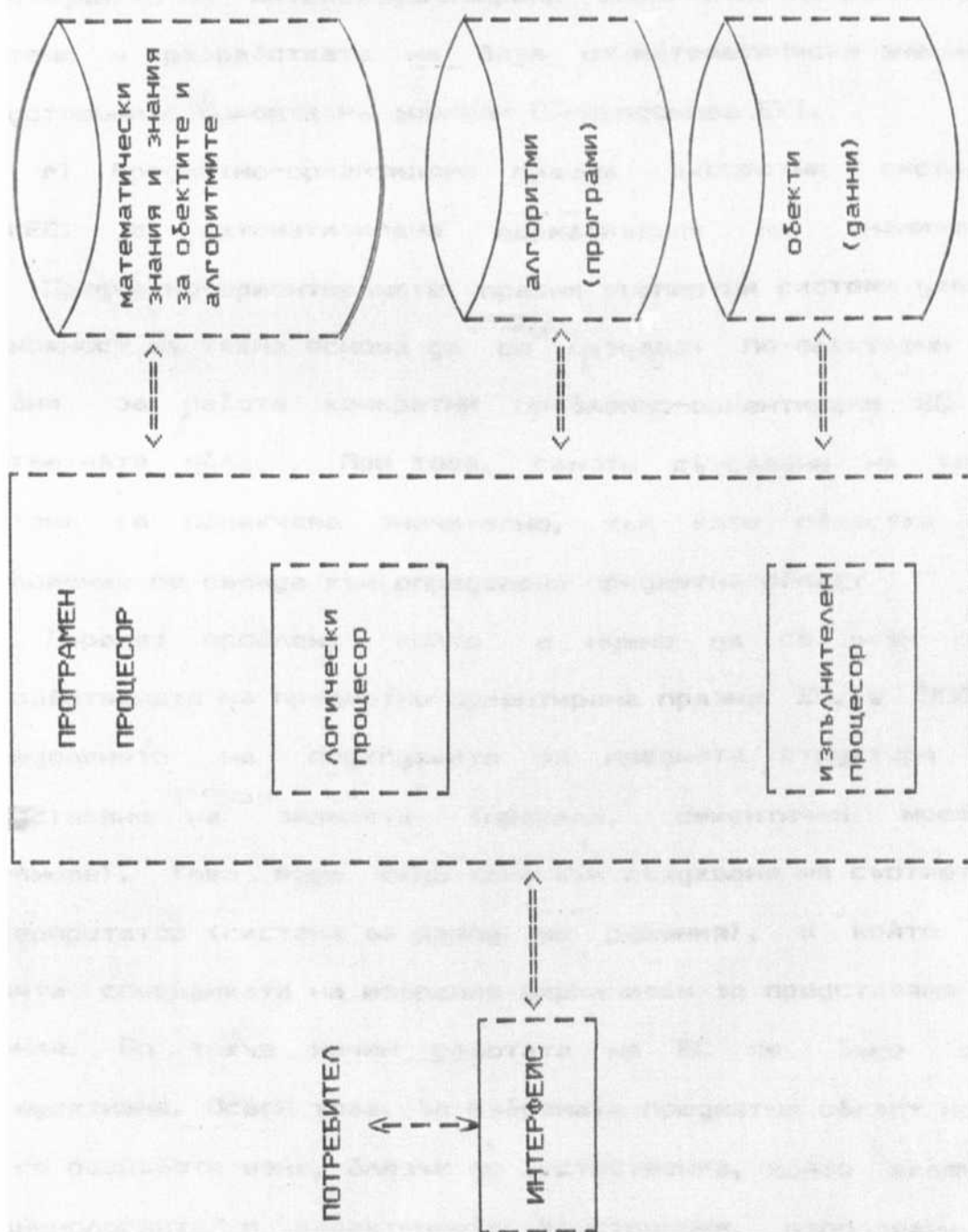
Фиг. 3. Обща схема на ИВОС.

В това се състои и работата на програмния процесор на фиг. 3. Пример за такава система е системата ЕСИТ [Томов, Тачева, Григоров 87], предназначена за анализ на историко-географски текстове. Системата дава отговор на въпроси, свързани с местоположението, населението и стопанско-икономическото значение на различни историко-географски обекти въз основа на текстове от гревни автори. Знанията за предметната област се представят с помощта на семантични мрежи и се въвеждат на език, близък до естествения. Тази информация се използва от програмния процесор въз основа на знанията, представени във вид на правила. По същество тази част на системата представлява ЕС.

в) Интелектуализирани аналитично-изчислителни системи. (ИАИС).

При използване на аналитично-изчислителни системи за решаване на конкретни задачи системата трябва не само да изпълнява съответните аналитични преобразования и числени пресмятания, но и да избира метод, по който те трябва да се осъществяват и последователност от стъпки за решаване на задачите [Томов, Спиридонова, Геров 84]. Това обстоятелство заедно с изискването за удобен за потребителя интерфейс определя необходимостта от интелектуализирани аналитично-изчислителни системи. На фиг. 4 е дадена обща схема на такава система. За нея е характерно:

- използване на знания за отделни възможни действия, за типовете променливи и за различните формули;
- планиране на действията (последователността от математически действия);
- прилагане на методите на изкуствения интелект в някои



Фиг. 4. Обща схема на ИИС.

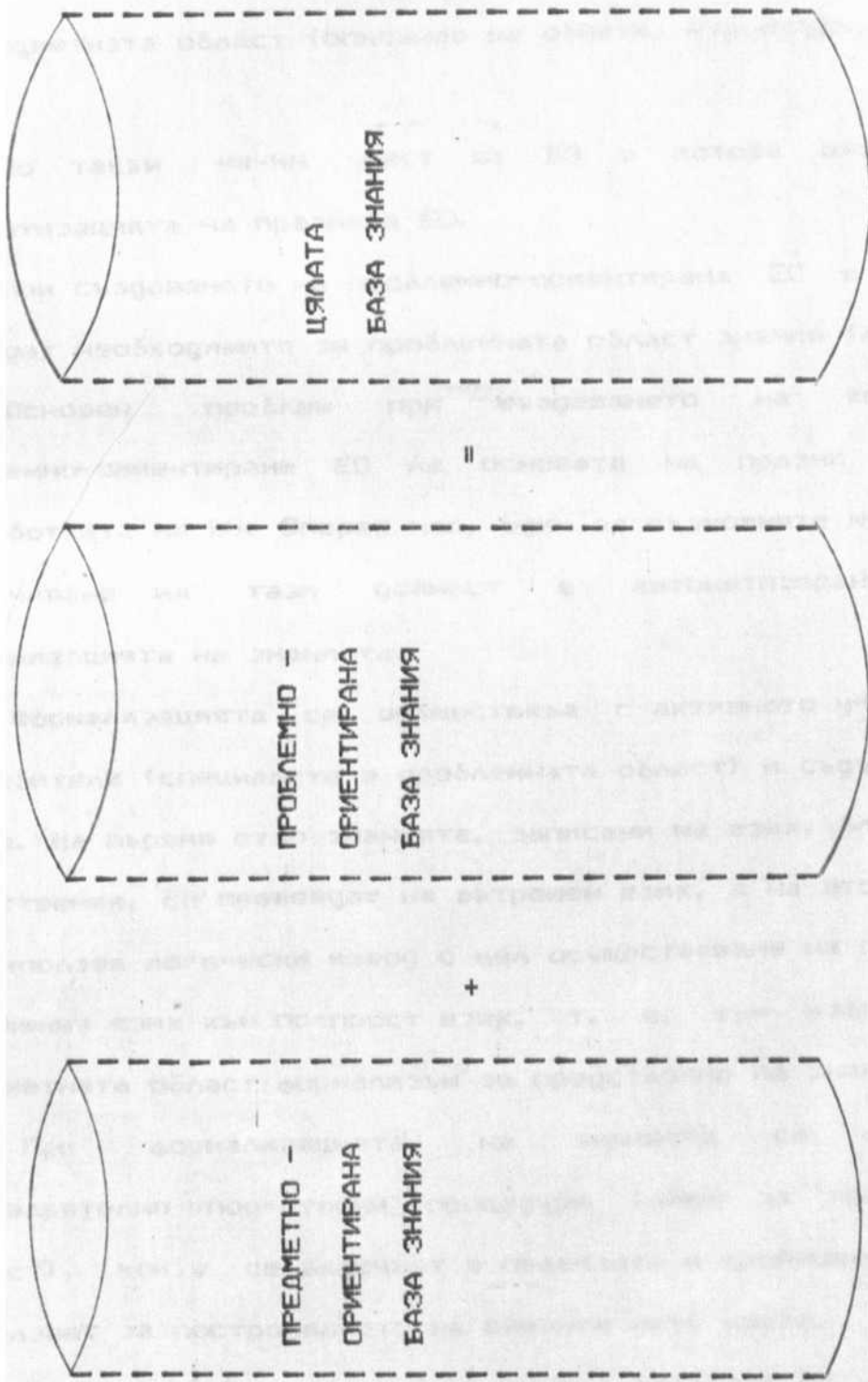
от алгоритмите.

Конкретен пример е системата за работа с матрици [Геров 87], в която на основа на знанията за отделните алгоритми и за типовете математически обекти, с които системата може да работи, се осъществява предварително планиране на действията на изпълнителния процесор. Друг важен елемент при изграждането на интелектуализирани аналитично-изчислителни системи е разработката на база от математически знания, представени с помощта на формули [Спиридонова 87].

г) Предметно-ориентирани празни експертни системи (ПОПЕС) с автоматизирана формализация на знанията.

Предметно-ориентираните празни експертни системи дават възможност на тяхна основа да се създават по-ефективни и удобни за работа конкретни проблемно-ориентирани ЕС в съответната област. При това, самото създаване на тези системи се облекчава значително, тъй като областта на приложение се свежда към определена предметна област.

Първият проблем, който е нужно да се реши при разработването на предметно-ориентирана празна ЕС, е ПОПЕС определянето на подходящите за предмета структури за представяне на знанията (правила, семантични мрежи, фреймове). Това води също така към създаване на съответен интерпретатор (система за извод на решения), в който се отчита спецификата на избрания формализъм за представяне на знания. По такъв начин работата на ЕС ще бъде още по-ефективна. Освен това, за избраната предметна област може да се разработи език, близък до естествения, който включва терминологията и характерните конструкции, използвани в предметната област.



Фиг. 5. Обща структура на цялата база знания.

В празната експертна система могат да се разработят и използват общи процедури, които съответствуват на избрания предмет и са общозначими в тази предметна област.

БЗ на предметно-ориентираната празна ЕС на практика не е абсолютно празна. В нея се въвеждат основните общи знания за предметната област (описание на обекти, структури, методи и др.).

По такъв начин част от БЗ е готова още преди конкретизацията на празната ЕС.

При създаването на проблемно-ориентирана ЕС в БЗ се въвеждат необходимите за проблемната област знания (фиг. 5).

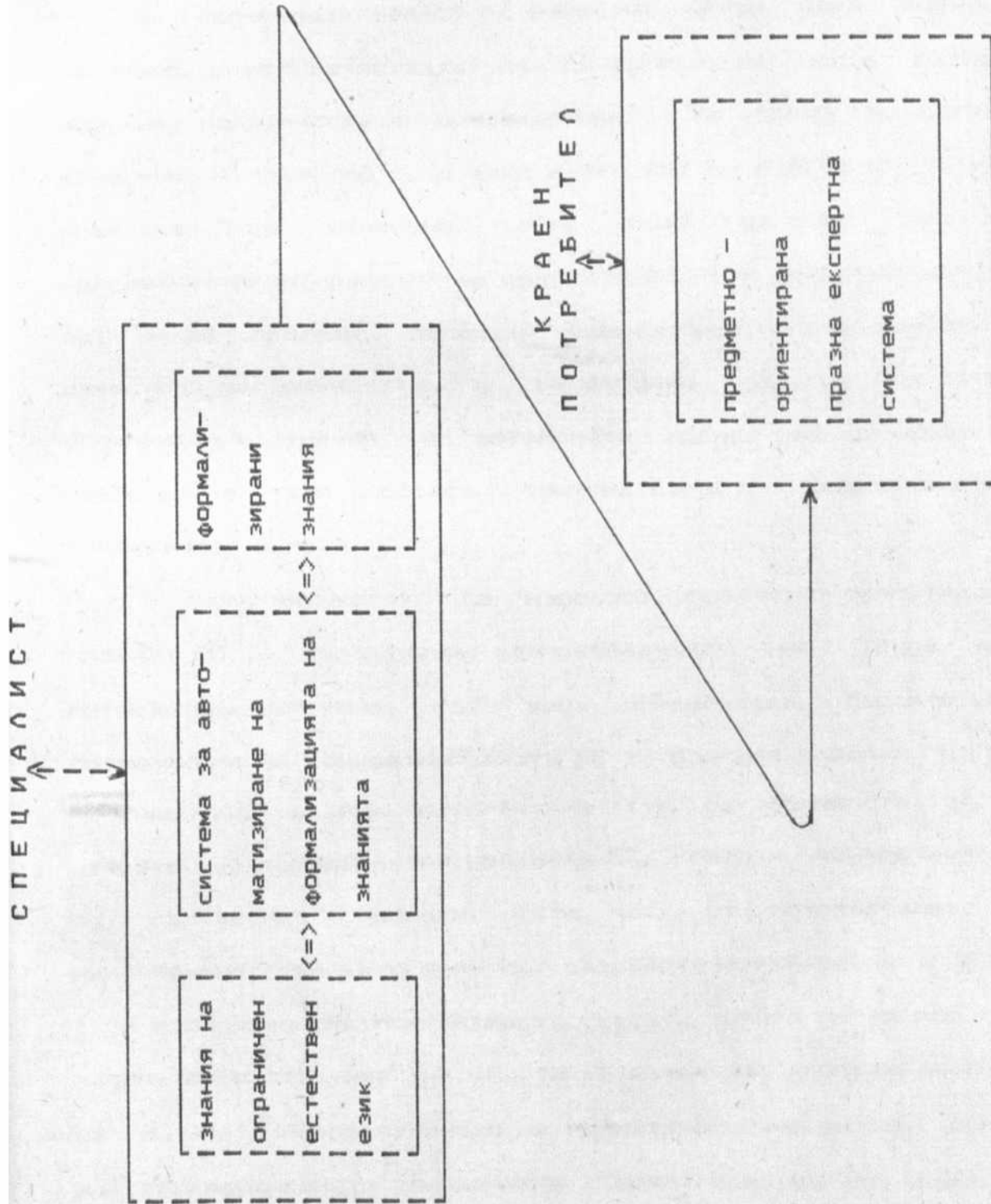
Основен проблем при създаването на конкретни проблемно-ориентирани ЕС на основата на празни ЕС е разработката на БЗ. Според нас, един от възможните методи за облекчаване на тази дейност е автоматизирането на формализацията на знанията.

Формализацията се осъществява с активното участие на потребителя (специалиста в проблемната област) и съдържа два етапа. На първия етап знанията, записани на език, близък до естествения, се превеждат на вътрешен език, а на втория етап се използва логически извод с цел осъществяване на преход от вътрешния език към по-прост език, т. е. към избрания за предметната област формализъм за представяне на знания.

При формализацията на знанията се използват предварително подготвени процедури (обща за предметната област), които се включват в правилата и фреймовете или се използват за построяването на семантичната мрежа.

Общата схема на такава система е показана на фиг. 6.

Ще илюстрираме тази концепция с един пример. Нека



Фиг. 6. Обща схема на предметно-ориентирана празна експертна система (ПОПЕС) с възможност за автоматизиране на формализацията на знанията.

предметната област е оценка на качествата на личности и колективи. В тази област могат да се отделят редица проблеми. Например, оценка на трудовите качества, оценка на организационните качества на личността, оценка на качествата на надарени деца в дадена област на изкуството и др.

За предметната област са известни някои общи знания. Например, че качествата на личността са преди всичко морални, политически и професионални. За всяко конкретно качество е известно също така и към коя от изброените групи може да бъде отнесено, т.е. може да се построи предметно-ориентирана БЗ на празната ЕС. При конкретизацията за някой проблем, например оценка на организационните качества на личността е необходимо да се въведат допълнителни знания за детайлната оценка на най-важните качества в тази област (инициативност, оперативност, контактност и пр.).

В случая, когато се използва предметно-ориентирана празна ЕС, е характерно разпределението на труда на потребителите. Има два вида потребители. Първите са потребители на конкретизираната ЕС за дадения проблем – ще ги наричаме крайни потребители (те са болшинството), а вторите – потребители на празната ЕС, които я конкретизират за съответния проблем. Тази част от потребителите е малочислена, тъй като един път разработена конкретната ЕС ще бъде използвана на всички места, където трябва да се решават задачи от съответния тип. По такъв начин за конкретизацията на ЕС и за разработването на нейната БЗ може да се съберат най-квалифицираните специалисти и може лесно да се реагира на измененията в концепциите за предметната област, на

измененията на нормативната база и т. н. Във връзка с това БЗ може да бъде актуализирана централизирано и актуализираната ЕС да стигне бързо до своите крайни потребители.

В посочения пример критериите за оценка са знания от глобален характер, без да се влияят от конкретните данни. Това повишава нивото на обективност и еднаквост на оценката. В заключение ще кажем, че предлаганият метод представлява разумен компромис между универсални и специализирани системи.

г) Интелектуализирани приложни програмни системи, разработени въз основа на вече създадени програмни системи.

Съвсем накратко може да се каже, че в редица случаи е възможно да се настрои вече създадена система с някои допълнителни подсистеми, даващи възможност за интелектуализиране на нейната работа. Тези подсистеми, като правило осъществяват вход и изход, но в някои случаи дават възможност за построяване на план от настроена логическа подсистема, който впоследствие ще бъде изпълнен от основната система.

3. ПРЕДМЕТНО-ОРИЕНТИРАНИ ПРАЗНИ ИППС. ПРОБЛЕМНО-ОРИЕНТИРАНИ ИППС.

Разработването на ИППС за решаване на задачи в дадена проблемна област е сложен и труден процес, изискващ не малко усилия и време както на специалисти по изкуствен интелект, така и на специалисти в съответната област. Този процес обаче може да се облекчи и отчасти автоматизира, когато

трябва да се създават няколко ИППС в различни, но сродни проблемни области, т.е. проблемни области, които са части от една по-обща предметна област [Томов, Геров, Григоров, Колчев 88]. В този случай, знанията, които са необходими за решаването на задачи от дадена проблемна област, включват знания, общовалидни в цялата предметна област и специфични знания за проблемната област. Следователно, може да се създаде предметно-ориентирана празна ИППС, която има знания, общовалидни за предметната област. След това тя може да се конкретизира за различни проблемни области чрез въвеждане на съответните специфични знания и да се получат различни проблемно-ориентирани ИППС. Този подход може да се използва както за създаване на предметно-ориентирани празни ЕС (това описахме в предната точка), така и за създаване на предметно-ориентирани празни ИВОС и т.н., и е валиден не само при ИППС, но и изобщо – за системи, основани на знания.

Предметно-ориентираните празни ИППС не са абсолютно "празни" системи, тъй като отразяват спецификата на предметната област, т.е. съдържат знания, формализми, алгоритми и др. за тази област.

Един от най-важните проблеми при създаването на предметно-ориентирани празни ИППС е определянето на подходящи структури (правила, фреймове, семантични мрежи и др.) за представяне на знанията в предметната област. Освен това трябва да се разработят съответни алгоритми и процедури за интерпретиране на знанията, отчитайки спецификата на избрания формализъм за представяне на знанията, т.е. да се създаде програмен процесор, който да работи достатъчно ефективно при решаването на задачи от предметната област.

Трябва да се проектира и разработи и интерфейсът с потребителя. За дадената предметна област може да се разработи език, близък до естествения, който включва терминология и характерни конструкции от тази област, и да се създаде съответен лингвистичен процесор. Освен това в БЗ трябва да се въведат и общовалидните знания за предметната област. По този начин, част от БЗ ще бъде готова още преди конкретизацията на празната ИППС.

Конкретизацията на предметно-ориентираната празна ИППС става чрез въвеждането на специфични знания за проблемната област. Възможно е въвеждането да става на ограничен естествен език, като се извършва автоматична формализация на знанията [Томов, Геров 87].

буквалността на даден модел за представяне на знания за
определени задачи. От друга страна, съдържанието на системите,
разполагащи с всяко полезна база знания и по-късни алгоритми
ГЛАВА II. ПРЕДСТАВЯНЕ И ОБРАБОТКА НА ЗНАНИЯТА.

друга, не претърпява важна задача на изкуствения интелект
и, следователно, една база на съдържанието модел на

1. ОБЩИ СВЕДЕНИЯ Поради това, изкуственият интелект
има да се определи не само като наука за използване на

В нашето изложение под понятието знания ще разбираме
такава обща, относително неизменяема и адекватна на
действителността информационна структура, която описва
класове обекти, процеси, отношения, закони и др. [Томов и
др. 87].

Изкуственият интелект, като наука, има за цел
създаването на системи, извършващи дейности, считани за
интелектуални или близки до интелектуалните. Необходимо
условие за извършването на такива дейности е системата (била
тя човек или машина) да притежава определени знания. Това
обуславя особено важното място, което заемат представянето и
използването (прилагането) на знания в системите с изкуствен
интелект. Счита се, че в момента представянето на знания е
най-бързо развиващата се област на ИИ [Бар, Фегенбаум 81].
Въпреки, че човешкото мислене и човешкият интелект са обект
на изучаване от самото начало на човешката култура, до днес
все още няма създадена единна и стройна теория на мисленето
и на интелекта. От една страна, този факт създава сериозни
трудности пред изследователите в областта на представянето и
използването на знания, тъй като поради липсата на единна
теория е трудно или невъзможно да се определи предварително

ефективността на даден модел за представяне на знания за определена задача. От друга страна, създаването на системи, разполагащи с все по-големи бази знания и по-сложни апарати за тяхното представяне и извличане, води до разкриването на друга, не по-малко важна задача на изкуствения интелект – изследването (на базата на създадените модели) на естествения интелект. Поради това, изкуственият интелект може да се определи не само като наука за създаване на интелигентни системи (интелигентни технологии – [Бар, Фегенбаум 81]), но и за изследване на технологията на интелекта.

Интелектуализираните програмни системи (някъде ги наричат още системи с изкуствен интелект) работят на базата на три неделими части – данни, знания и програми за обработка на знанията и данните. Тези три части са организирани съответно в база данни, база знания и интерпретатор на знанията (ИЗ). Веднага се вижда, че за разлика от традиционните програмни системи, работещи на базата на програми и данни, интелектуализираните програмни системи съдържат и трета част – базата знания. Традиционните програмни системи също съдържат знания, но те са вложени в самите програми (тези знания съвпадат до някъде с т. нар. процедурни знания – вж т. 4 на тази глава). Например, една програма за решаване на квадратно уравнение "знае" как се решават квадратни уравнения. Разликата в сравнение с ИПС е, че такава програма не обработва знанията, а само ги използва, и то по начина, по който програмистът или съставителят на алгоритъма ги е осмислил.

Отношението между данни и знания е един от основните

проблеми на информатиката в последно време. В същност, тези две понятия представляват двете страни на категорията информация. Знанията са общата, относително постоянната и неизменяема част от информацията. Те представляват информация за клас обекти, за закони, взаимовръзки и др. Този тип информация (знанията) е относително независим от отделните обекти. Противоположност на знанията (предимно по отношение на общност и динамичност) са данните, които представляват конкретната информация за всеки обект. Те са относително по-динамичната част от информацията. В този смисъл, знанията и данните се явяват две взаимно допълващи се, но различни части на информацията като цяло. Например, фактите, че кривата K е окръжност и радиусът ѝ е 1 см са данни, но фактът, че дължината на всяка окръжност се намира по формулата $S=2\pi R$, е знание. В този смисъл трябва да отбележим, че в понятията "данни" и "знания" не се влага точно житейския смисъл с който сме свикнали. Например, когато казваме "Знам че радиусът на окръжността K е 1 см", това не означава че, $R = 1\text{см}$ е знание. Рязка граница между данни и знания не съществува. В този смисъл не съществуват "чисти" знания и "чисти" данни. Това, което представлява знания за системата на едно ниво, може да е данни на друго и т. н., както и данните могат да се интерпретират като знания в определени случаи.

Една, макар и не абсолютна характеристика, отличаваща знанията от данните, е тяхната динамичност. Докато данните са динамичната част на системата, знанията са относително по-постоянни, макар че и те могат да подлежат на промяна.

Друг съществен момент е, че върху данните се извършват

действия, докато знанията се използват за вземане на решение, правене на извод или заключение, съставяне на план и др., т. е. те носят информация за връзките между данните. В този смисъл, знанията на системата, когато върху тях започнат да се правят изменения, влизат в ролята на данни относно изменящата ги подсистема (подсистемата за обработка на знанията). Ако се върнем на връзката между традиционните програмни системи и интелектуализираните програмни системи, можем да видим че знанията при програмните системи са вградени в самите програми и не подлежат на промяна, докато при интелектуализираните програмни системи е възможно знанията да се променят (което би съответствало на гъвкавост и изменяемост на програмите в една програмна система, ако това е възможно).

Друга класификация на разликите между знания и данни срещаме в [Представяне... 84]. Според Д. А. Поспелов, знанията се отличават от данните по три основни характеристики: интерпретируемост, връзка и наличие на класифициращи връзки. Докато данните са записани в съответна структура и няма възможност за тяхната интерпретация, знанията носят информация за интерпретацията им в самите тях. Общите признаци, родово-видовите връзки, унаследяването на свойства – това са вертикалните връзки в знанията, наречени класифициращи връзки. Хоризонталните връзки, наречени още ситуативни отношения, са третата основна характеристика, отличаваща знанията от данните.

2. ТРИ ТИПА ЗНАНИЯ

Знанията могат да се разделят на три основни типа:

- знания за обекти и факти;
- знания за връзките между обектите и фактите;
- метазнания, т.е. знания за самите знания.

Това деление е необходимо поради използването на отделните типове знания по различен начин, от различни подсистеми и в различни етапи от работата на ИПС.

2.1. Знания за обекти и факти

Това са знания за обекти, факти, за характеристиките на класове обекти и факти, обединяващи ги като такива. Те представляват относително затворени описания на обектите или елементите на предметната област.

Тези знания служат за представяне на свойствата и характеристиките на обектите и са най-близки до данните. С тях се работи лесно – описанието и използването им е сравнително просто. Този тип знания се използва предимно в интерфейсната част на системата за разпознаване на заявката към системата и за извличане на информацията за обектите при работата на интерпретатора на знанията. Ако се върнем на примера за окръжността, знанията от този тип биха били:

окръжност

радиус – R

дебелина – $2PR$

ъгли – няма

гладкост – навсякъде

2.2. Знания за връзките между обектите и фактите

Това са знанията, отразяващи връзките между обекти от предметната област и между отделните компоненти на различните обекти. Те представляват правила, умения, методи и др. Знанията от този тип се използват предимно при работата на интерпретатора на знанията за решаване на поставената задача. На базата на тези знания интерпретаторът управлява процеса на решаване на поставената задача, взема решение кога какво да се анализира и кой обект да се разглежда.

Знанията за връзките между обектите носят информация за интерпретацията на знанията от първия тип и се делят на две групи:

- знания за вертикалните връзки, наричани още класифициращи връзки – това са общите признаци, родово-видовите връзки, наследяването на свойства;

- знания за хоризонталните връзки – ситуативните отношения и причинно-следствените отношения.

Така както няма рязка граница между знания и данни, още по-размита е границата между отделните типове знания. Например, знанието, че ако K е окръжност с радиус R то

гължината ѝ е 2PR може да се тълкува като връзка в конкретна ИПС за решаване на геометрични задачи, но може да се тълкува и като факт (т.е. знание от първия тип) при ИПС за разпознаване на геометрични тела. Принадлежността към един или друг клас знания се определя преди всичко от начина, по който се използват съответните знания в системата.

2.3. Метазнания

3.1. Семантични елементи

Това са знания за самите знания (от първите два типа), за тяхното представяне, за тяхното използване, за тяхното изменение и не на последно място знания за предметната област. Метазнанията са знания от "по-високо ниво". Тъй като представляват знания за самите знания, характерно за тях е че или са вградени в интерпретатора на знанията, или, ако са част от базата знания, се използват предимно от интерпретатора. Самите понятия, свързани със знанията, които ще разгледаме по-нататък, като предикат, терм, правило, фрейм, семантичен елемент и др., са метазнания. Например, понятието "правило" е знание за структурата на знанията в системите продукции, за структурата на двойките от типа условие - следствие, за връзката между условието и следствието и т. н. В този смисъл, самият интерпретатор на знанията е метазнание, представено по определен начин.

3. ПРЕДСТАВЯНЕ НА ЗНАНИЯТА

Структурата на представянето на знанията се определя от две основни характеристики – градивните елементи на базата знания, или т. нар. семантични елементи, и от връзките между отделните единици знания – т. е. между семантичните елементи.

3.1. Семантични елементи

При избирането на схемата за представяне на знанията веднага изниква въпросът за семантичните елементи, т.е. за "речника" на базата знания. Така например, при представяне на знанията с помощта на предикати трябва да се определят предикатите, при представяне със семантични мрежи трябва да се определят възлите и връзките, при представяне чрез фреймове – да се определят фреймовете и т. н. Споменатият по-горе пример за дължината на окръжността може да се представи чрез предикати и чрез семантични мрежи по няколко различни начина:

а) чрез предикати

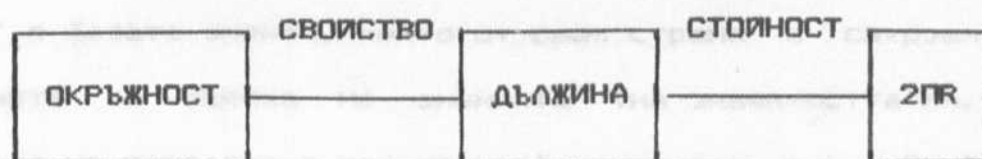
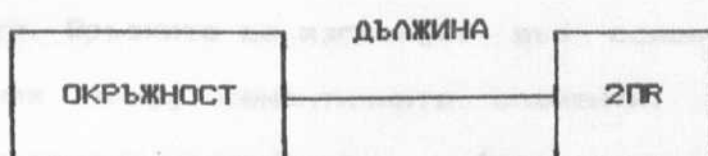
ДЪЛЖИНА (2ПР)

ДЪЛЖИНА (ОКРЪЖНОСТ , 2ПР)

ДЪЛЖИНА (ОКРЪЖНОСТ , R , 2ПР)

СВОЙСТВО (ОКРЪЖНОСТ, ДЪЛЖИНА , 2ПР)

б) чрез семантични мрежи



По различни начини може да се представи този факт и чрез фреймове, процедурно и т. н. Общи правила за избора на семантичните елементи няма. Те се определят в зависимост от конкретното приложение, от нуждите на потребителите, от възможностите на изчислителната система и др. Изборът на семантичните елементи определя до голяма степен схемата за представяне на знания, която ще се използва в конкретната реализация. Трябва да се подчертае, че освен че е зависим от конкретната приложна област, изборът на семантични елементи не е еднозначен, т. е. едни знания могат да се представят чрез различни типове семантични елементи.

3.2. Връзки между семантичните елементи

Другият структуроопределящ фактор при представяне на знанията – това са връзките между отделните семантични елементи. Не случайно, на много места базата знания се интерпретира като граф, в който възлите са семантичните

елементи, а ребрата са връзките между семантичните елементи. Връзките се изграждат въз основа на смисловите отношения между семантичните елементи. В общия случай са неравномерно разпределени, йерархични и наподобяват телефонна мрежа или карта на автомобилните пътища [Мински 74]. Този начин на изграждане на връзките от своя страна поражда проблема за избор на "главни центрове" и "главни връзки" в базата знания, който от своя страна е свързан с проблемите за оценка на знанията (на значимостта им, на честотата на използване или на необходимостта им). Особено въпросът за значимостта (оценката) на знанията за сега е открит.

4. ДВА ОСНОВНИ ТИПА ПРЕДСТАВЯНЕ НА ЗНАНИЯ

Преди да започнем разглеждането на съществуващите в момента схеми за представяне на знания, нека се спрем накратко на двата основни типа, на които се делят тези схеми. Още при първите насочени опити за представяне на знания, схемите за представяне рязко се разграничават на два типа – процедурно и декларативно представяне на знанията. Основната разлика между двата типа е в това, дали базата знания и интерпретаторът са изградени като едно цяло или са обособени като отделни части, т. е. дали знанията и метазнанията се вграждат в системата заедно или поотделно.

При процедурния тип представяне на знания, знанията на системата се представят явно – вграждат се "твърдо" в системата. Това обикновено е случаят, в който базата знания и интерпретаторът са изградени като едно цяло. Този начин на

представяне дава възможност за създаване на много връзки и за изграждане на развита вътрешна структура на базата знания, което води до висока ефективност при използването им. Вграждането на връзките и знанията в самата система обаче води до големи затруднения при представянето (кодирането) на знанията и особено при тяхното допълване и изменение. За решаването на този проблем в някои случаи се използва хетероерархична структура [Уинстон 84] от отделни процедури ("демони"), които "знаят" как да реагират в съответната ситуация.

При декларативния тип представяне на знания обикновено интерпретаторът се създава независимо от базата знания. При този тип представяне на знания имаме голяма модулност на знанията - базата знания е изградена от еднакви градивни елементи, което води до голяма гъвкавост на системата и лекота на изменение на данните. В този случай, еднаквостта на отделните подструктури знания води след себе си до хомогенност на системата, т.е. липса на дълбока йерархия в структурата, което е причина за сериозни затруднения при използването на знанията. Тъй като интерпретаторът е отделен от базата знания, метазнанията (т. е. знанията, вградени в интерпретатора) са общи и зависят от характерните особености на предметната област в много по-малка степен, отколкото при процедурното представяне, където метазнанията са неделима част от базата знания и зависят силно от предметната област.

Като следствие от различията в структурата на базата знания и интерпретатора при декларативните и процедурните типове се явяват и различия в управлението (използването) на знанията. При декларативния тип за представяне на знания

управлението е глобално и не отчита характера на знанията и на предметната област, тъй като интерпретаторът е отделен от базата знания. При процедурния тип за представяне на знания, управлението е локално, тъй като метазнанията, т. е. знанията за самите знания и за тяхното управление и използване са в самата база знания. И двата вида управление (глобално и локално) имат своите преимущества и недостатъци, тъй като при единия се запазва общият поглед върху нещата, но не се отчитат особеностите, свързани с характера на задачата и на предметната област, а при другия обратно – отчитат се особеностите, но се губи общият поглед върху работата на системата.

Противоречието между типовете за процедурното и декларативно представяне на знания е основно противоречие в областта на представянето на знания [Чинстон 84]. То датира още от началото на насочените изследвания в областта на представянето на знания (края на 60-те години). В резултат от развитието и противопоставянето на двата типа представяне, през 1974 г. Марвин Мински стига до идеята за обединяване на тези два типа в процедурно – декларативна структура за представяне на знания, наречена от създателя си "фрейм" (рамка, скелет) [Мински 74]. Фреймовете дават възможност за удобно представяне на знания и от трите типа, разгледани в т. 2 на настоящата глава, като запазват добрите възможности на процедурното представяне за създаване на дълбока структура и висока ефективност, но освен това притежават удобствата на декларативното представяне за изменение и глобално управление на базата знания.

5. ОСНОВНИ СХЕМИ ЗА ПРЕДСТАВЯНЕ НА ЗНАНИЯ

В процеса на изследванията в областта на представяне и използване на знанията са се оформили няколко основни схеми за представяне на знания. Ще направим кратка характеристика на тези схеми. Ще илюстрираме схеми с едни и същи примери, за да покажем по-добре предимствата и недостатъците на всяка от тях.

Основните схеми за представяне на знания, които ще разгледаме, са пет – предикатно смятане от първи ред, процедурно представяне, семантични мрежи, системи продукции и фреймове.

5.3. Семантични мрежи

5.1. Предикатна логика от I ред

Това е схема от декларативен тип за представяне на знания, в който знанията се представят с помощта на предикати, а използването на знанията става чрез прилагане на определени правила за извод. Например знанието за дължината на окръжността може да се представи чрез предикат по следния начин:

ДЪЛЖИНА (ОКРЪЖНОСТ , $2\pi R$).

5.2. Представяне чрез процедури

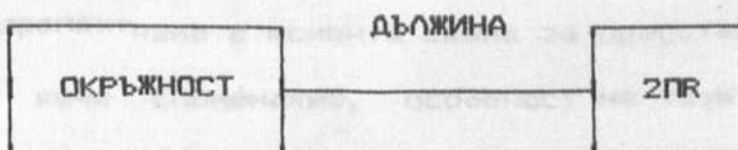
Това е схема от типичен процедурен тип представяне на знания (което се вижда и от самото му име). При процедурното представяне, знанията са заложи в малки процедури, които

"знаят" как да действат в специфични ситуации. Този начин на представяне на знания се използва често при управление на работи. В литературата за представянето на знания чрез процедури често се използва терминът "демони". Нашият пример за знанието за дължината на окръжността ще изглежда по следния начин:

```
PROC LENGTH
  READ R
  L = 2 * П * R
  PRINT L
```

5.3. Семантични мрежи

Това е схема от декларативен тип за представяне на знания, представляваща мрежа от възли (клетки, позиции) и връзки между тях, като възлите и връзките носят определен смисъл. Възлите са носители на определени факти, а връзките задават отношенията между фактите. Знанието за дължината на окръжността може да се представи чрез семантична мрежа по следния начин:



5.4. Системи продукции

Това е в основата си схема от декларативен тип за представяне на знания, в която знанията се представят във вид на отделни елементи, наречени правила (продукции). Продукциите се състоят от две части – условие и следствие, като следствието се изпълнява (или е в сила), ако е изпълнено условието. Структурата на отделните правила, които са подобни на малки процедури показва, че макар и декларативна, тази схема съдържа елементи и от процедурния тип представяне. Това ни дава право да я определим като смесен тип. Съчетаването на елементи и от процедурните и от декларативните схеми е причината за високата ефективност на системите продукции (спрямо другите декларативни схеми), а от там и за тяхната популярност при създаване на интелектуализирани програмни системи.

Знанието за дължината на окръжността може да се представи чрез продукция по следния начин:

IF (ФИГУРАТА Е ОКРЪЖНОСТ) THEN (ДЪЛЖИНА = $2\pi R$).

5.5. Фреймове

Фреймовете представляват структури за описание на стандартни ситуации. Това е най-новата и считана за най-перспективна в момента схема за представяне на знания. Както вече споменахме, особеност на тази схема е, че тя е комбинирано процедурно – декларативно описание на знания. Това позволява фреймовете да обединят предимствата както на процедурните, така и на декларативните типове схеми за

представяне на знания. Ако се върнем към примера за знанието за дължината на окръжността, можем да го представим чрез част от фрейм по следния начин:

```

#FR: ОКРЪЖНОСТ IS: ФИГУРА

РАДИУС : R

ДЪЛЖИНА : 2πR

ПЛОЩ : πR2

```

По-подробно понятието фрейм е разгледано в глава III, т.2 във връзка с езика FRL.

До тук се спряхме накратко на основните схеми за представяне на знания. Някои автори ([Бар, Фегенбаум 81]) отнасят към схеми за представяне на знания и пространството на състоянията в задачите за търсене в пространството на състоянията. Ние няма да се спираме на него, тъй като търсенето в пространство на състоянията е изследвано много подробно независимо от връзката му с представяне на знания, а проблемите там са свързани преди всичко с търсене в дърво. Друга схема, на която няма да се спираме, е т. нар. директно (аналогово) представяне. Тази схема е свързана преди всичко с управление на процеси и там представянето на знанията е силно зависимо от предметната област.

ГЛАВА III. ИНСТРУМЕНТАЛНИ СРЕДСТВА

1. ЛИСПОДОБНИ ЕЗИЦИ ЗА ПРОГРАМИРАНЕ

Езикът ЛИСП е създаден през 1959 г. от Джон Маккарти [Маккарти 60], [Маккарти 60а], с цел по-удобно и бързо описване на алгоритми за обработка на данни със списъчна структура. Това се налага много често при задачи за моделиране на мисловни процеси, аналитични преобразования и други подобни проблеми, включвани в понятието "изкуствен интелект" [Томов, Геров, Спиридонова].

В последствие езикът ЛИСП търпи развитие и така се стига до различни версии, диалекти и дори нови езици.

1.1. Описание на езика ЛИСП

Атоми, списъци и функции. Атомът е основна неделима информационна единица. Има три вида атоми:

- символни - последователност от букви и цифри, започваща с буква, например: АТОМ, ИМЕ, СТОЙНОСТ;
- числови константи - това са допустимите числови константи в езика ФОРТРАН от тип реален и цял;
- символни низове - последователност от символи, заградена със специални символи, например: [НИЗ], [А+В], 15+3/1.

Атомите могат да се използват като литерали, т. е. самите те да представляват и име, и стойност на величината, или като имена на други обекти, т. е. имена на атоми, списъци и функции. В езика има два атома, които имат постоянна стойност. Това са атомите: T, имащ стойност T (истина), и атома NIL, имащ стойност NIL (лъжа).

Списъкът е последователност от атоми и списъци, заградена в скоби. Списък например е:

```
(СОФИЯ ИХТИМАН КОСТЕНЕЦ БЕЛОВО СЕПТЕМВРИ
 (ВЕЛИНГРАД ЯКОРУДА)
 ПАЗАРДЖИК НОВИ-КРИЧИМ (КРИЧИМ) ПЛОВДИВ)
```

Празният списък е списък, който не съдържа нито един елемент. Той може да се запише по следните два начина: () или NIL, т. е. атомът NIL.

Функциите представляват възможните обработки над атомите и списъците в езика. Тяхното представяне е също списъчно. Общият вид на обръщение към функция е:

```
<ИМЕ НА ФУНКЦИЯ> <АРГ. 1> <АРГ. 2>... <АРГ. N>
```

Синтаксис на програмата. Програмата представлява последователност от обръщения към функции, за които са позволени както рекурсия, така и суперпозиция. Синтаксисът е прост и се описва леко.

Средства на езика. Средствата на езика се определят от набора функции, които са включени в реализацията на езика, т. е. вградените в транслятора функции. Освен тези функции, потребителят на езика може да дефинира произволен брой допълнителни функции чрез самия език ЛИСП. Функциите,

дефинирани от потребителя могат да се използват по същия начин както вградените функции. Тяхното изпълнение ще бъде сравнително бавно при обичайния за ЛИСП режим на интерпретация. Това неудобство може да се преодолее, като се използва вградената функция за компилиране на новодефинираните функции. Така бързината на тяхното действие се приближава до тази на вградените функции. Следва описание на малка част от вградените функции чрез примери:

ФУНКЦИИ-ОБРАБОТКИ:

(QUOTE AK) дава AK, (QUOTE (A B)) дава (A B), 'AK дава AK – третиране на аргумента като литерал.

(CAR '(A B C)) дава A – първи елемент на списъка.

(CDR '(A B C)) дава (B C) – списък без първия елемент.

(SETQ P '(A B H)) дава (A B H) и присвоява на P (A B H).

(LIST 'A1 'H 'P P) дава (A1 H P (A B H)) – образуване на списък.

(APPEND P '(E M)) дава (A B H E M) – свързване на два списъка.

(REVERSE P) дава (H B A) – обръщане на наредбата.

ФУНКЦИИ-ПРЕДИКАТИ:

(ATOM P) дава NIL, (ATOM 'P) дава T – проверка за атом.

(NULL H) дава NIL, (NULL NIL) дава T – проверка за празен списък.

ФУНКЦИЯ ЗА УСЛОВИЕ:

(COND <<ПРЕДИКАТ1> <ИЗРАЗИ1> <ИЗРАЗИ2> ... <ИЗРАЗИN1>>
 <<ПРЕДИКАТ2> <ИЗРАЗИ21> <ИЗРАЗИ22> ... <ИЗРАЗИ2N2>>
 ...
 <<ПРЕДИКАТK> <ИЗРАЗИK1> <ИЗРАЗИK2> ... <ИЗРАЗИKNK>>

Функцията предизвиква изпълнението на първата последователност от изрази, пред която съответният предикат има стойност, различна от NIL.

Освен изброените видове функции в езика има и функции за вход и изход, за група изрази, в която може да се използват преходи и излизания от групата (PROG, GO и RETURN), функции за аритметични действия и др.

Изпълнение на програмата. Транслаторът на езика ЛИСП работи в режим на интерпретация. Това означава, че последователно се въвежда всеки израз от най-високо ниво (изрази от програмата), пресмятат се неговите аргументи, изпълнява се указаната функция и се извежда полученият резултат. Присвоените стойности и дефинираните функции са глобални за цялата програма. Има възможност за група изрази (в PROG) да се обявяват локални променливи.

1.2. Стил на програмиране

Съставянето на програми на ЛИСП се свежда в основни линии до дефинирането на нови функции, които от своя страна могат да се използват пряко или рекурсивно при дефинирането на други функции. Самата програма от своя страна е

равнително проста от логическа гледна точка, понеже изразите в програмата се изпълняват последователно и те обикновено са обръщения към сложни, дефинирани от програмиста, функции.

Следва един пример за дефиниране на функция, който явява стила на програмиране. Нека разгледаме дадения в началото като пример списък. Елементите, които са атоми, са мена на гари по линията София - Пловдив. Елементите, които са списъци, представляват отклонения по тази линия. Така този списък показва до кои гари може да се стигне по направление София-Пловдив. Целта на функцията е да отпечата тези гари. Следователно трябва да се отпечатат тези елементи на списъка, които са атоми, а елементите, които са списъци, трябва да се разглеждат отново като списъци и да се отпечатат само елементите атоми и т. н. в зависимост от дълбочината на влагане.

Първо да опишем алгоритъма на функцията с думи, и то по начина, по който обикновено човек расъждава.

АКО няма елементи в списъка

ТО - нищо не се прави.

(това беше съвсем лесно и затова почнахме от него)

АКО първият елемент е атом

ТО - печат на този елемент;

(това следваше непосредствено от условието, а сега трябва да изпълним функцията за останалата част на списъка)

- и рекурсивно изпълнение на

функцията за останалата част

(остава най-сложният случай – ако първият елемент е списък; но тогава просто трябва да приложим нашата функция последователно за първия елемент и за останалата част от списъка)

АКО	първият елемент не е атом
ТО	– рекурсивно изпълнение на функцията за първия елемент;
	– и рекурсивно изпълнение на функцията за останалата част.

При описването на алгоритъма следвахме непосредствено условието на задачата. При това описахме крайния случай – когато списъкът е изчерпан – и началния случай – когато разглеждаме първия елемент на списъка. Останалите случаи рекурсивно свеждаме към първите два.

Така описаният алгоритъм се представя чрез функция на ЛИСП по следния начин:

```
(DE V (S)
  (COND
    ((NULL S) NIL)
    ((ATOM (CAR S)) (PRINT (CAR S))
      (V (CDR S)))
    (T (V (CAR S))
      (V (CDR S)))
  ))
```

Тук функцията DE служи за дефиниране на функции, а COND съответства на конструкцията АКО... ТО...

РЕКУРСИВНОСТ

В показания пример функцията е къса и логическата структура е проста благодарение на възможността да се използва рекурсивно самата функция, която дефинираме. Това ни дава възможност да се съсредоточим само върху крайните случаи в задачата, без да обмисляме междинните ситуации.

АПЛИКАТИВНОСТ

Една от основните цели при създаването на езика ЛИСП е била да се създаде основа на математическата теория на изчисленията [Евтимов 80]. Това се изразява с факта, че главна роля в ЛИСП имат понятията функция, апликация (прилагане на функцията към аргументите) и оценка (определяне на значението на израза). Това дава възможност при програмирането на ЛИСП да се избегнат многократните присвоявания на стойности на една променлива и от тук по-висока степен на надеждност и вярност на програмата, а също позволява прилагането на автоматични средства за проверка на програмата.

СТРУКТУРНОСТ

В началото бе отбелязано, че програмата на ЛИСП представлява обръщания към функции, които обикновено се дефинират от програмиста, а тези дефинирани функции най-често включват в себе си обръщания към също такива функции. Това дава възможност програмирането да се извършва отгоре надолу, като първо се пише реда на изпълнението на функциите и после последователно се дефинират функциите, започвайки от най-външните, а вътрешните само се маркират.

Този стил се нарушава в известен смисъл при използването на косвена рекурсия. В този случай е необходимо във всяка функция да има средства за проверка на коректността на нейното извикване.

1.3. Версии на езика ЛИСП

Първата реализация на езика е за машината IBM 704 от Artificial Intelligence Group на MIT. Непосредствено след нея е направена и реализацията за IBM 7090 под името ЛИСП 1.5 [Маккарти 62]. В нея са вградени редица нови функции, които могат да се опишат и на ЛИСП. Реализации на езика ЛИСП 1.5 има и за машините CDC-3600, IBM 360/370, UNIVAC 1100, БЭСМ-6, PDP-1, PDP-6 и др. Направен е интерпретатор за ЛИСП 1.5 на езика ФОРТРАН, като по този начин се постига една сравнително независима от машината реализация [Нордстром 70].

Всички споменати реализации се различават сравнително малко и носят общо име ЛИСП 1.5. Но с разширяването на областта на приложение започна създаването на нови варианти и версии на езика ЛИСП [Деранзарт 79]. Различията са в средствата, които в първоначалния ЛИСП не са били застъпени – например средствата за вход и изход – и в добавянето на едни или други функции и конструкции, което става сравнително леко благодарение на възможностите на езика за разширение със средствата на самия език.

Около 1970 г. се създават три нови версии на езика ЛИСП, които получават голяма популярност. Това са ЛИСП 1.6, MACLISP и INTERLISP. Те включват голям обем от средства за

създаване на програми [Сангеуол 73], [Тейтелман 74], [Маурер 76], [Чайл 78].

Отличителните черти на тази група версии на езика и различията между тях са:

- силно развити средства за включване на програми, написани на асемблер, чрез използване на специален файл за листови асемблерови програми (LAP - файл);
- средства за работа с потребителски програми. В ЛИСП 1.6, има създаден специален редактор ALVINE. За MACLISP се използва стандартния текстов редактор на системата ITL (създадена в MIT) с малки изменения, който работи паралелно с транслатора. В INTERLISP средствата за работа с потребителски програми са развити най-добре. Създаден е специално за езика текстов редактор на интерактивен принцип;
- добре развити средства за търсене на грешки и проследяване на изпълнението на програмата. В INTERLISP има възможност за тестване и коригиране в диалогов режим;
- средства за програмиране отгоре надолу. В езика ЛИСП е разрешено да се пишат обръщания към функции, които не са дефинирани, но ако в програмата се стигне до тяхното изпълнение, се индицира грешка. В MACLISP и особено в INTERLISP тази концепция е разширена с възможността потребителят да указва какви действия да се предприемат в момента на обръщение към недефинираната функция;
- макро средства. И в трите версии е включен макрочетец, а в MACLISP има средства за дефиниране на макроопределения;
- освен операторите за присвояване и преход, включени и в езика ЛИСП 1.5, в INTERLISP има реализирани някои

АЛГОЛ-подобни средства, като конструкциите за цикъл FOR, за условие IF... THEN... ELSE и т. н. [Тейтелман 74];

- в трите версии са включени компилатори за ЛИСП - функции [Стийл 77].

Напоследък голямо разпространение и приложение получиха микрокомпютрите. Това предизвика създаването на нови версии на езика LISP, които по принцип включват повечето възможности на досегашните версии на езика, но обогатени и развити. Ще отбележим някои от най-важните характеристики на тези нови версии:

- разширен набор стандартни функции;
- подобрени възможности за вход-изход и обработка на графична информация;
- развити възможности за включване на функции и програми, написани на други алгоритмични езици;
- повече средства за проверка на програмите.

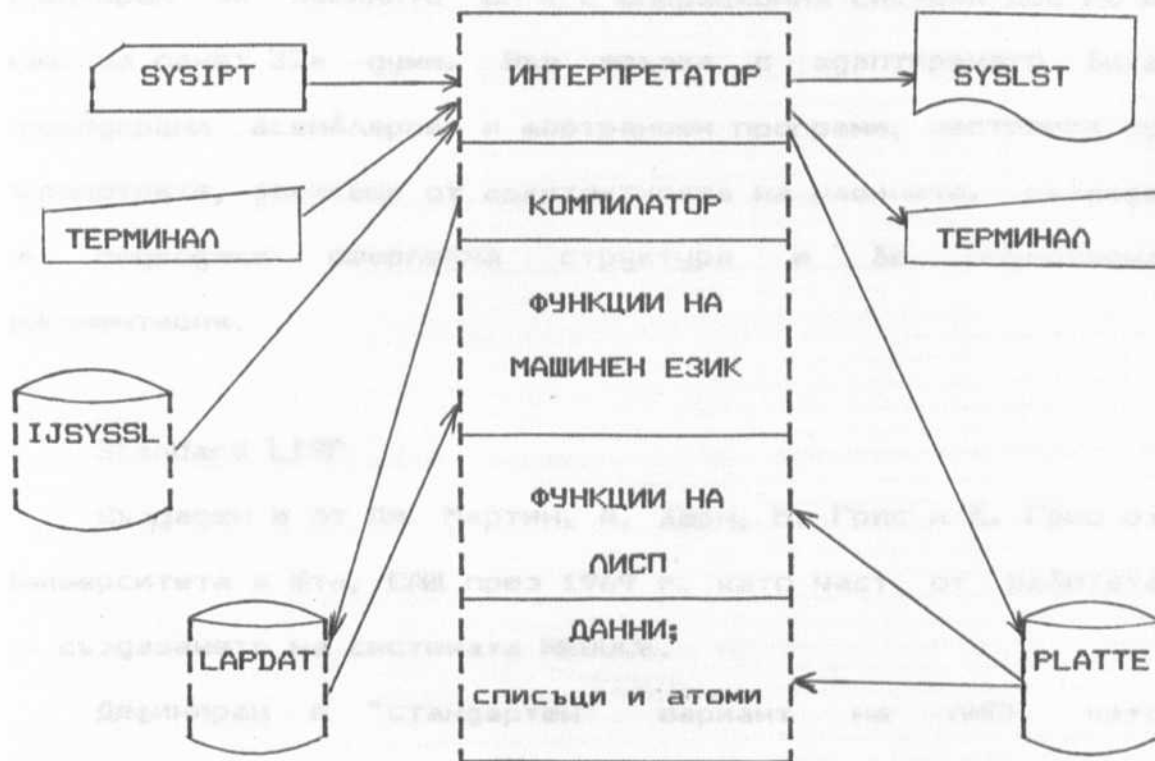
Типични представители на тези нови версии на езика LISP са: IQLISP и Golden Common LISP.

1.4. Транслатори от ЛИСП, внегренни у нас

TU LISP (ЛИСП 1.6)

Реализацията е направена в Дрезденския технически университет за ЕС ЕИМ. У нас транслаторът е усвоен в ЕЦММ при БАН. Проучени са възможностите на транслатора и е внегрен в заинтересованите институти. Подготвена е документация на български език ([Стоян 79], [Стоян 80], [Геров 80]). Ползва се за целите на обучението в СУ"Кл. Охридски" и Националната природо-математическа гимназия.

На фиг.7 е дадена обща схема на транслятора



Фиг. 7. Обща схема на транслятора TU LISP

Функции на машинен език (вградени функции). Това са функциите, реализирани в транслятора и функциите, съставени от програмиста на асемблер или на ЛИСП и транслирани с помощта на компилатора. Такива функции може да има записани и във файла LAPDAT, от където се въвеждат в оперативната памет при обръщение към тях в програмата.

LISP F3 (INTERLISP)

Друг транслятор за ЛИСП, който е усвоен в ЕЦММ при БАН и в БСНИПИ "Интерпрограма" е LISP F3 (версия на INTERLISP), реализиран на езика ФОРТРАН в Datalogilaboratoriet, UPPSALA, SWEDEN. Този транслятор се използва на ЕС ЕИМ с операционна система ОС и у нас е направена генерация, която да работи в

диалоговата система СИНТЕР (TSO). Същият транслатор е адаптиран за машините CM 4 с операционна система DOS PB и изисква памет 32к думи. Във връзка с адаптирането бяха прекодирани асемблерови и фортранови програми, настроиха се параметрите, зависещи от архитектурата на машината, създаде се подходяща оверлейна структура и бе подготвена документация.

Standard LISP

Създаден е от Дж. Мартин, А. Херн, М. Грис и К. Грис от Университета в Юта, САЩ през 1969 г. като част от работата по създаването на системата REDUCE.

Дефиниран е "стандартен" вариант на ЛИСП като подмножество на ЛИСП 1.5 и неговите версии, такова, че всяка програма, написана на това подмножество, да може да се изпълнява с всяка "разумно направена" ЛИСП-система.

Транслаторът е усвоен в ЕЦММ при БАН за ЕС ЕИМ в операционната система ОС.

IQLISP и GCLISP

Транслатори от IQLISP и GCLISP за персоналните микрокомпютри IBM PC/XT и IBM PC/AT са усвоени и се използват от ИМ с ИЦ при БАН, от БСНИПИ "ИНТЕРПРОГРАМА" и други организации.

за която се създава ЕС.

Според функционалната си структура и предназначение ЕС изискват различни ЕИМ – от малки и средни до ЕИМ с големи памети и скорости и с развита периферия. Особено подходящи за ЕС се оказват и някои персонални компютри.

Алгоритмичните езици, на които се разработват ЕС, са предимно езици на изкуствения интелект като LISP, PROLOG и др.

При ЕИМ от новите поколения се поставя въпросът за апаратно реализиране на някои от основните модули на ЕС.

1.1. Основни характеристики на ЕС

Основно място в изкуствения интелект заемат въпросите, свързани с представянето на знанията. Това е, може би, и най-характерната черта на ИИ в сравнение с останалите направления в информатиката. Същото може да се каже и за ЕС. При тях знанията за определена предметна област играят водеща роля. Една от основните характеристики на ЕС е наличието на три отделни нива – програми, данни и знания. Последните две нива често се означават с термините "база от данни" и "база от знания". Начинът, по който взаимодействат тези нива в ЕС, е принципно различен в сравнение с други системи. Тук данните не се използват пряко, а до тях се достига чрез БЗ. Знанията са тези, които определят в кой момент да се използват или преработват определени данни, като това не е определено в програмите, а зависи от конкретната ситуация, създадена от текущата задача. Може да се каже, че достъпът до данните е смислово определен, което

е следващата характеристика на ЕС. По аналогичен начин и достъпът до знанията от БЗ не е детерминиран от програмите, а се определя от възникналата конкретна ситуация – от знания, които указват в даден момент от решаването на задачата какви други знания са необходими. Така се стига до следващата характеристика на ЕС, а именно че управлението е "контекстно зависимо", т. е. определя се от смисъла на ситуациите, възникнали при решаване на конкретната задача, а не от предварително заложен в програмата детерминиран алгоритъм. Накрая ще се спрем и на зависимостта на знанията от програмите. От факта, че знанията определят управлението на решението на задачата и от факта, че достъпът до знанията и до данните се извършва чрез самите знания, следва независимостта на знанията от програмите, което е още една основна характеристика на ЕС.

1.2. Област на приложение

1.3. Класификация на ЕС

Областта на приложение на ЕС ще представим чрез определяне характеристиките на задачите, които се решават с тяхна помощ [Ленат и др. 83].

На първо място това са задачи, свързани с търсене в пространството на състоянията в случаите, когато това пространство е голямо и се налага използване на ЕИМ за решаването на съответните задачи.

Втора характеристика на задачите, решавани с помощта на ЕС, е липсата на традиционни средства за моделиране на задачата и от там за нейното решаване. При наличието на традиционни средства е по-добре да се използват те, понеже

се постига по-висока скорост на решаване на задачите и по-голяма увереност в успеха при решаването им.

Тъй като при ЕС знанията се представят по начин, който съдържа в себе си естествените връзки между отделните понятия, те са предназначени за решаване на задачи, за които е характерно наличието на голямо разнообразие от обекти, връзки (отношения) и операции в предметната област, от която са задачите.

Друга характеристика на този вид задачи е наличието на голям набор от евристични (емпирични) правила за вземане на решения в съответната проблемна област. Тези правила са част от знанията за предметната област и се намират в БЗ.

Накрая ще отбележим, че при разработване на конкретна ЕС за дадена предметна област е необходимо поне един специалист от тази предметна област да участва активно при нейното разработване и особено при изграждането на БЗ.

1.3. Класификация на ЕС

Ще класифицираме ЕС в зависимост от типовете задачи, които се решават с тяхна помощ. В основни линии те са следните [Хейс-Рот и др. 83], [Стефик и др. 82]:

- интерпретация на данни;
- диагностика;
- контрол или управление на процеси;
- прогнозиране;
- планиране;
- проектиране.

При ИНТЕРПРЕТАЦИЯ НА ДАННИТЕ целта на работата е да се

определи смисъла на някаква съвкупност от данни. На базата на едно представяне на даден обект е необходимо да се намери друго представяне на този обект, което поради някакви причини е по-удобно за потребителя. Например, при ЕС DENDRAL [Бар, Фегенбаум 82], [Буханан и др. 69] по данните от мас-спектралния анализ се търси структурното представяне на молекулата. Интерпретирането на данните в ЕС става въз основа на знания от БЗ за конкретната предметна област, за разлика от традиционните интерпретатори, където цялата информация се съдържа в програмите и в данните. При този тип задачи често се налага анализирането на голямо множество варианти, приближавайки се до пълно изчерпване. Друг тежък проблем при интерпретирането е свързан с неточността или непълнотата на данните. Това означава, че ЕС трябва да може да работи с противоречива или непълна информация, за сметка на знанията, заложили предварително за предметната област.

В областта на ДИАГНОСТИКАТА съществуват най-голям брой ЕС [Шортлиф 76]. Това е тясно свързано с целта, която се поставя пред този вид ЕС – да се открие повреда или неизправност в дадена система, като в това се включва и поставянето на диагноза при заболяване на човек (ЕС MYCIN). Данните, които се използват при тези ЕС, могат да се окажат, както недостатъчни от една страна, така и ненужни от друга страна. Дори в някои случаи данните могат да бъдат неправилни. В тези случаи ЕС трябва не само да преодолее неточността в данните, но и да определи и изиска от потребителя информация за необходимите данни.

Характерни за задачите, свързани с КОНТРОЛ или УПРАВЛЕНИЕ НА ПРОЦЕСИ, са както проблемите, свързани с

интерпретация на данните (поради факта, че самата система възприема непосредствено необходимите данни), така и проблемите, свързани с диагностика (за да се определят причините за отклоненията в поведението на управляваната система). Към тези, вече известни проблеми от първите два типа ЕС, се прибавя и необходимостта за работа на ЕС в реално време, което е продиктувано от управлението на система, действаща в реално време.

При задачите за ПРОГНОЗИРАНЕ е необходимо на базата на непълни знания и непълна информация да се предскаже поведението на конкретна система в следващ момент от времето. Основната характеристика е непълнотата на знанията и/или данните. В противен случай прогнозирането може да стане чрез изчисление, а не чрез експертиза.

ПЛАНИРАНЕТО представлява създаване на план от действия за постигане на предварително поставена цел. Тъй като, при планирането данните и знанията могат да бъдат непълни и е необходимо да се предвиди въздействието на разнообразни фактори, то при решаването на този тип задачи съществуват проблеми, близки до тези при прогнозирането. Освен това от голяма важност е възможността за генериране на алтернативни планове и съобразяването на тези планове с ограниченията на наличните ресурсите за изпълнение на плана.

Накрая ще се спрем на ПРОЕКТИРАНЕТО. Този вид системи до голяма степен приличат на системите за планиране, като целта е създаване на конкретна система с предварително определени характеристики. Това налага необходимостта както от прогнозиране, планиране и създаване на варианти, така и от обосновка на отделните решения.

Разбира се, допустими са и други класове ЕС, различни от посочените по-горе. Така например, интерес представляват ЕС, предназначени за моделиране на процеси и ситуации, за отговаряне на въпроси за оценка и класификация, за разпознаване на образи и т.н.

1.4. Обща структура на ЕС

Както вече споменахме, характерна особеност на ЕС е наличието на три нива – знания, програми и данни (фиг. 8).

Ще се спрем по-подробно на отделните части на ЕС.

БАЗА ОТ ЗНАНИЯ

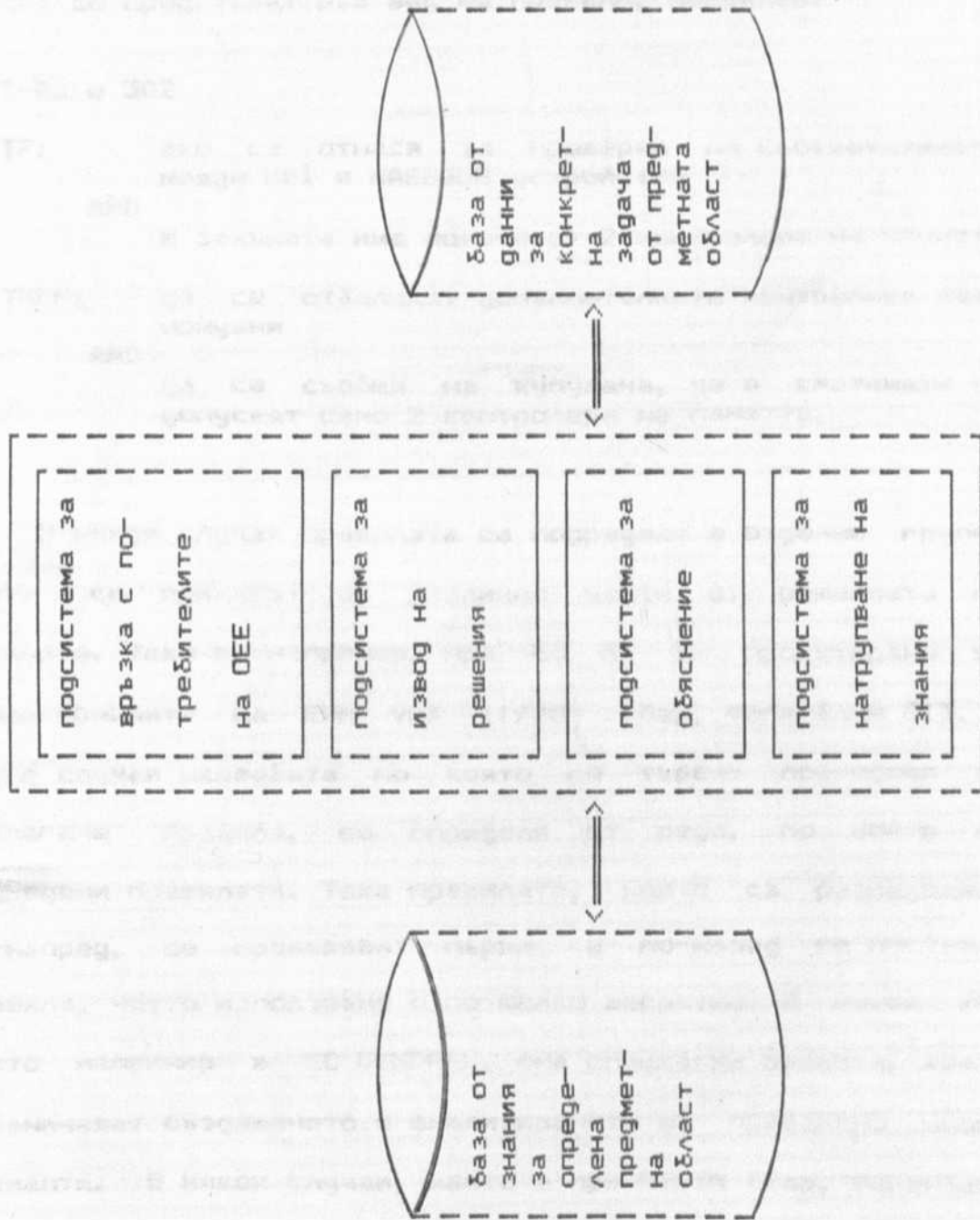
В БЗ са представени по еднороден начин знанията за предметната област, за която е създадена ЕС. При създаване на БЗ е необходимо да се осигури представяне както на родово-видовите и причинно-следствените връзки, така и на ситуативните (времени, пространствени и др.) отношения. По време на решаването на конкретна задача БЗ остава неизменяема и поради това може да се смята, че е разположена в дълговременната памет на ЕИМ [Уинстон 84]. Разбира се при създаването и усъвършенстването на ЕС е необходима възможността за постепенно натрупване на знания в БЗ, както и за промяна на знанията.

Ще се спрем на някои начини за представяне на знанията в БЗ. Най-напред ще разгледаме представянето на знания с помощта на продукции. Продукцията има три части. Първата част е нейното име. Втората част представлява условие, а третата – следствие. Под условие се разбира както единичното

ДАНИИ

ПРОГРАМИ

ЗНАНИЯ



Фиг. 8. Обща структура на ЕС.

условие, така и конюнкция, дизюнкция и отрицание от условия, ситуации, образци, факти и стойности на определени характеристики. Пог следствие се разбира едно или повече заключения или действия, които трябва да бъдат извършени. С помощта на такъв вид продукции знанията от предметната област се представят във вид на правила, например:

T-Rule 502

```

IF:      ако се отнася за проверка на съответствието
        между SBI и MASSBUS устройства
      AND
        в заявката има повече от 2 контролера на паметта

THEN:   да се отбележат допълнителните контролери като
        ненужни
      AND
        да се съобща на купувача, че в системата се
        допускат само 2 контролера на паметта.
  
```

В някои случаи правилата са подредени в отделни групи, които се прилагат за различни части от решението на задачата. Така е, например, при EC R1 за проектиране на конфигурациите на ЕИМ VAX 11/780 [Бар, Фегенбаум 81]. В други случаи наредбата по която се търсят подходящи за прилагане правила, се определя от реда, по който са подредени правилата. Така правилата, които са разположени по-напред, се проверяват първи, а по-назад се поставят правила, чието използване е по-малко вероятно. В някои ЕС, както например в EC DENDRAL, има специални правила, които ограничават създаването и анализирането на прекалено много варианти. В някои случаи, както е при MYCIN [Бар, Фегенбаум 82], в правилата има възможност да се указва и т. нар. фактор на увереност.

Фреймовете представляват друг начин за представяне на

знанията в БЗ. Такъв подход е използван, например, в ЕС SCHOLAR [Бар, Фегенбаум 82], предназначена за обучение на студенти по география. Фреймът представлява процедурно-декларативно описание на знанията за конкретен предмет или явление. Системата от фреймове дава възможност да се свързват знанията за отделните обекти и явления чрез родово-видови връзки и ситуативни отношения.

В някои ЕС, например в PROSPECTOR [Бар, Фегенбаум 82], [Дуга и др. 79], предназначена за геологични проучвания на евентуални находища на полезни изкопаеми, за представяне на знанията се използват семантични мрежи. Семантичните мрежи дават възможност за представяне на N-арни отношения (докато продукциите осигуряват представяне на бинарни отношения) между условие и следствие.

БАЗА ОТ ДАННИ

За да се приложат знанията от БЗ за решаване на конкретна задача, е необходимо да се въведат данни, които я определят. В зависимост от задачата тези данни могат да представляват визуална информация, стойности на определени характеристики, факти за задачата, хипотези и цели, които са поставени от потребителя и т.н. Този вид данни се представят в БД. По време на решение на задачата данните могат както да се променят (в резултат на обобщаването им или при откриване на неточност в данните), така и да се увеличават (в резултат на изводи, основани на знанията или като се генерират нови, междинни подцели и хипотези). Това показва, че данните са най-динамичната част от ЕС и може да се смята, че са разположени в кратковременната памет на ЕИМ

[Уинстон 84].

Необходимо е да се изясни поне неформално разликата между данни и знания. Докато данните са свързани до голяма степен със сетивното познание за съответната област или задача, то знанията представят абстрактното познание. От друга страна на определено ниво при решаване на една задача даден факт може да изразява знание, а на друго ниво - да представлява данни.

ПОДСИСТЕМА ЗА ИЗВОД НА РЕШЕНИЯ

Целта на тази подсистема е да свърже данните със знанията по такъв начин, че да изведе нужното решение. За това е необходимо да се открият и извлекат тези знания, които от една страна ще ни доведат до решението на задачата, а от друга - ще могат да се приложат към наличните данни.

Ще се спрем по-подробно на случая, когато знанията са представени с помощта на продукции. Подсистемата за извод на решения или интерпретаторът, както ще го наричаме по-нататък, трябва да избира последователно подходящи продукции и да проверява дали данните удовлетворяват тези продукции. В случая, когато търсим продукция, която се удовлетворява от наличните данни, извършваме действията, указани в следствието на продукцията (често това е запис в БД на нови данни или заключения) и повтаряме многократно този процес, докато стигнем до следствие, указващо крайния резултат (решението на задачата). В този случай имаме възходящ анализ. Интерпретаторът може да работи и по обратния път. На базата на първоначално поставената цел се търси продукция, която да съдържа тази цел като следствие и

по този начин от условието на продукцията се определя подцел или подцели, които трябва да са изпълнени, за да се изведе първоначалната цел. Тези подцели се записват в БД и с тях се постъпва по същия начин, докато текущите подцели не се окажат факти, въведени като условие на задачата. В този случай имаме низходящ анализ. В редица случаи е неудобно използването само на единия от двата начина. Тогава може да се използва комбиниран анализ, при който анализът започва от данните, стига се до някакви заключения и на базата на тези заключения се търси продукция, в която тези заключения участват като част от условието, а другата част от условието се разглежда като хипотеза, за която се прилага низходящ анализ, водещ обикновено до въпрос към потребителя за конкретни данни.

При действието на интерпретатора голяма част от времето се губи за търсене на подходяща продукция, която да се приложи. Поради това е необходимо да се предвидят специални средства за ускоряване на този процес. Едно просто, но не винаги възможно средство е продукциите да са подредени по реда на най-честото използване. Друг начин е продукциите да се разделят на групи, като всяка група съдържа продукции за част от процеса на решение на задачата. Възможна е и стратегия, при която се избират за проверка тези продукции, които изискват удовлетворяването на най-много условия; по този начин се очаква намаляване на броя на продукциите, които ще се използват. В някои ЕС се използва средство за маркиране на тези продукции, които на даден етап от решението на задачата се смятат за ненужни. Често проблемите, свързани с ускоряване на търсенето на нужните

продукции, са толкова важни, че се създава отделен блок за избор на стратегия.

Накратко ще споменем, че при представяне на знанията на базата на фреймове интерпретаторът има за цел да намира подходящ екземпляр на фрейм и да търси връзките между конкретни фреймове (ситуативните отношения). В този случай е необходим и механизъм за запълване на екземплярите на фреймовете.

ПОДСИСТЕМА ЗА ОБЯСНЕНИЯ

Задължителна част от ЕС са средствата за обяснение на взетите решения. Това се налага по следните причини:

а) При тестването на ЕС е необходимо да се открие кои знания предизвикват грешни решения.

б) При редовна експлоатация ЕС може да предлага алтернативни решения. Тогава потребителят ще се интересува от конкретните съображения за отделните решения.

в) В редица случаи ЕС се използват както за консултации, така и за обучение на специалисти. В последния случай обяснението за взетите решения е изключително важно.

г) За да бъдат убедителни решенията на ЕС (както и при решенията на хората), е необходима възможност за обяснение на взетото решение.

Обясненията на взетите решения, които дава подсистемата за обяснение, трябва не само да бъдат убедителни, но трябва и да са удобни за възприемане от потребителя, като най-добре е да се използва език, близък до естествения.

ПОДСИСТЕМА ЗА ВРЪЗКА С ПОТРЕБИТЕЛЯ НА ОГРАНИЧЕН ЕСТЕСТВЕН ЕЗИК

С ЕС общуват различен тип потребители. За да бъде това общуване ефикасно, е необходимо то да бъде удобно за потребителите. Един от начините за постигане на тази цел е използването на ограничен естествен език (ОЕЕ). Ще разгледаме различните дейности на потребителите, което определя до голяма степен ОЕЕ.

а) Най-големият клас потребители са хората, които използват ЕС като консултант или като учител. Или това са хората, които използват готовата ЕС за решаване на конкретни задачи. В този случай ОЕЕ трябва да осигурява лесното задаване на задачата от потребителя на ЕС и ясното даване на решението и обясненията от страна на ЕС.

б) Друг клас потребители, които на практика участват и в самото създаване на ЕС, са експертите, които въвеждат знанията. Макар и техният брой да е ограничен в сравнение с първия клас потребители, то необходимостта да им се осигурят удобни средства за извършване на тяхната работа е от първостепенна важност. Те трябва да могат на ОЕЕ да въвеждат знанията в БЗ, а при необходимост, след като ЕС вече се експлоатира, да коригират и разширяват тези знания.

ПОДСИСТЕМА ЗА НАТРУПВАНЕ НА ЗНАНИЯ

Тази част е свързана със създаването на БЗ. Тя трябва да осигури както първоначалното натрупване на знания посредством подсистемата за връзка с потребителя, така и да дава възможност за актуализиране и разширяване на БЗ. В редица ЕС е необходимо да се натрупват знания посредством

оценяване на правилността на взетите от ЕС решения, т.е. експертът формулира само степента на правилност на дадено решение, а подсистемата за натрупване на знания актуализира съответните знания.

В тази подсистема е необходимо да се предвидят и средства за проверка на непротиворечивостта на знанията, и за разделяне на знанията на отделни подгрупи, предвидени за различните етапи на решение на задачите.

1.5. Развитие на ЕС

До тук бяха разгледани някои основни характеристики на ЕС, като до голяма степен те бяха свързани с конкретни системи, предназначени за прилагане в определени области. Тенденциите в развитието на ЕС са насочени към разработка на такива ЕС или средства за бързо и лесно създаване на ЕС, които не са ориентирани към определени предметни области. Във връзка с това трябва да се отбележи огромното методологическо и практическо значение на езика PROLOG [Клоксин, Мелиш 82], който, макар и разработен сравнително неотдавна, получи голяма популярност. До голяма степен езикът PROLOG може да се разглежда като средство за създаване на определен тип ЕС, а именно на ЕС, в които представянето на знанията се извършва с помощта на продукции, а изводът на решението е низходящ. Разбира се тази универсалност на PROLOG се дължи на факта, че PROLOG е в крайна сметка език за програмиране и е предназначен за специалисти по информатика, което затруднява използването му от експерти – непрофесионалисти в програмирането. За да се

да даде възможност на експерта в съответната предметна област да въвежда знания без да се е обучавал предварително, е необходимо да се създаде съответната подсистема за връзка с потребителя. Тази подсистема е необходима и за работата на потребителите, които ползват ЕС като консултант. Към това трябва да се отбележи, че като система за програмиране PROLOG дава възможност да се реализират и други начини за представяне на знания и за извод на решения.

В момента се работи по създаването на т. нар. "празни" ЕС, които да се прилагат за различни предметни области. Успехите, които са постигнати до момента, се базират на усъвършенстването на създадени за конкретни области ЕС, така че те да могат да се използват и за други области единствено с промяна на БЗ [Буханан, Фегенбаум 78], [Ван Мелс 79]. Тук трябва да се отбележи, че начинът на представяне на знанията и изводът на решения са дефинирани предварително, още при създаването на "празната" ЕС. Разбира се, натрупаният опит би могъл да даде възможност за разработване на достатъчно мощни "празни" ЕС, в които потребителят ще може да дефинира, в зависимост от съответната предметна област, начина на представяне на знания и на извода на решения.

Този подход е привлекателен с голямата си универсалност, но крие в себе си големи опасности от загуба на ефективност и удобство (при експлоатацията) за получените по него конкретни ЕС.

Създаването на такава мощна "празна" ЕС се доближава много до другия подход в развитието на ЕС, който се състои в разработване на инструментални средства за създаване на ЕС, които да прераснат в инструментална ЕС [Дейвис, Ленат 82],

т. е. в ЕС, чиято област на приложение е проектирането и конструирането на конкретни ЕС.

У нас в областта на ЕС се работи в ЕЦММ при БАН, ИТКР при БАН, ВМЕИ "Ленин", БСНИПИ "ИНТЕРПРОГРАМА" и др.

Работата в страната се координира от Програмен съвет по "Експертни системи" към Националния съвет по автоматизация. Тя е съгласувана с концепциите и плановете на СИВ по ЕС.

ЕС заемат основно място в тематиката на Работна група 22 на Комисията по научни въпроси на изчислителната техника към Академиите на науките на социалистическите страни, в чиято работа участвува и нашата страна.

У нас са създадени или са в процес на създаване редица ЕС. Ще споменем само някои от тях: ЕС за разпознаване и класификация на археологическа керамика [Томов, Сахно 85], ЕС за анализ на антични историко-географски извори за древна Тракия [Томов, Тачева, Григоров 87], ЕС в областта на биофизиката за електросливане на клетки [Димитров, Томов, Шарков, Ангелова 87], ЕС за диагностика на широк клас технически обекти [Сгурев и др. 86], ЕС за автоматизиране на проектирането в машиностроенето [Димитров и др. 87], ЕС за планиране и управление на селското стопанство, ЕС в областта на медицината за диагностика, за консултации по реаниматология и др.

В заключение ще отбележим, че ЕС представляват широко поле за научни изследвания и независимо от някои недостатъци имат много и важни приложения в практиката.

2. ИНТЕЛЕКТУАЛИЗИРАНИ ПРИЛОЖНИ ПРОГРАМНИ СИСТЕМИ ЗА МЕМБРАННАТА БИОФИЗИКА.

2.1. Основна концепция

Най-важната характеристика на ИППС [Томов, Геров 87] е, че те използват знания за съответната проблемна област. Знанията са отделени от програмите в БЗ и за тяхното представяне се използват "по-свободни" формализми (правила, семантични мрежи, фреймове и т. н.) от традиционните системи, където знанията се представят във вид на алгоритми. Алгоритмите за решаването на задачите при ИППС се съдържат неявно в знанията, в принципите на логическия извод и в данните, т. е. използват се напълно предимствата на известните от ИИ "системи, базирани на знания".

От тази гледна точка "класическите" експертни системи за решаване на редица конкретни задачи в определена проблемна област могат да са разглеждат като проблемно-ориентирани ИППС. В случая е избран подхода, описан в гл. I т. 3, базиращ се на създаване на предметно-ориентирани празни ИППС и конкретизирането им до проблемно-ориентирани ИППС.

Мембранната биофизика изучава физичните принципи на организацията и функционирането на биологичните мембрани. В последното десетилетие значително се задълбочиха изследванията върху ефектите на физическите въздействия и по-специално на електрическите полета. В резултат на това се развиха едни от най-перспективните направления на мембранното инженерство – електросливане на клетки,

електропорообразуване, клетъчна диелектрофореза и електроформиране на липозоми. Всяко от тези направления има директни приложения в съвременните биотехнологии. Затова от изключителна важност е високата ефективност на научно-изследователската работа за разкриване и изучаване на механизмите на мембранните взаимодействия, на поведението на биологичните обекти при различни условия и въздействия и т. н. За подпомагане на тази дейност бяха създадени експериментални версии на ЕС за някои основни направления – PREFES (PRotoplast ElectroFusion Expert System) за електросливане на протопласти и LIPES (LIposome Production Expert System) за получаване на липозоми [Димитров, Томов, Шарков, Ангелова 87], [Шарков, Димитров 87].

Независимо от изключителната сложност и разнообразие на изследователската и експерименталната дейност в различните направления на мембранната биофизика, ясно се открояват следните особености: във всички конкретни направления се отчита действието на някои общи биофизични, физични или физикохимични закони и зависимости; моделите на различните процеси обикновено се базират на сходен математически и физикохимичен апарат; специалистите и изследователските колективи обикновено работят в няколко сродни направления; лабораторната апаратура обикновено обслужва едновременно експерименталната дейност в различни направления и т. н.

Поради тези общи особености и натрупания опит при създаване на експерименталните ЕС се оказва удачно разработването на обща предметно-ориентирана "празна" ИППС (ЕС) в областта на мембранната биофизика [Томов, Шарков 88]. Най-общо тя се състои от:

- обща БЗ за предметната област с йерархична и модулна структура;
- подсистема за организиране на общите методи за извод;
- подсистема за конструиране на конкретни ЕС в различни тесни направления;
- подсистема за извличане и натрупване на знания;
- подсистема за организиране на потребителски интерфейс;
- широк набор от инструментални средства.

На нейна база могат по-лесно да се разработват ЕС (проблемно-ориентирани ИППС) за различните направления на предметната област. Това се извършва, като с помощта на подсистемата за извличане и натрупване на знанията се допълни БЗ със съответните проблемно-ориентирани знания (използвайки издраните формализми за представяне на знанията), а с помощта на другите подсистеми се реализират съответните механизми за логически извод, потребителския интерфейс и другите проблемно-зависими компоненти.

Поради спецификата на предметната област е удачно разработването и на специализирана инструментална среда за създаване на ЕС в областта на мембранната биофизика, която включва: език за обектно-ориентирано програмиране и интерпретатор; език и интерпретатор за логическо програмиране; език и интерпретатор за системи продукции; апарат на основата на размитите множества; средства за създаване и поддържане на времеви модели на реалните биофизични процеси; средства за реализиране на размит логически извод; набор интерфейсни средства за връзка със СУБД, аналитично-изчислителна система, графична система,

система прозорци и менюта и др. Тази инструментална среда ще се използва от една страна за разработване на самата "празна" предметно-ориентирана ЕС, а от друга – на базата на своята модулност ще служи като основа за автоматизираното създаване на проблемно-ориентираните ЕС.

Предложеният подход значително улеснява работата по създаването на няколко ЕС в различни направления на мембранната биофизика. От друга страна, празната предметно-ориентирана ЕС може да се използва и при разработване на подобни празни системи в други сродни предметни области. Поради сравнително общия си характер, инструменталната среда от своя страна може да се използва като основа за създаване на най-различни системи, базирани на знания. Понастоящем се извършва разработването на празна предметно-ориентирана ЕС за мембранната биофизика и специализираната инструментална среда на базата на езика Golden Common LISP за IBM PC – AT.

2.2. Експертни системи за получаване на липозоми и електросливане на протопласти

Представени са експертни системи за планиране на биофизичните експерименти по електросливане на протопласти (PREFES) и получаване на липозоми (LIPES) чрез използване на правила и фреймове [Димитров, Томов, Шарков, Ангелова 87].

Липозомите са затворени мембранни структури. Използват се широко за моделиране на клетки и имат големи възможности за медицината и биотехнологиите. Липозоми могат да бъдат

получавани чрез голямо разнообразие от методи. Основен проблем при получаването на липозоми е как да се получи висок добив на липозоми с предварително определено разпределение по размер и физикохимични свойства. Протопластите са растителни клетки без клетъчни стени. Сливането на протопласти може да произведе хибридни клетки и, съответно, нови растения. Главна задача на клетъчното сливане е висок добив на жизнени хибриди за минимално време, разноси и повреди на клетките.

Механизмите на двата процеса, образуването на липозоми и електросливането на протопласти почти не са известни [Димитров 83], [Димитров, Джейн 84], [Ангелова, Димитров 86], [Димитров 86], [Цимерман 82]. Няма стандартна процедура, приложима за всички случаи на различни липозоми и протопласти при различни условия. Получаването на липозоми и електросливането на протопласти могат да бъдат проведени успешно само от опитни специалисти. Ето защо са избрани експертните системи като средство за улесняване на планирането на биофизичните експерименти.

По-долу са описани две системи за планиране на биофизичните експерименти: PREFES – за протопластно електросливане и LIPES – за получаване на липозоми.

2.2.1. Естество на проблема и необходимост от съвет

Електросливането се състои от два главни етапа: плътно доближаване на клетките чрез нехомогенно променливо поле (дieleктрофореза) и прилагане на постояннотоккови импулси с малка продължителност и голяма амплитуда за дестабилизиране

и сливане на мембраните. Диелектрофорезата зависи от свойствата на клетъчните мембрани (заряд, геометрия, структура) и средите (проводимост, тоничност), както и от приложеното електрично поле (напрежение, честота, продължителност). Електростабилизирането и сливането на мембраните зависи от напрежението, продължителността и броя на постоянните токови импулси и физикохимичните свойства на мембраните и заобикалящите среди. Например, ако проводимостта на средата е висока, то диелектрофорезата е невъзможна, следователно сливането е невъзможно.

Получаването на липозоми включва набухване на липизи във водни разтвори и прилагане на външни въздействия. Методът за електрополучаване на липозоми се базира на набухване на липизи във външни електрични полета. Добивът и свойствата на липозомите зависят значително от състава на липида, дебелината на липидния слой, средата и електричното поле. Получаването на липозоми с желаните свойства е възможно при доста тясна област от условия. Например, образуването на липозоми е невъзможно в разтвори с висока йонна сила. Освен това, по време на хидратацията могат да се образуват различни нелипозомни структури.

Електросливането на клетки и получаването на липозоми са доста подобни по отношение на физикохимичните механизми и прилаганите електрически полета, но се различават по крайния "продукт". В допълнение, различните обекти и процедури изискват различен анализ и планиране на експериментите. Затова двата процеса могат да бъдат представени най-напред чрез два различни модела, които в последствие да бъдат обединени.

Понастоящем много фирми продават липозоми и апарати за клетъчно електросливане и получаване на липозоми. Някои изследователски лаборатории използват техни собствени средства за получаване на хибриди чрез електросливане и за получаване на липозоми. Въпреки това, само малък брой изследователи имат необходимия опит, за да осъществят успешно тези процеси. Проблемът за намиране на подходящата процедура за клетъчно сливане и получаване на висок добив от желаните хибриди може да бъде решен само от няколко експерта. Изглежда, че методът за електросливане придобива все по-голяма популярност сред учените и изследователите. В близко бъдеще необходимостта от експерти в областта на електросливането на липозоми ще нараства силно. Това оправдава усилията за създаване на експертни системи (или други типове ИППС за подпомагане на изследователите.

Тези ЕС могат да служат за следните цели:

- да предсказват поведението на биофизичната система при дадени условия;
- да планират експерименти при зададени цели;
- да обясняват своите "разсъждения";
- да създават модели на нови научни резултати;
- да обучават потребителя.

2.2.2. Експертните системи PREFES и LIPES

а) Обща структура.

Основните компоненти на PREFES са: база знания, интерпретатор, база данни. Тя има още три допълнителни модула: управление на базата знания (УБЗ), потребителски

интерфейс и справки по анализа. LIPES има подобни компоненти. Общата структура на PREFES е показана на Фиг. 9.

б) Представяне на знанията и данните.

PREFES и LIPES съдържат много факти, зависимости и друга информация. Тези знания са представени чрез фреймове, правила и таблици в съответни БД.

Знанията за елементите на биофизичната система (протопласти, среда, променливо поле, диелектрофореза и т. н.) са структурирани във фреймови структури [Чинстон 84], [Чинстон 84а]. Фреймовете слотове съдържат експертната информация за свойствата на тези обекти. Различни информационни клетки са използвани за улесняване на представянето на голямото количество от фактически материал - дефинициите на свойствата, техните възможни стойности, въпросите към потребителя, препоръчваните предварителни експерименти и т. н. Например, средата за диелектрофореза е представена в системен фрейм по следния начин:

(MEDIUM

(TEMPERATURE (EXPECT (NUMB-IN 0 40))

(UNITS (degrees))

(TYPE (primitive))

(QUESTION (standard))

(USED-IN (R15 R21...))

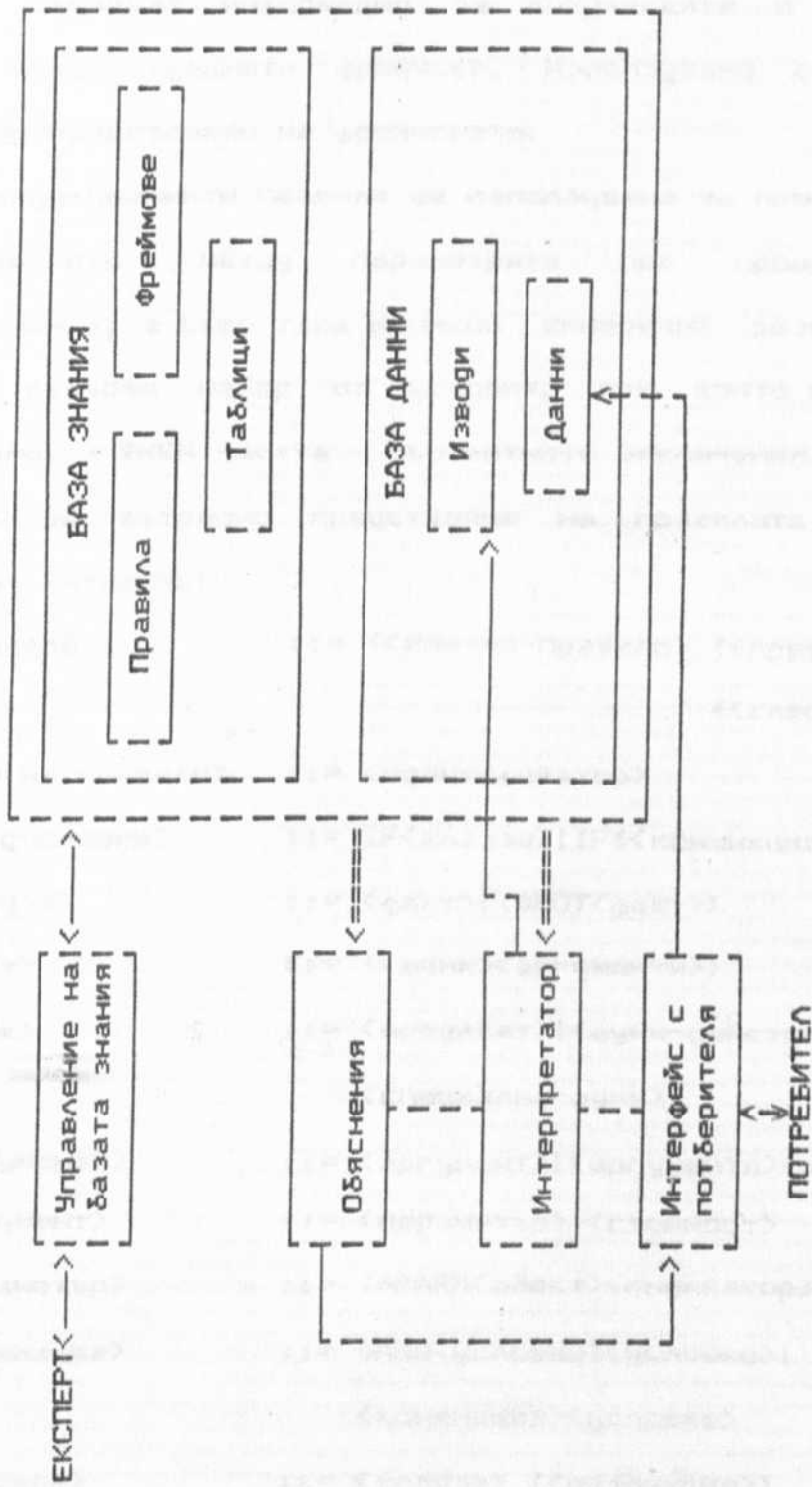
(TONICITY (EXPECT (ONE-OF

(isotonic hypotonic hypertonic
slightly-hypotonic)))

(TYPE (system))

(CONCLUDED-IN (R63 R67...))

(USED-IN (R15...))



Фиг. 9. Блок-схема на PREFES.

Има още две специални клетки, CONCLUDED-IN и USED-IN, които свързват параметрите с процедурите за откриване и използване на техните стойности. Те съдържат списъци с имена на съответните правила. В допълнение, няколко фреймови клетки съдържат информация за логическите и структурни връзки между отделните фреймове. Използувано е подходящо вътрешно представяне на фреймовете.

Продукционните правила са използвани за представяне на зависимостите между параметрите при провеждане на експеримента, а също така и някои емпирични резултати. IF частта съдържа набор от условия, при които правилото е приложимо, а THEN частта – съответните заключения. Създаден е език за вътрешно представяне на правилата (RIRL) със следния синтаксис:

```

<правило> ::= (<име-на-правило> (<предусловие>
                (<следствие>))
<име на правило> ::= <идентификатор>
<предусловие> ::= IF<клауза>| IF(<конюнкция>)
<клауза> ::= <факт>| ($NOT<факт>)
<факт> ::= (<функ><аргументи>)
<функ> ::= <предикат>|<функ-действие>|
          <специална-функ>
<аргументи> ::= <аргумент>|<аргументи><аргумент>
<аргумент> ::= <параметър>|<стойност>
<параметър> ::= ($PARAM<обект><индикатор>)
<конюнкция> ::= $AND<условие><условие>|
              <конюнкция><условие>
<условие> ::= <клауза> (<дизюнкция>)
<дизюнкция> ::= $OR<клауза><клауза>|

```


<дизюнкция><клауза>

<следствие> ::= THEN<факт>|

<следствие><факт>

Както е показано, условията на правилото е или отделна клауза, или конюнкция от клаузи. Клаузата представлява факт или отрицание на факт. Дизюнкции от условия могат да бъдат представени чрез отделни правила с еднакви THEN части. В условието могат да се изразяват и конюнкции от факти. За оценяване стойностите на параметрите в IF частта се използват предикатни функции от рода на IS, LESS, BETWEEN и т.н.

Тези конвенции налагат известни ограничения, но осигуряват възможност за представяне на произволен булев израз в едно или повече правила. Следва пример на едно правило в RIRL форма и формулировката му на български език:

```
(R31 (IF ($AND (IS ($PARAM de collection) possible)
              (EMPTY ($PARAM pp damages))
              (EMPTY ($PARAM de side-effects))
              (IN-TAB-1 ($PARAM pp species)
                       ($PARAM dc-pluse amplitude)
                       ($PARAM dc-pluse duration))
              ...))
      (THEN (CONCLUDE ($PARAM ef result) possible)))
```

R31:

AKO диелектрофорезата (DE) е възможно и протопластите (PP) не са разрушени и няма странични ефекти на диелектрофорезата (side-effects) и амплитудата и продължителността на импулса (DC) са съответни за вида протопласти и ...

ТО електросливането (EF) е възможно.

За съхранение на правилата са използвани специални видове фреймове. В допълнение към "логическата" информация те съдържат и клетки с данни за автора на правилото, литературата, датата, текста на английски език и т. н. [Файкс, Келер 85]. Други клетки пазят информация за статуса и състоянието на правилата.

Част от биофизичната информация е представена в таблици. Те съдържат подробен запис за зависимостите между стойностите на параметрите. За тяхното използване са дефинирани специални функции.

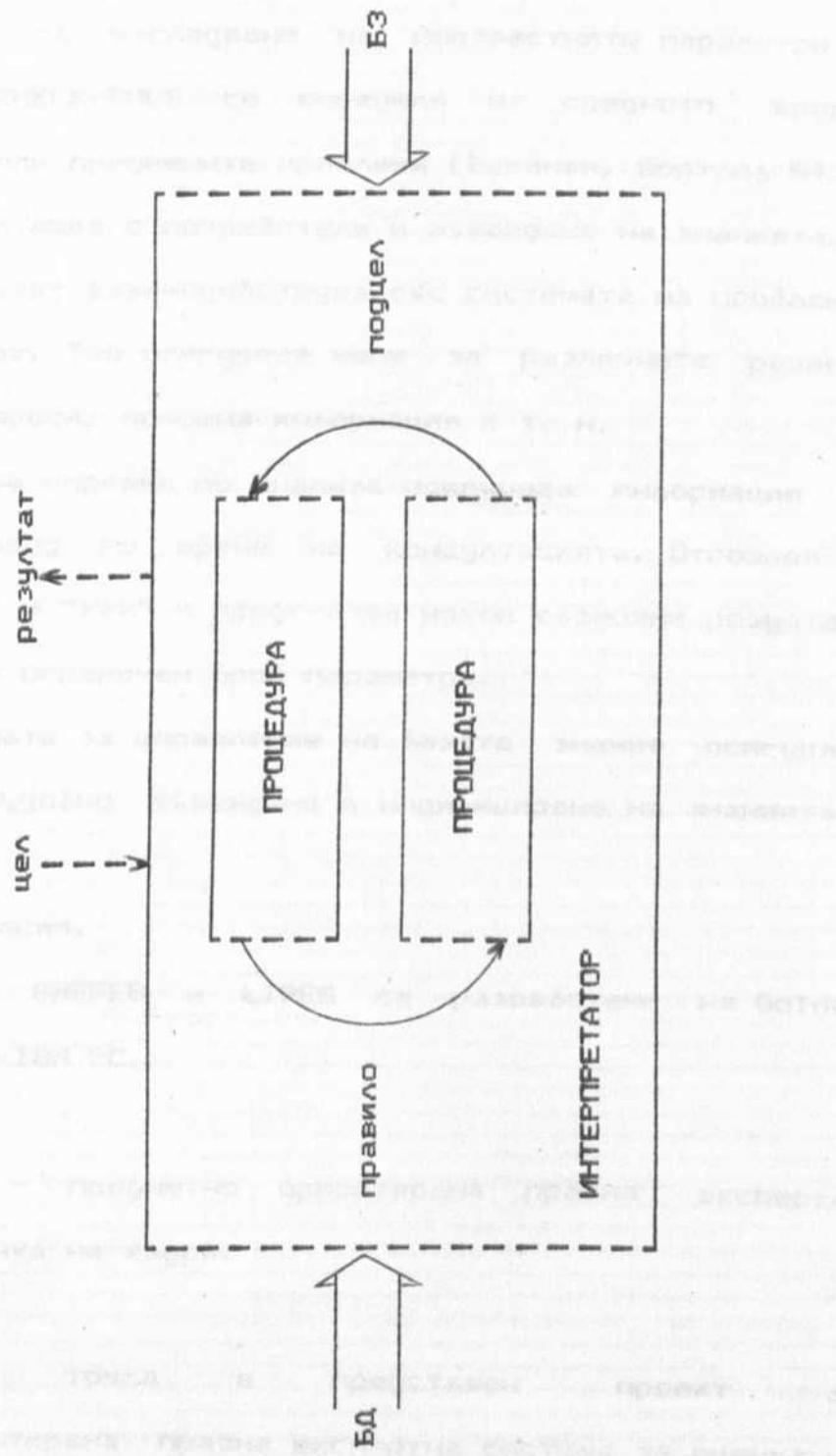
Базата от данни съдържа цялата информация за конкретния биофизичен експеримент чрез използване на асоциативни тройки (обект - индикатор - свойство).

в) Интерпретатор.

Основната задача на интерпретатора е да направи заключения за поведението на биофизичната система на базата на конкретните стойности на параметрите, използвайки правилата и фреймовете от базата знания.

Управлението на интерпретацията е от целите към данните. Във всеки момент интерпретаторът работи за откриване стойността на някой параметър, както е показано на фиг. 10.

Потребителският въпрос относно резултата от експеримента извиква процедурата SEARCH-FOR с аргумент (electrofusion result). Тогава за всяко правило, което води до тази цел, процедурата SEARCH-FOR прилага процедурата CHECK-RULE. Тази процедура оценява условията и ако те са удовлетворени, прави съответните заключения. Ако тя открие параметри в условието с все още неизвестна стойност, отново се извиква (рекурсивно) процедурата SEARCH-FOR. В зависимост



Фиг. 10. Функционална схема на интерпретатора.

от типа на параметрите се задават и въпроси към потребителя.

За да се извършва търсенето по-ефективно, процедурата SEARCH-FOR използва правилата по реда на тяхната използваемост. Коефициентът на използваемост изразява "опита" на системата от предишните консултации. За същата цел и редът за изследване на неизвестните параметри от процедурата CHECK-RULE се определя от средното време, изразходвано при предишните проблеми [Буханан, Шортлиф 84].

г) Консултация с потребителя и въвеждане на знанията.

Потребителят взаимодействува със системата на проблемно ориентиран език. Той осигурява меню за различните режими, разбираеми въпроси, помощна информация и т. н.

Модулът за справки по анализа осигурява информация за логическия извод по време на консултацията. Отговаря на въпроси "защо" и "как" и предсказва някои възможни резултати при промяна на ограничен брой параметри.

Подсистемата за управление на базата знания осигурява средства за удобно въвеждане и модифициране на знанията в БЗ.

г) Реализация.

Системите PREFES и LIPES са разработени на Golden Common LISP за IBM PC.

3. ЕСОК - предметно ориентирана празна експертна система за оценка на кадри.

В тази точка е представен проект за предметно-ориентирана празна експертна система за оценка на кадри [Томов, Геров, Григоров, Колчев 88]. Тази система е

изградена на базата на концепцията за интелектуализирани приложни програмни системи [Томов, Геров 87].

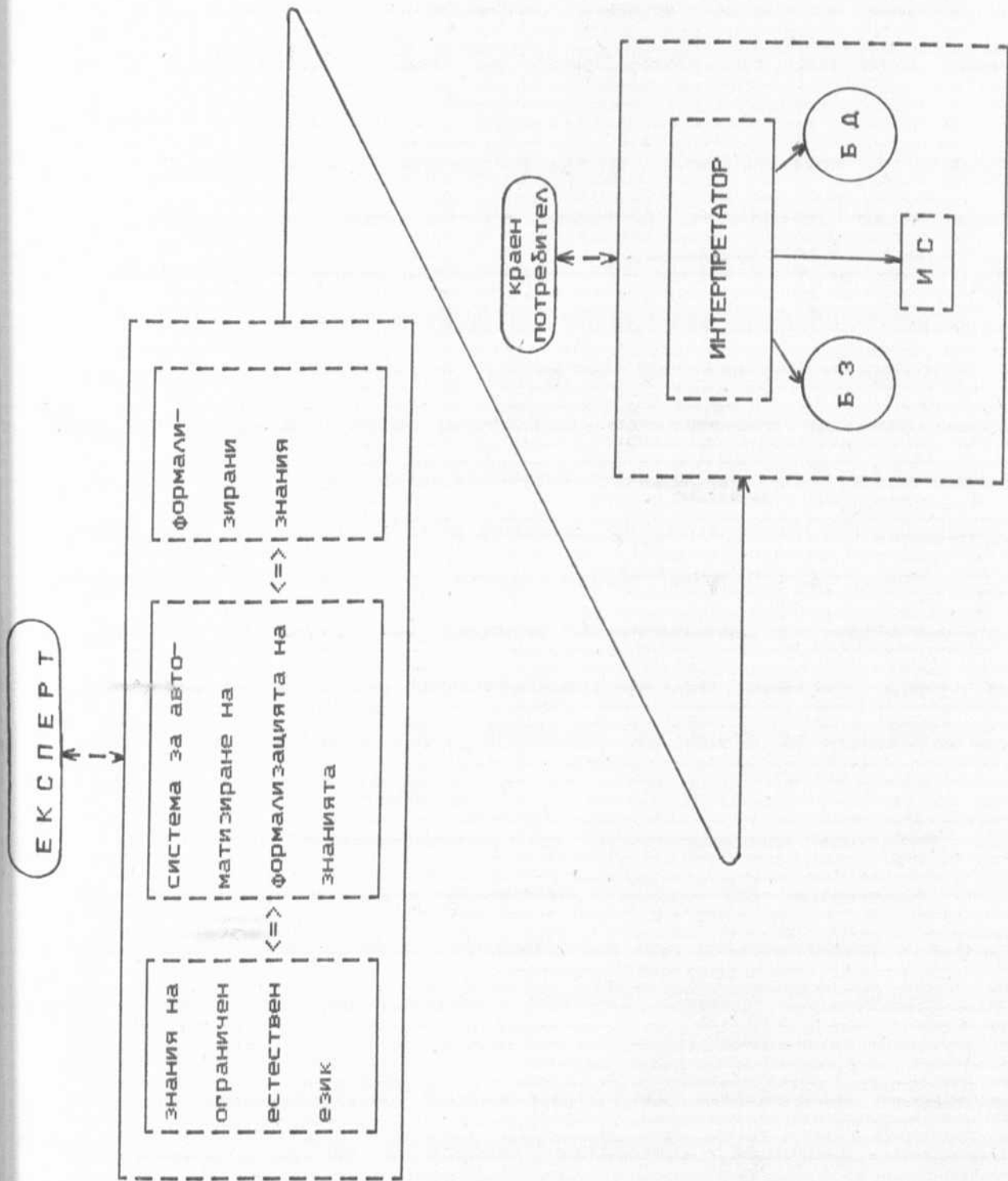
Идеята за създаване на такава система е дадена от В. Томов и П. Данев. По-нататъшната разработка и реализация се извършват от Ал. Геров, К. Колчев и П. Данев (пог общото ръководство на В. Томов).

Предметната област на системата е оценка на кадри, независимо дали това са личности, колективи, стопански ръководители, специалисти, надарени деца или партийни кадри.

ЕСОК се състои от две части (фиг. 11). Първата от тях включва база знания, база данни – контекст, информационна система (ИС) и интерпретатор на знанията.

Другата част на ЕСОК се състои от подсистема за формализация и натрупване на знанията и е предназначена за използване при конкретизиране на ЕСОК до ЕС за определен проблем.

БАЗАТА ЗНАНИЯ на ЕСОК е частично запълнена с включените в нея знания, които са общи за предметната област. Една част от знанията се отнасят за оценката на кадрите и за принципите на оценяването. Друга част от знанията описва методите за използване на данните от ИС, БД, тестовете и анкетите. Всичките тези знания са за предметната област. В тази група са включени и знания, свързани с основните характеристики на личността, с основните принципи на оценяване, знания за обработката на данните от тестовете и анкетите. На практика само тези знания се намират в БЗ. При конкретизацията (проблемното ориентиране) на ЕСОК се добавят знанията, свързани с типа на кадрите, които се оценяват.



Фиг. 11. Обща схема на експертна система за оценка на кадри.

Според специалистите в предметната област, най-подходящ за оценка на кадрите е методът на тестове и анкети [Попов 78]. Тъй като тестовете и анкетите се състоят предимно от въпроси-критерии, а това по своята същност са правила, в ЕСОК е възприета схема за представяне на знанията чрез правила.

Поради спецификата на областта на приложение - оценка на кадри, не бива да се допуска въвеждане от крайния потребител на дадена информация, при условие че тя може да се изведе. Това означава, че въпроси към крайния потребител се задават само за факти, които не се явяват следствия в някои правила. Това осигурява обективност на получения резултат. По този начин крайният потребител няма възможност да дава субективна информация, да се намесва в процеса на извода на оценката и да влияе върху модела за оценяване, който е възприет и заложен в системата. От друга страна това, че въпроси към потребителя ще се задават само за факти, до голяма степен облекчава системата за обработка на потребителския отговор.

С оглед повишаването на ефективността на ЕСОК се въвеждат отделни групи продукции, които се използват при получаване на крайни (положителни или отрицателни) оценки, при оценки с особено висока важност (което се указва от потребителя) и др.

В процеса на оценяването на конкретния човек е необходимо да се използват критерии, засягащи неговата професионална дейност, длъжностната му характеристика, трудовия му стаж и т.н. Това обуславя прилагането на конкретен набор от критерии във всеки отделен случай. Затова

2. ДРУГИ ЕЗИЦИ И СИСТЕМИ

Езикът ПРОЛОГ

Друг език за програмиране, който се използва като инструментално средство на изкуствения интелект е езикът ПРОЛОГ [Клоксин, Мелиш 82]. Той е създаден около 1970 г. от А. Колмероер и неговите сътрудници от Университета в Марсилия. ПРОЛОГ е основан на подмножество на класическата логика, от където идва и името му (PROgramming in LOGic, т. е. ПРОграмиране в ЛОГиката).

Характерни особености на езика са следните:

- интерактивен език за обработка на символна информация;
- съчетава в себе си някои от основните методи на изкуствения интелект като съпоставяне по образец, обхождане с връщане назад (backtracking), логически извод и генериране на алтернативни решения;
- език от високо ниво, т. е. в него няма машинно-ориентирани концепции;
- програмата и данните са идентични по форма;
- програмата освен процедурна семантика има и декларативна семантика.

Всичко това прави ПРОЛОГ едно мощно инструментално средство за разработване на програми с изкуствен интелект.

През последните години ПРОЛОГ стана много популярен и намира приложение в редица области като:

- създаване на експертни системи;
- създаване на релационни бази данни;
- автоматично доказателство на теореми в математическа-

та логика;

- създаване на интерфейс на ограничен естествен език;
- планиране на действията при работи;
- създаване на системи за аналитични преобразования;
- създаване на транслатори;
- други области на изкуствения интелект.

Езикът FRL

На края ще се спрем на една от най-известните реализации на система за работа с база фреймове – езикът FRL (Frame Representation Language) [Мински 74], [Робертс, Голдшейн 77], [Поспелов 86]. Базата знания се състои от система фреймове, представени като списък, и система от ЛИСП-функции за обработка на фреймовете и данните. В FRL фреймът се състои от пет нива – слот (slot), фасет (facet), данна (datum), етикет – наречени още коментар (label, comment) и съобщение (message). Фреймът се представя чрез списък и има следния вид:

```
(fr1
  (slot1 (facet1 (datum1 (label1 (message1 ... messageN)
    - DEFAULT - в който label2 ( . . . . . )
    . . .
    - SIF-INDEX и datum2 ( . . . . . )
    . . .
    facet2 ( . . . . . )
    - SIF-INDEX . . . в който се съдържа SIF-INDEX )
    slot2 ( . . . . . )
    . . .
  )
```

Фреймовете служат предимно за описване на структурата на понятията (обектите) и на процедурите, които трябва да се изпълнят при работа с тях.

Обикновено името на фрейма съвпада с името на понятието, имената на слотовете съвпадат с имената на свойствата, характеризиращи понятието, а съдържанието на фасетите показва вида и структурата на стойностите на съответните свойства и процедурите и предикатите, които трябва да се включат в действие.

Конкретизацията на стойностите на свойствата води до конкретизация на целия фрейм, който вече описва определен конкретен представител на понятието.

Такъв конкретизиран фрейм се нарича екземпляр на съответния фрейм.

Всеки фрейм в това представяне притежава два системни слота – AKO (A Kind Of), указващ връзките нагоре в системата и INSTANCE, съдържащ имената на екземплярите (т. е. връзките надолу). Освен това, всеки слот има по шест фасета:

– \$VALUE – в който се съдържа стойността на дадения слот;

– \$DEFAULT – в който се съдържа стойността на слота по подразбиране;

– \$IF-ADDED и \$IF-REMOVED – в които се съдържат ЛИСП функции, които се използват когато в слота се добавя или премахва информация;

– \$IF-NEEDED – в който се съдържа ЛИСП – функция, обработваща стойностите на слота;

– \$IF-REQUIRE – в който се съдържат предикати, които се прилагат върху стойностите на слота.

Очевидно, горепосочените структурни и процедурни възможности на езика FRL осигуряват удобство и ефективност както при формиране на фреймове, така и при работа с тях.

Системи за обработка на знания

Вследствие големия интерес към използването на ИППС и по-специално на ЕС в различни области на практиката, в последно време се появиха редица инструментални програмни средства насочени специално към обработка на знания и създаване на експертни системи. За разлика от езиките за програмиране от типа на LISP и PROLOG, използвани за представяне на знания, тези системи са предназначени специално за представяне и обработка на знания. Системите от този вид са снабдени със средства, обезпечавщи специфичните дейности, свързани не само с представянето но и с обработката на знанията.

Първата група такива инструментални системи за представяне и използване на знания са така наречените празни експертни системи. Празните ЕС, разглеждани в чист вид, представляват ЕС, в които БЗ не е запълнена - т. е. интерпретатор и празна, но подготвена и програмно обезпечена информационна структура, въз основа на която се изгражда БЗ. Използването на празни ЕС е много удобно особено за приложни цели, тъй като създаването на ЕС на базата на празна ЕС се състои само в попълване на БЗ. Разбира се, в някои по-мощни празни ЕС са предвидени и възможности за създаване на потребителски процедури за обработка на знанията, за интерфейс и т. н., но една по-проста ЕС може да бъде създадена само чрез попълване на БЗ, т. е. с минимум усилия.

Празните ЕС имат и своите недостатъци. Един от тях е, че схемата за представяне на знания е вградена "твърдо" в празната ЕС, което донякъде затруднява прилагането ѝ в по-сложни или по-необичайни области. Например празна ЕС, базирана на правила, е неудобна за използване в области, където знанията са йерархични и имат много връзки помежду си. Този недостатък се избягва чрез създаването на отделни празни ЕС, обезпечавщи различни формализми за представяне на знания и дори техни комбинации, което на практика означава наличие за всяка приложна област на празна ЕС, в която схемата за представяне на знания е подходяща за областта.

Друг недостатък на по-ранните празни ЕС е това, че освен "чистата" ЕС – т. е. интерпретатор, БЗ и контекст, те обезпечават твърде малко други функции. Вграждането на едни или други помощни функции и странични възможности в една празна ЕС по принцип е полезно. Но празните ЕС се създават без да се знае областта на тяхното приложение и веднъж създадени, те трябва да могат да се използват за най-разнообразни приложни задачи. Вграждането на помощни функции и обезпечаването на допълнителни възможности за всевъзможни области на приложение е на практика невъзможно, тъй като би направило системите огромни и тежки за работа. От друга страна, избирането и реализирането на тези от възможностите които са относително универсални е трудна и практически неразрешима задача. На практика, при вграждането в празните ЕС на допълнителни функции и възможности, се търси оптималната граница между удобството (т. е. наличието на такива функции) и ефективността (т. е. затрудненията при

използването на празните ЕС, до които води включването на много такива функции).

Както се вижда от казаното дотук, съществен недостатък на празните ЕС е това, че те трудно се поддват на ориентирани към различни предметни области. Другият подход – това е осигуряване на инструментални средства за създаване на функции, възможности и пр., необходими за различни предметни области. Това на практика е идеята на другия тип инструментални програмни системи за обработка на знания наречени среди за обработка на знания. Такива са например системите OPS – 5, OPS – 83, GURU, KEE, както и българският продукт "Интерексперт" създаден в БСНИПИ "Интерпрограма".

За разлика от празните ЕС, средите за обработка на знания поддържат библиотека от различни части на ЕС, както и инструментални средства за комбиниране на тези части и за създаване на нови функции и възможности. Системата OPS-5 например съдържа реализирани осем различни стратегии за търсене в система продукция, като на потребителя е дадена възможност за избор коя от стратегиите да се използва.

Наличието на голям брой различни фрагменти от ЕС – заготовки и на инструментални средства за работа с тях и за създаване на нови такива дава възможност за много гъвкава работа със средите за представяне и използване на знания и създаване чрез тях на ЕС, максимално адекватни към областта на приложение. Разбира се, комбинирането на отделните елементи, както и създаването на нови такива е свързано с известен програмистки труд, но от една страна ефективността на създадените по този начин системи е голяма, а от друга страна съвременните среди за обработка на знания са толкова

развити (например системата KEE или системата PERSONAL CONSULTANT), че работата с тях е лека и изисква относително по-малко усилия. Метод за повишаване на ефективността на работата и облекчаване на създаването на ЕС на базата на универсални среди за създаване на ЕС е специализацията на средите за създаване на ЕС. По този начин например се предвижда да се реализира ИППС за оценка на кадри ЕСОК. Предвижда се системата ЕСОК да бъде създадена като надстройка на средата за обработка на знания "Интерексперт", създадена в БСНИПИ "Интерпрограма". Специализацията ще се състои с частично запълване на БЗ със знания, общи за всички задачи свързани с оценката на кадри, както и със специални функции, обезпечавщи някои специфични дейности свързани с оценката на кадри, като например обработка на тестове, анкети и др. Изградена по този начин, системата ЕСОК ще направи създаването на конкретни ЕС, предназначени за конкретни задачи в областта на оценката на кадри (например атестиране, оценка на възможностите за творческо развитие на надарени деца, оценка на стопански ръководители и др) много по-лесно, отколкото създаването на тези ЕС на базата на универсалната среда за създаване на ЕС "Интерексперт".

ГЛАВА IV. ЕКСПЕРТНИТЕ СИСТЕМИ КАТО ИППС

1. ЕКСПЕРТНИ СИСТЕМИ И ПРИЛОЖЕНИЯ – ОБЩИ СВЕДЕНИЯ

Експертните системи представляват мощно средство за представяне и обработка на знания. Това определя важното място, което те заемат във фундаменталните изследвания. Освен това, ЕС имат много и най-различни приложения, които са може би главната причина за големия интерес към тях в целия свят.

Какво представлява ЕС? Това е програмна система, която разполага с експертни знания за дадена област. В зависимост от предназначението си, използвайки тези знания, ЕС може да интерпретира данни, да отговаря на въпроси, да дава съвети, да извършва диагностика, да планира, да проектира, да прогнозира, да консултира и т.н. [Томов, Геров 87а]. Най-общо казано, ЕС може да извършва някои от дейностите, характерни за експерта в дадената област. По такъв начин ЕС икономисва време и средства и допринася за повишаване на качеството на научно-приложната и производствена дейност. Основна трудност при създаването на ЕС е формализирането на знанията за дадената област, т. е. представянето им във вид, който позволява въвеждането и обработката им в ЕС. Това формализиране на знанията се извършва съвместно от специалисти информатици и специалисти в съответната област

при конкретизацията на ЕСОК е необходимо да се въведе и група правила за избор на последователността от критерии за оценяване. Тази група може да се разглежда като отделна система от правила, като "миниекспертна система", вградена в ЕС.

В БАЗАТА ДАННИ (контекста) се съхраняват текущите данни за оценявания обект. Данните се набират от ИС и от отговорите на оценявания. Поради това, данните от отговорите на оценявания и данните от ИС е необходимо да се представят на еднакъв вътрешен език. Тези данни от БД, които диха били необходими за по-нататъшна работа на системата (при друго оценяване), се прехвърлят в ИС след прикличване на работата, т.е. след изработване на оценката.

ИНФОРМАЦИОННАТА СИСТЕМА съдържа обща информация за кадрите, данни за характера на извършваната от тях работа, данни за предишни оценки и т. н. На базата на тази информация се намират автоматично значенията на редица листа от дървото на знанията без да се задава въпрос към оценявания.

Основен проблем при създаване на конкретни ЕС чрез използване на ПОПЕС е допълването на БЗ. Метод за облекчаване на тази дейност е автоматизиране, доколкото е възможно, на формализирането на знанията. Формализацията се извършва с активното участие на потребителя (специалист в проблемната област) и се състои от два етапа. При първия знанията, въведени на език, близък до естествения, се "превеждат" на вътрешен език, а при втория етап се преминава от вътрешния език към правила.

ПОДСИСТЕМАТА ЗА ФОРМАЛИЗАЦИЯ И НАТРУПВАНЕ НА ЗНАНИЯТА е

предназначена за работа със силно ограничен естествен език. Реализацията на тази система е сравнително лесна поради тясната предметна област. Първо се отделят компонентите на условието и следствието. Във втория етап се формализират знанията, т. е. генерира се правило или правила, отразяващи знанията, включват се така създадените правила в БЗ, изследват се за връзки с други правила, за противоречивост, за цикличност на знанията и т. н. Ако условието на правилото съдържа данни, за които ще се задава въпрос на потребителя, създава се набор от евентуални отговори. Характерно в случая е, че се предвижда непрекъснат контрол върху процеса на формализация от страна на специалиста-експерт, който въвежда знанията. В съответствие с тези два етапа, системата за формализация и натрупване на знанията се състои от две части – лингвистичен процесор и подсистема за натрупване на знанията.

ЕКСПЕРТНАТА СИСТЕМА ЗА АТЕСТИРАНЕ НА РЪКОВОДНИ КАДРИ (ЕСА) представлява конкретизация на ЕСОК и може да извършва три основни вида дейности:

1) Атестиране на личности или колективи на базата на автоматично определени набори критерии от БЗ.

2) Определяне на набори критерии за атестиране на базата на знания за предметната област, за длъжностната характеристика и т. н. Това позволява за всяка отделна длъжност да се създаде автоматично конкретна, характерна само за нея, последователност от критерии за атестиране, отчитаща максимално спецификата на длъжността.

3) Консултации по въпроси в областта на атестирането.

Самата област на приложение на системата – оценка на

ръководни кадри и в частност атестиране предполага приложението на ЕСА на различни места и на различни нива. Освен това, системата е носител на експертни знания, т. е. на знанията на най-изявените специалисти в областта на атестирането. Съдържайки в себе си знания от най-високо ниво, ЕСА се превръща не само в помощник при атестирането, но и в справочник и учител в тази област, като имаме предвид че атестирането не е професия или основна дейност за повечето от хората, извършващи тази дейност. Имайки предвид подсистемата за извличане на знания може да се каже, че ЕСА е мощно средство за изследвания в областта на атестирането.

При проектирането на ЕСОК и ЕСА авторите получиха съществени консултации за предметната област от доц. к. ф. н. П. Данев – инструктор в отдел "Организационен" в ЦК на БКП и от други сътрудници на този отдел.

В БЗ на ЕСОК се въвеждат знания (правила), свързани с атестирането на ръководни кадри. Това са преди всичко критерии, отразяващи изискванията на заеманата длъжност, конкретните условия, принципите на атестирането и др. В проблемната част на БЗ са заложили във вид на правила и постановките за единство на политическите и делови качества, теоретичните постановки от областта на атестирането, критерии за атестиране, правила за избор на критерии и т. н.

ГЛАВА V. ИНТЕЛЕКТУАЛИЗИРАНИ ВЪПРОСНО-ОТВЕТНИ СИСТЕМИ (ИВОС)

1. ПРЕДМЕТНО-ОРИЕНТИРАНА ПРАЗНА ИВОС ЗА АНАЛИЗ НА ИСТОРИКО-ГЕОГРАФСКИ ТЕКСТОВЕ

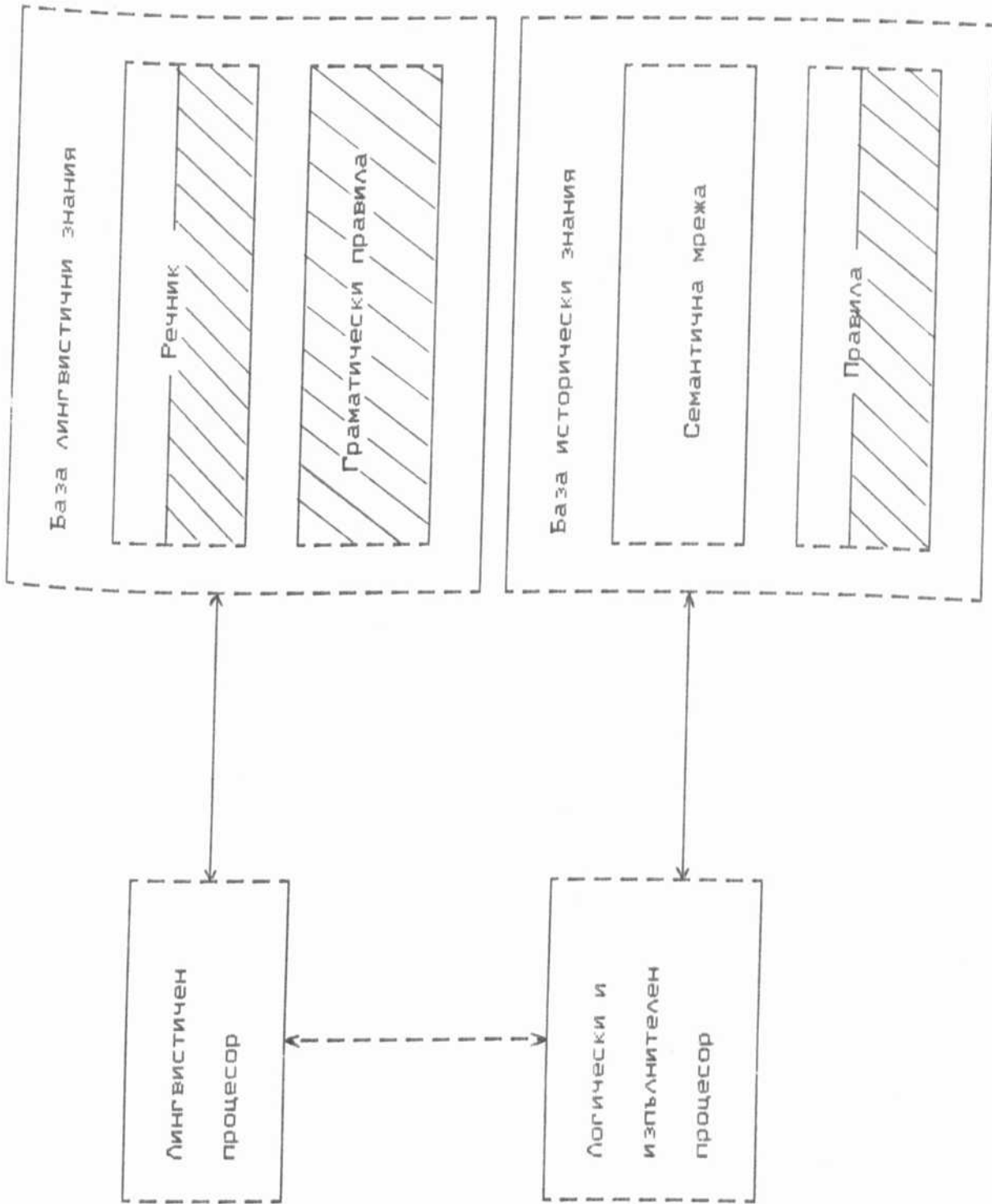
Системата е проектирана като интелигентен помощник на историците в техните изследвания по историческа география на Тракия и служи за проверка на различни хипотези и твърдения на древни и съвременни автори [Томов, Геров, Григоров, Колчев 88].

Интелектуализираната въпросно-ответна система ЕСИТ за анализ на историко-географски текстове за древна Тракия се разработва, като се използва горепосоченият (гл. I, т. 3) подход за създаване на предметно-ориентирани празни ИППС. Разработва се предметно-ориентирана празна ИВОС за анализ на историко-географски текстове и след това тя се конкретизира за логически и географски анализ на историко-географски текстове за древна Тракия и се получава проблемно-ориентирана ИВОС.

Общата схема на предметно-ориентираната празна ИВОС е дадена на фиг. 12.

Системата се състои от следните основни модули: лингвистичен процесор, база лингвистични знания, изпълнителен процесор и база исторически знания.

Лингвистичният процесор осигурява интерфейса с



Фиг. 12. Обща схема на предметно-ориентирана празна ИВРС.

потребителя на ограничен естествен език [Григоров 87]. Ограниченият естествен език е подмножество на българския език и е ориентиран към предметната област, т. е. в него се използват термини и изрази, характерни за историческата география. Лингвистичният процесор превежда изречения на български във фрази на вътрешния език на системата и обратно. За целта той използва базата лингвистични знания, която се състои от речник и граматически правила.

В речника се съдържа информация за думите и техните морфологични категории (род, число, определител и т. н.). В празната ИВОС в речника е записан минимален запас от характерни за предметната област думи.

Граматическите правила служат за описание на структурата на изреченията на ограничения естествен език и се използват за извършване на синтактичен анализ.

Основните функции на системата се изпълняват от логическия и изпълнителен процесор. Той поддържа базата исторически знания, добавя нови факти в нея, търси съпоставими семантични мрежи и извършва логически изводи.

За представяне на историческите знания се използват семантични мрежи и правила за извод [Григоров 87a]. Семантичната мрежа служи за представяне на информацията от историко-географските текстове. В същност това са данни за системата и затова в предметно-ориентираната празна ИВОС семантичната мрежа не е попълнена. Правилата в базата исторически знания служат за извършване на изводи над семантичната мрежа. Чрез тях могат да се извежат факти, които не са дадени в явен вид в семантичната мрежа. Базата исторически знания на празната ИВОС съдържа правила, които

са обобщовалидни за цялата предметна област.

На фиг. 12 частите от базите знания, които са попълнени в предметно-ориентираната празна ИВОС, са застриховани.

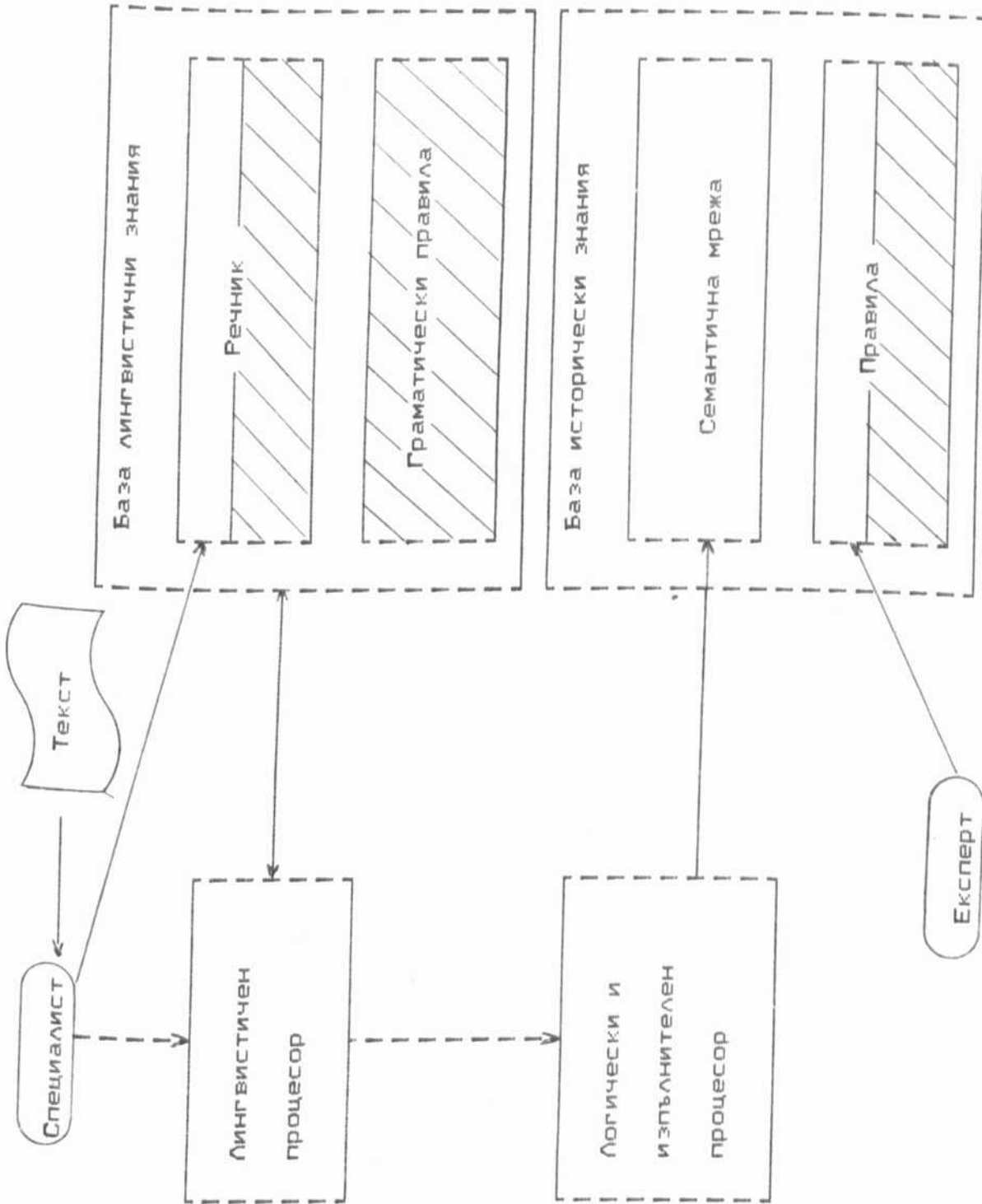
2. ПРОБЛЕМНО-ОРИЕНТИРАНА ИВОС ЗА ГЕОГРАФСКИ И ЛОГИЧЕСКИ АНАЛИЗ НА ИСТОРИКО-ГЕОГРАФСКИ ТЕКСТОВЕ ЗА ДРЕВНА ТРАКИЯ

На фиг. 13 е показан процесът на конкретизация на предметно-ориентираната празна ИВОС. Това става чрез попълване на проблемно-ориентираните знания и на данните за системата.

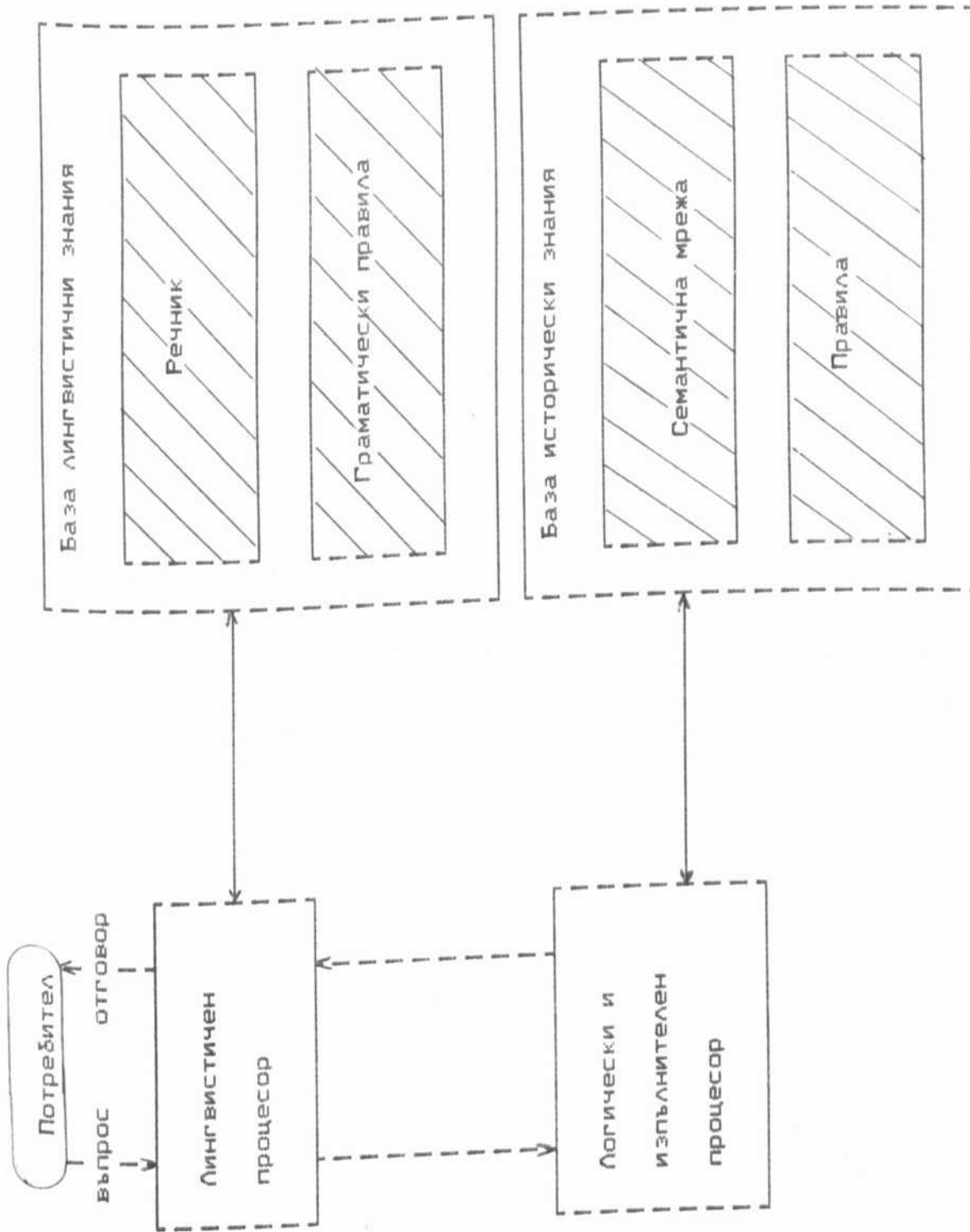
Проблемно-ориентираните правила в базата исторически знания се попълват от експерт. Това са правила, чрез които може да се потвърди или отхвърли дадена хипотеза или твърдение относно географското разположение на даден исторически обект. С тях могат да се търсят и противоречия между становищата на различни автори.

Въвеждането на историко-географската информация от текстовете на древните автори е отчасти автоматизирано. Информацията от текстовете се въвежда от специалист във вид на изречения на ограничен естествен език. Освен това, специалистът добавя в речника и новите думи от текстовете. Лингвистичният процесор превежда изреченията във фрази на вътрешния език на системата, които представляват линеен запис на съответните семантични мрежи. Накрая логическият и изпълнителен процесор добавя семантичната мрежа към базата исторически знания.

По този начин се получава проблемно-ориентирана ИВОС за



Фиг. 13. Конкретизация на предметно-ориентираната празна ИВОС.



Фиг. 14. Проблемно-ориентирана ИВРС за логически и географски анализ на историко-географски текстове за древна Тракия.

логически и географски анализ на историко-географски текстове за древна Тракия. Нейното функциониране е дадено на фиг. 14.

Потребителят задава заявка към системата във вид на въпрос на ограничен естествен език. Той се превежда от лингвистичния процесор в заявка към логическия и изпълнителен процесор. Логическият и изпълнителен процесор изпълнява заявката, като търси в базата исторически знания съпоставима семантична мрежа и извършва логически изводи. В резултат той връща фраза на вътрешния език, която се превежда от лингвистичния процесор в изречение. Това изречение представлява отговора на системата.

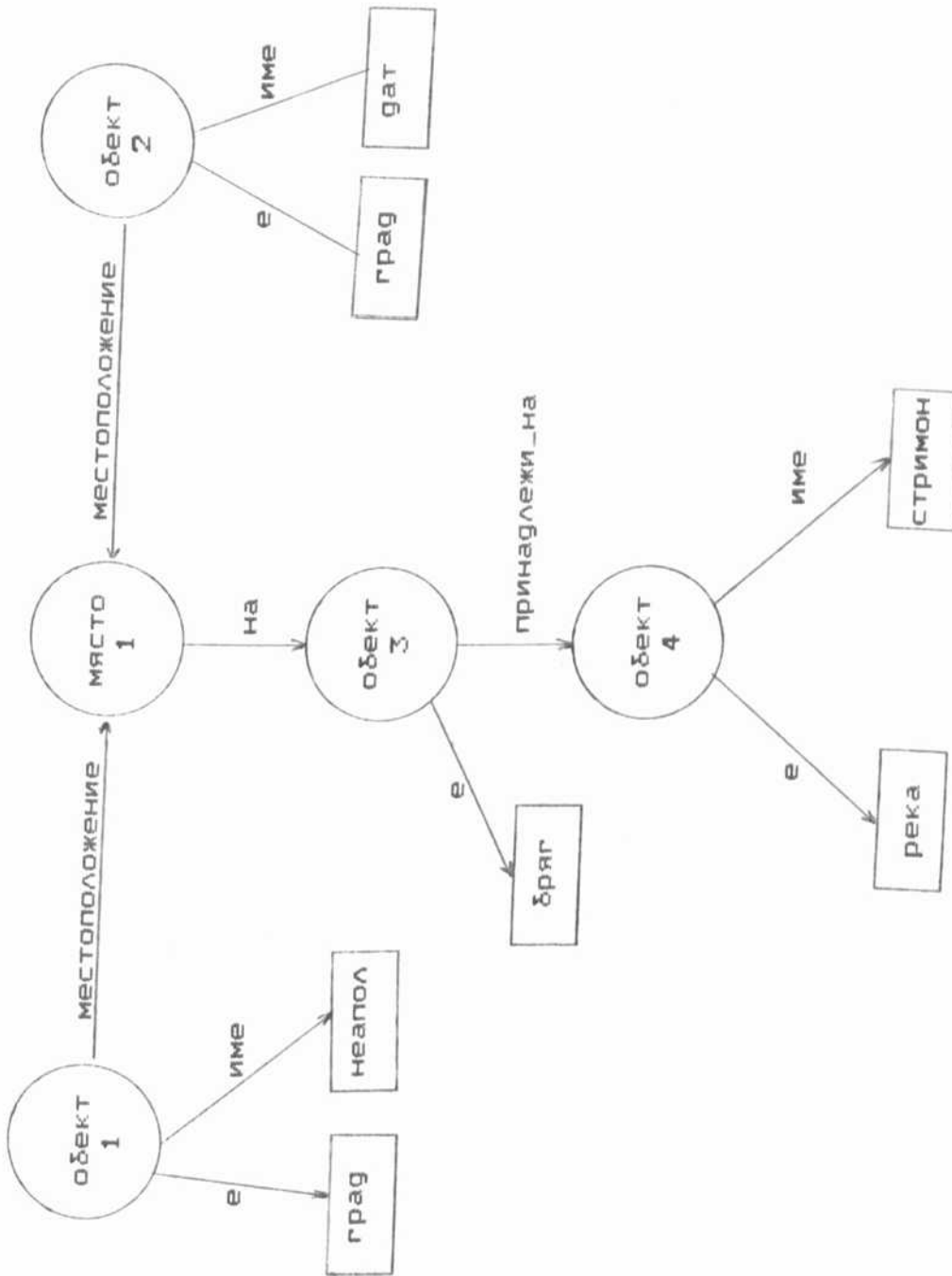
Описаната система може да служи като прототип за създаване на интелектуализирани въпросно-ответни системи. След сравнително малко изменения тя може да се използва и за други предметни области.

3. ПРЕДСТАВЯНЕ НА ИСТОРИЧЕСКИТЕ ЗНАНИЯ В СИСТЕМАТА ЕСИТ

Както споменахме по-горе, базата исторически знания се състои от семантична мрежа и правила за извод. Семантичната мрежа се използва за представяне на историко-географските знания от текстовете на древните автори. Тя се състои от именувани възли и насочени дъги. Възлите представят обекти, понятия, събития и действия, а дъгите задават релациите между тях.

Ще дадем няколко примера. Нека разгледаме следното изречение:

(1) "Град Неапол и град Дат са на брега на река Стримон."



Фиг. 15. Семантична мрежа, съответстваваща на изречение (1).

Съответстващата на изречението семантична мрежа е дадена на фиг. 15.

Обектите в семантичната мрежа са нарисувани с кръгчета, а понятията с правоъгълници. Обектите в системата са номерирани, като всеки обект има единствен номер. Така например, "обект 1" означава град Неапол, а "обект 4" – река Стримон.

"е"-връзката задава типа на обектите – типът на "обект 1" е "град", докато типът на "обект 3" е "бряг".

"име"-връзката задава името на обекта.

Местоположението на обектите се задава с връзката "местоположение". Така например местоположението на "обект 2" е "място 1", като "място 1" е на "обект 3", зададено чрез връзката "на".

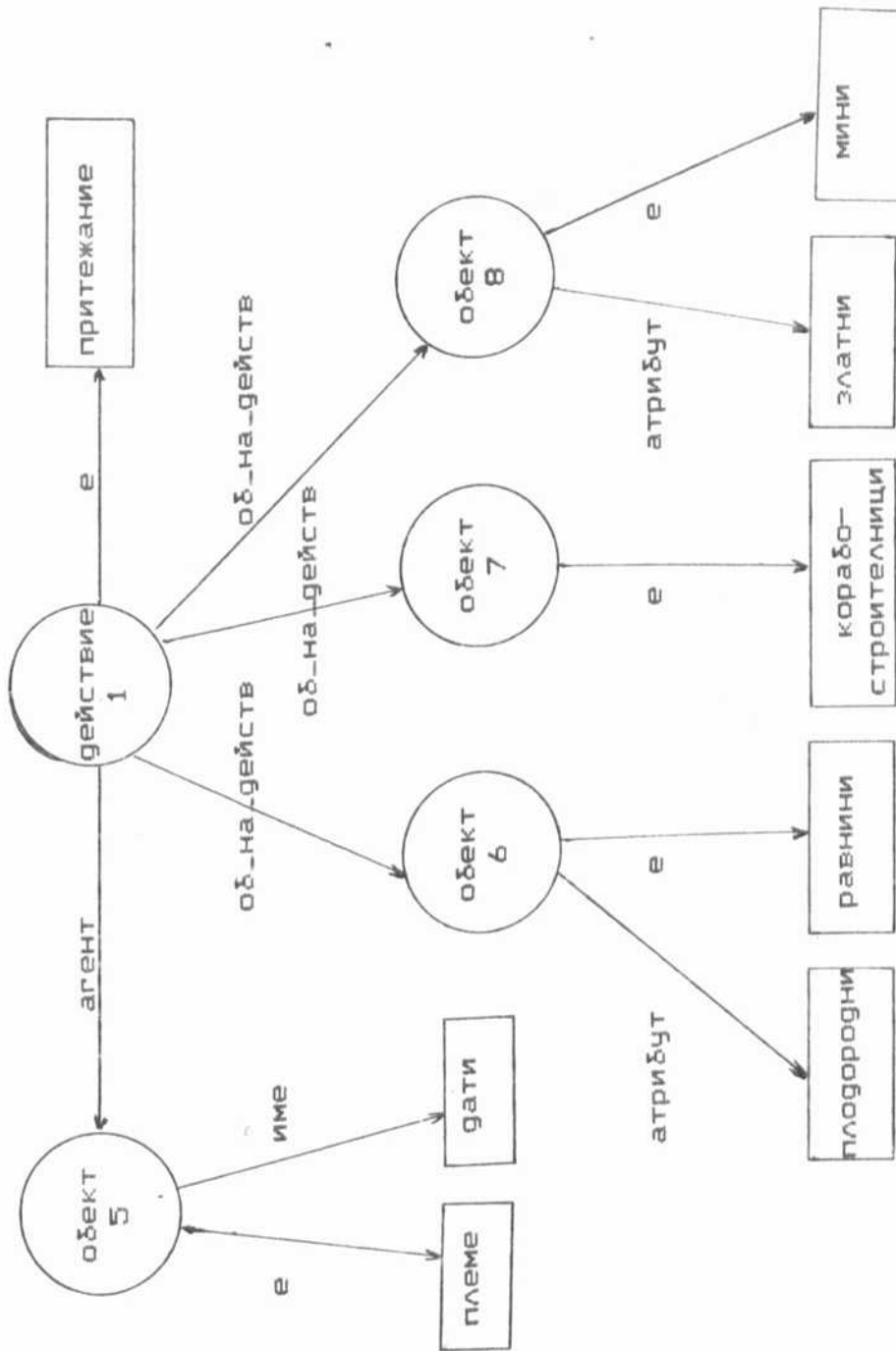
"принадлежи_на"-връзката означава, че даден обект принадлежи или е част от друг обект.

По-сложните отношения между обектите се представят в семантичната мрежа като се разбиват на бинарни релации. Такива са например действията, събитията, ситуациите и т. н., които условно ще наричаме "действия". На фиг. 16 е дадена семантичната мрежа, съответстваща на следното изречение:

(2) "Племето гати притежава плодородни равнини, корабостроителници и златни мини."

Тук "действието" е притежание. То се представя чрез възел "действие 1", който е от тип "притежание". Притежателят се задава чрез връзката "агент", а притежаваните обекти чрез връзките "обект_на_действието".

Въпросите към системата също се представят във вид на



Фиг. 16. Семантична мрежа, съответстваща на изречение (2).

семантична мрежа. В този случай някои от възлите на мрежата са променливи. Например въпросът:

(3) "Къде е град Дат?"

се представя чрез семантичната подмрежа, показана на фиг.

17. Тя има следния смисъл:

"Съществува ли обект с номер X, който е град с име Дат и местоположението му е място Y? "

Семантичните мрежи, съответстващи на въпросите към системата, служат като образец за търсене в базата знания.

Сега ще се спрем на представянето на семантичните мрежи чрез езика ПРОЛОГ. Както е показано в [Ковалски 79] и [Делияни, Ковалски 81], семантичните мрежи могат лесно да се изразят чрез средствата на логиката, а следователно и чрез ПРОЛОГ. Семантичната мрежа се представя като съвкупност от факти на ПРОЛОГ, като всеки факт обозначава отделна дъга. Функционаторът на факта задава името на връзката, а аргументите му - възлите, свързани с тази дъга. Например, семантичната мрежа от фиг. 15 се представя чрез следните факти на ПРОЛОГ:

е(обект(1), град).

е(обект(2), град).

е(обект(3), дряг).

е(обект(4), река).

име(обект(1), неапол).

име(обект(2), дат).

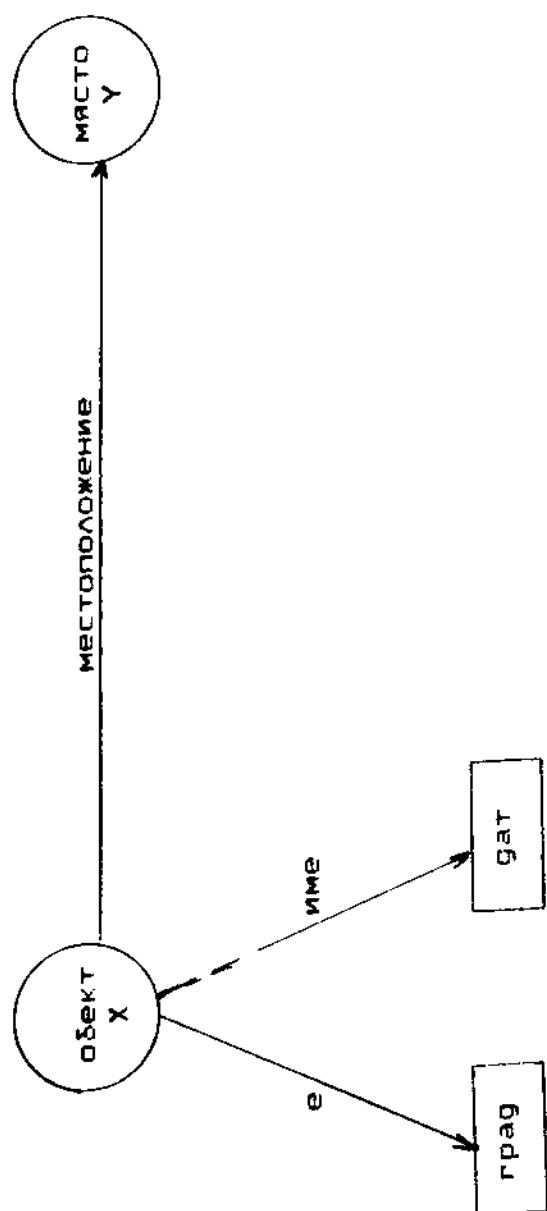
име(обект(4), стримон).

местоположение(обект(1), място(1)).

местоположение(обект(2), място(1)).

на(място(1), обект(3)).

принадлежи_на(обект(3), обект(4)).



Фиг. 17. Семантична мрежа, съответстваща на въпрос (3).

Правилата за извод са също част от базата исторически знания. Те изразяват мета-знание за предметната област. Правилата за извод се представят като клаузи на Хорн от следния вид:

(4) $R_0 \leftarrow R_1 \ \& \ R_2 \ \& \dots \ \& \ R_N.$

Те имат следния смисъл:

" R_0 е вярно, ако R_1 и R_2 и... и R_N са верни."

Така например правилото:

$\text{на_север_от}(X, Y) \leftarrow \text{на_юг_от}(Y, X).$

означава, че X е на север от Y , ако Y е на юг от X .

А следното правило:

$\text{на_север_от}(X, Y) \leftarrow \text{на_север_от}(X, Z) \ \& \ \text{на_север_от}(Z, Y).$

изразява факта, че "на_север_от" е транзитивна релация.

Правилата за извод имат също и процедурна интерпретация. Правило (4) се интерпретира по следния начин:

"За да докажеш, че R_0 е вярно, докажи, че R_1, R_2, \dots, R_N са верни. "

4. ДЕЙСТВИЕ НА ЛОГИЧЕСКИЯ И ИЗПЪЛНИТЕЛЕН ПРОЦЕСОР НА СИСТЕМАТА ЕСИТ

Логическият и изпълнителен процесор изпълнява няколко основни функции:

- поддръжка на базата исторически знания;
- идентифициране на обекти;
- търсене по образец на семантични подмрежи;
- извършване на логически заключения.

Този процесор комуникира с лингвистичния процесор чрез

фрази на вътрешния език. Фразата на вътрешния език представлява линеаризация на семантичната подмрежа, съответстваща на заявката или отговора. Да предположим, че искаме да добавим към базата знания изречението (1). Фразата, съответстваща на това изречение е:

```

местоположение(
    (обект(X) \ е(обект(X), град) &
      име(обект(X), неапол)) &
    (обект(Y) \ е(обект(Y), град) &
      име(обект(Y), гат) ),
    място(Z) \ на(място(Z),
      обект(U) \ е(обект(U), дряг) &
      принадлежи_на(обект(U),
обект(V) \ е(обект(V), река) &
      име(обект(V), стримон)
      )
    )
  ).

```

Конструкцията

```

обект(X) \ е(обект(X), град) &
  име(обект(X), неапол)

```

обозначава обекта с номер X, където обект X е град и има име Неапол. Номерата на обектите в заявката са променливи. Когато изпълнителният процесор получи заявка, той се опитва да идентифицира обектите в нея. Ако успее да намери даден обект, в базата знания се добавят само новите отношения, в които обекта участва. Ако изпълнителният процесор не може да идентифицира обекта, той създава нов обект. В дадения случай, ако базата знания е празна, изпълнителният процесор

ще създаде семантичната мрежа, дадена на фиг. 15.

Когато е зададен въпрос към системата, логическият и изпълнителен процесор търси в базата знания съпоставима семантична подмрежа. Да предположим, че базата знания се състои от семантичната мрежа на фиг. 15 и искаме да зададем въпроса:

"Къде е град Дат? "

Фразата на вътрешния език е следната:

местоположение(обект(X) \ е(обект(X), град) &
име(обект(X), дат), място(Y)).

и тя съответствува на семантичната подмрежа от фиг. 17. Процесорът намира съпоставима семантична подмрежа, като при това променливите X и Y получават стойности съответно 2 и 1. По нататък, процесорът извлича от базата знания информацията, определяща "място 1" и връща като резултат следната фраза:

местоположение(
обект(2) \ е(обект(2), град) &
име(обект(2), дат),
място(1) \ на(място(1),
обект(3) \ е(обект(3), бряг) &
принадлежи_на(обект(3),
обект(4) \
име(обект(4), стримон))).

Тя съответствува на отговора:

"Град Дат е на брега на Стримон. "

В процеса на търсене на съпоставима семантична подмрежа логическият и изпълнителен процесор използва правилата в базата знания за да провери дали даден факт, който не е зададен явно в семантичната мрежа, може логически да се

изведе от останалите факти в базата знания. Реализираната стратегия при извършване на логическия извод е отгоре надолу и в дълбочина. В процеса на логическия извод изпълнителният процесор отбелязва установените факти, за да може по-късно системата да поясни отговора си.

Интелектуализираната въпросно-ответна система за анализ на историко-географски текстове е разработена от колектив от математици от сектор "Изкуствен интелект" при ИМ с ИЦ - БАН и от специалисти по история на Тракия от ИФ на СУ "Кл. Охридски". Системата е реализирана на езика ПРОЛОГ за микрокомпютри, съвместими с IBM-PC /XT.

ГЛАВА VI. ИНТЕЛЕКТУАЛИЗИРАНИ АНАЛИТИЧНО-ИЗЧИСЛИТЕЛНИ СИСТЕМИ

1. СИСТЕМИ ЗА АНАЛИТИЧНИ ПРЕОБРАЗОВАНИЯ (САП)

Развитието на информатиката и нейните приложения през последните години все по-често дават основание да се говори за действително широко навлизане на ЕИМ в различни области на живота. И макар, че традиционните приложения на ЕИМ – за числени пресмятания – не губят своята актуалност, важно значение имат редица нови възможности, характерни за последните едно-две десетилетия. Такава е, например, възможността с помощта на ЕИМ да се извършват аналитични преобразования, т. е. формални преобразования на математически изрази, зададени в буквено-цифров вид, по определени правила. Това означава манипулиране с формули, опростяване на математически изрази, като например привеждане на подобните членове в многочлен, аналитично диференциране и интегриране, намиране точното решение на дадено диференциално уравнение и др.

Например:

$$(a + b)^2 - (a - b)^2 = 4ab$$

$$(x + 3)^{20} = x^{20} + 60x^{19} + 1710x^{18} + \dots + 3486784401$$

$$\frac{\log(a^{2x} + 2a^x + 1)}{\log(a^x + 1)} = 2$$

$$\frac{d}{dx} \sin(x^2) = 2x \cos(x^2)$$

$$\int (1-y)^2 dy = \frac{(y-1)^3}{3}$$

$$\det \begin{vmatrix} a+1 & a \\ 5 & a-1 \end{vmatrix} = a^2 - 5a - 1$$

Какви възможности дава автоматичното, т.е. с помощта на ЕИМ, извършване на аналитични преобразования? Преди всичко, получаване на надежден резултат в задачи, свързани със сложни преобразования на дълги изрази. Още по-важна е възможността за намиране на точното решение (т.е. решение в аналитичен вид) на много задачи, които с помощта на ЕИМ до неотдавна се решаваха само числено. Аналитичният вид на решението позволява качествен анализ на изследваните с помощта на съответната (математическа) задача процеси и явления [Томов, Геров, Спиридонова 85].

Първите успешни опити с програми за аналитично диференциране [Кариманиан 53], [Нолан 53] датират от 1953 г. През 1958-61 г. се създават първите езици за работа със символна информация, представена във вид на списъци. Особено удобен се оказва създаденият през 1961 г. език LISP [Маккарти 62], който и днес намира широко приложение.

През 1961 г. се появява програмата SAINT [Слейджл 67], предназначена за аналитично интегриране. Нейните възможности съответствуват на нивото на студент от първи курс на университета.

Първите системи за аналитични преобразования (САП) с общо предназначение са създадени през 1964 г. Това са системите FORMAC [Бонг и др. 64] и ALPAK [Браун 64].

Наред със системите с общо предназначение се създават и специализирани системи, ориентирани към определена област, например неъесната механика, квантовата теория на полето, теоретичната астрономия и др.

Общият ърой на създадените до днес САП (с общо предназначение и специализирани) е около 60. Сведения за тези системи и за тяхното приложение могат да се намерят в статиите [Гергт, Тарасов, Ширков 80], [Геров, Капитонова, Спиридонова, Томов 84], [Ван Хулцен, Калме 82], [Калме, Ван Хулцен 82].

САП обикновено включват езици за удобно описание на аналитични преобразования и съответни транслятори.

От гледна точка на потребителите САП се характеризират с допустими изрази (полиноми, ъроъно-рационални функции, трансцендентни функции, степенни редове и др.) и възможните преобразования на тези изрази (действия с полиноми и ъроъно-рационални функции, диференциране, интегриране, линейни трансформации – на Лаплас, Фурие и др., линейна алгебра и т.н.), а също така и с удобството на езика, възможност за работа в диалогов режим, използване на бази от математически знания и т.н.

На "вътрешно" ниво САП се характеризират с избрания

език за реализация, организацията на паметта, реализирането (обикновено) на рационална аритметика с неограничена точност и др.

САП намират широко приложение в научно-изследователската и приложна дейност в областта на математиката, физиката, механиката, геодезията, техниката и др.

1.1. Основни сведения за някои САП

Ще разгледаме три от най-известните САП – REDUCE, FORMAC и SAC-2. Трите системи са с общо предназначение и са усвоени в ИМ с ИЦ при БАН със съдействието на ЛИИЦ при СУ "Кл. Охридски".

Възможностите на REDUCE [Хърн 85] включват:

- действия с цяла и рационална аритметика с произволна точност; развитие и привеждане на полиноми; действия с полиноми и рационални функции; субституции в изрази; автоматично и контролирано от потребителя опростяване на изрази; аналитично диференциране и интегриране; аналитични пресмятания с матрици;
- специални преобразования, свързани със задачи от физиката на високите енергии;
- режим, позволяващ използването на езика за обработка на символна информация LISP, на който е написана системата;
- възможност за извеждане на преобразуваните от системата изрази във вид на програма на FORTRAN.

Езикът на REDUCE е погоден на езика ALGOL. Той е лесен за изучаване и удобен за записване на необходимите

аналитични преобразования. Включва оператори за присвояване, условен и безусловен преход, възврат, цикъл, субституции, диференциране, работа с външни файлове, съставни оператори и др.

Системата FORMAC [Бонг и др. 64] е разработена от фирмата IBM. Първата версия FORTRAN - FORMAC е завършена през 1964 г. По-късно се появява и втора версия - PL/I - FORMAC, която през 1973 г. е подобрена от Баар, ФРГ [Ван Хулцен, Калме 82]. Основната концепция при разработване на системата е добавянето на средства за аналитични преобразования към вече съществуващ език за числени пресмятания. Системата дава възможност за работа с алгебрични изрази, съдържащи и трансцендентни функции. Има възможност за аналитично извършване на основните действия с изрази, опростяване, диференциране, сравняване и анализиране на изрази. Има и много други средства за връзка с числени процедури. От 1981 г. в ИМ с ИЦ при БАН е внедрена и се използва версия на FORMAC с базисен език PL/I.

SAC-2 [Колинз, Лоос 80] е система с общо предназначение, създадена от Дж. Колинс и Р. Лоос през 1980 г. Тя включва езика за обработка на символна информация ALDES, компилатор за него и набор от подпрограми, реализиращи основни алгебрични преобразования. Езикът на системата може да се използва и самостоятелно като език за обработка на символна информация. ALDES използва следните типове данни: цели числа, символи (аналогично на атомите в LISP), списъци и низове, и разполага с всички основни структурни оператори. Наборът от вградени функции покрива съществена част от функциите на LISP. При определяне на

подпрограми се разрешава използването на рекурсия. Компиляторът е написан на FORTRAN и извършва превод от ALDES на FORTRAN, което осигурява получаването на бързи и машинно-независими програмни продукти.

Подпрограмите за основните алгебрични преобразования могат да се разделят на три групи:

- аритметични. В тази група са включени около 120 програми, реализиращи аритметичните действия с цели и рационални числа с неограничена точност;
- за работа с полиноми. Групата включва около 200 програми, обработващи полиноми на една или повече променливи с коефициенти цели и рационални числа. Възможни са основните аритметични операции с полиноми, а също така пресмятане стойност на полином, диференциране, интерполиране, намиране на най-голям общ делител, разлагане на множители и др.
- за линейна алгебра. Групата включва 22 програми, осигуряващи основните операции с вектори и матрици, чиито елементи могат да бъдат цели числа или полиноми. Съществуват възможности за решаване на диофантова система от първа степен и намиране на базисното решение по модул на система от първа степен.

1.2. Специализирани САП

Най-популярните САП обикновено дават възможност за приложение в различни области, т.е. те са системи с общо предназначение. Но обикновено за решаване на конкретна задача се налага, използвайки готовите средства на системата, да се създават и нови средства, ориентирани към

областта, от която е задачата. Това изисква съществени умения в областта на програмирането и налага потребителят да бъде специалист освен в областта, за която се прилага САП, още и в областта на информатиката. Естествено такова изискване ограничава масовото използване на САП. Начин на решаването на този проблем е създаването на САП с развити възможности за редица области. Но това означава създаването на прекалено големи системи, които ще изискват както голям ресурс от програмисти при тяхното създаване, така и мощни ЕИМ при тяхното използване. В този случай затрудненията за потребителя ще са по-малко, но въпреки това той ще трябва да изучи използването на една система с много възможности, част от които при решаването на неговите конкретни задачи ще му са ненужни. Съществен проблем е и необходимостта от големи изчислителни ресурси и оттам ниската ефективност при използването на такива универсални САП. От значение е и това, че сравнително ограничената предметна област на специализираната САП обуславя повишената ефективност по време и памет на системата.

Едно добро решение на проблема е създаването на специализирани САП за определени области на приложение. За основа на такива специализирани САП е подходящо да се използват вече създадени САП с общо предназначение и за дадена конкретна област да се развият специализирани средства. Колективът, който ще извършва тази дейност, трябва да е съставен както от специалисти по информатика, така и от специалисти в съответната област, за която е предназначена специализираната САП. Областта, за която се създава такава система, и степента на нейното развитие се определят чрез

подходящ компромис така, че системата да бъде от една страна достатъчно ефективна и от друга, да има достатъчно широка област на приложение. По този начин ще се избегне необходимостта потребителят да бъде специалист по програмиране. Освен това потребителят ще трябва да изучава използването на една система, която включва само нужни за него действия, т.е. минимум от нови знания, които да са във неговата специалност. Също така съществено ще се намали необходимият изчислителен ресурс за използването на такъв вид системи.

При създаването на специализирани САП е необходимо да се има предвид и възможността за евентуално преминаване от използване на дадена специализирана система за определен клас задачи към друга САП за друг клас задачи. В този случай е добре знанията, необходими за работа с първата система, да спомагат за усвояване на втората система. Това може да се постигне чрез стандартизиране на средствата за работа с различни специализирани САП.

В ИМ с ИЦ при БАН най-голям опит за работа е придобит със системите REDUCE и SAC-2. Всяка от тези системи, разглеждана като съвкупност от език и транслатор, е използвана за разработка на редица специализирани САП. Така например, на основата на REDUCE са разработени: система за аналитично решаване на някои типове задачи от механиката [Спиридонова, Хаджиков, Динева 83], [Спиридонова, Ищев, Филипков 83], система за операционно смятане, осъществяваща символно преобразование на Лаплас [Спиридонова, Казасов 83] и др. [Спиридонова, Джамбазова 84].

Системата за операционно смятане е предназначена за

аналитично извършване на право и обратно преобразование на Лаплас за определен набор функции, позволяващ приложение на операторния метод при решаване на практически задачи.

Друга група разработки са създадени на основа на системата за аналитични преобразования SAC-2: система за пресмятане на неопределени интеграли от дробно-рационални функции, система за работа с матрици [Гергов 87], [Гергов, Григоров 83], [Гергов, Дечева 83], система за работа с непрекъснати дробни [Нишева, Тонев 84] и др. [Гергов, Куманов 84]. В момента се завършва и система за факторизация на полиноми.

Специализираната аналитично-изчислителна система за работа с матрици има за цел да осигури аналитични преобразования на обекти, необходими в линейната алгебра (вектори и матрици), а също и да даде възможност за използване на изчислителни процедури, написани на FORTRAN и PL/1, предназначени за работа с вектори и матрици. Създадени са функции, чрез които по подходящ начин се въвеждат и извеждат вектори и матрици, чиито елементи могат да бъдат цели числа, рационални числа, полиноми и дробно-рационални функции за осъществяване на някои традиционни действия с матрици: транспониране на матрица, намиране на детерминанта, поелементно събиране и изваждане, умножение на матрици и др. Създадени са програми за аналитично решение на матричното уравнение $A \cdot Y = B$, където A е ненулева матрица с размер $M \times N$, а B - матрица с размер $M \times K$. Друга група програми дават възможност за намиране на характеристичен полином $(\lambda) = \det(A - E \lambda)$, собствени стойности $(\lambda) = 0$ и собствени вектори $(A - E) \cdot x = 0$

на матрицата A .

При реализирането на специализирани САП особено внимание се обръща на създаването на удобен език за потребителя, имащ общи принципи за всички системи.

2. РАЗВИТИЕ НА САП

2.1. Изисквания към съвременните САП

Първите програми за аналитични преобразования са си поставяли скромната цел да докажат възможността ЕИМ да извършва дейности, считани за интелектуални. В последствие, резултатите в това направление започват да се прилагат при решаването на редица научни и инженерни проблеми. Така се стига до поставянето на сложни задачи, изискващи силно развити САП. Съвременните математически задачи, възникващи в различни области на науката и техниката, много често изискват разнообразни методи за тяхното решаване с помощта на ЕИМ. За да може една САП да се справи с такава задача, е необходимо [Геров, Григоров 83] тя да има средства както за извършване на аналитични преобразования с помощта на точни или евристични алгоритми, така и за извършване на числени пресмятания. По този начин е възможно решаването на задачи, за които са необходими както извършването на трудоемки аналитични преобразования, така и изпълнението на голям обем числени пресмятания. Евентуалното използване на различни системи за двата вида работа при съществуващите различия за представянето на данните и на начина на използването им би затруднило дотолкова възможните

потребители, че те много често биха се отказали от използването на ЕИМ за голяма част от работата си. Особено ясно се вижда необходимостта от единна система, включваща и двата вида изчисления при задачи, изискващи неколккратно редуване на аналитични преобразования с числени пресмятания.

Друго съществено изискване към САП е възможността за работа с разнообразни математически обекти. Тези обекти могат да бъдат числа, полиноми, или по-общо функции, вектори, матрици, непрекъснати гроби, редове, интеграли, множества и т.н. От друга страна тези обекти могат да бъдат дефинирани над различни полета, например цели, рационални, ирационални, комплексни, реални в смисъл на езика FORTRAN и т.н. Това изискване поставя сериозни задачи както в областта на създаване на различни алгоритми за работа с такива обекти, така и в областта на представянето на това разнообразно множество от обекти.

Не на последно място са изискванията за удобство на потребителя при ползването на такива САП. На първо място е необходима възможност за задаване на задачата в диалогов режим, а при задачи, изискващи малко процесорно време – изцяло диалогова работа. Друго съществено изискване е потребителят да не е длъжен да се съобразява с вътрешното представяне на различните обекти (дори да не е необходимо да го познава) и да не е задължен да знае различните алгоритмични и програми, реализиращи действията с различните по тип обекти. Необходимите знания трябва да се свеждат до познаване на възможностите на системата (допустими действия) и тяхното използване от потребителя да е независимо от конкретните математически обекти и области над които са

дефинирани.

2.2. Интелектуализирани аналитично-изчислителни системи

От поставените по-горе изисквания следва, че тези системи трябва равноправно да съдържат както средства за аналитични преобразования, така и средства за числени пресмятания и в такъв случай едно подходящо наименование за тях е аналитично-изчислителни системи (АИС).

Освен това, редица от изискванията поставят проблеми, които могат да бъдат решавани със средствата на изкуствения интелект [Геров, Капитанова, Спиридонова, Томов 84], [Томов, Спиридонова, Геров 84] (това са, преди всичко, удобната връзка с потребителя и автоматизирането на планирането и избора на метод за решаване на поставената от потребителя задача). Решаването на тези проблеми води до създаване на интелектуализирани аналитично-изчислителни системи.

Програмите и програмните системи за аналитични преобразования от началото на своето създаване използват преди всичко методите за обработка на символна информация и изкуствения интелект. Ако пак цитираме програмата за аналитично интегриране SAINT, дихме отбелязали, че през 1961 г. тя се е разглеждала като постижение в областта на изкуствения интелект – отбелязвало се е, че нейните "интелектуални" способности в областта на аналитичното интегриране са съответствували на тези на студент от I курс.

Една от най-мощните съвременни системи за аналитични преобразования MACSYMA често се цитира като експертна

система за аналитични преобразования [Уинстон 84]. Тя съдържа стотици правила, формулирани от експерти (главно математици и физици). Всяко от тези правила задава преобразуването на даден израз по определен начин. За решаването на конкретна задача, свързана с аналитично преобразуване на даден израз е необходимо построяването на верига от правила, с помощта на които ще бъде извършено необходимото преобразуване.

Друг пример от последните години е написаната на езика PROLOG и основана на правила система PRESS (PROlog Equation Solving System) [Геров, Капитонова, Спиридонова, Томов 84].

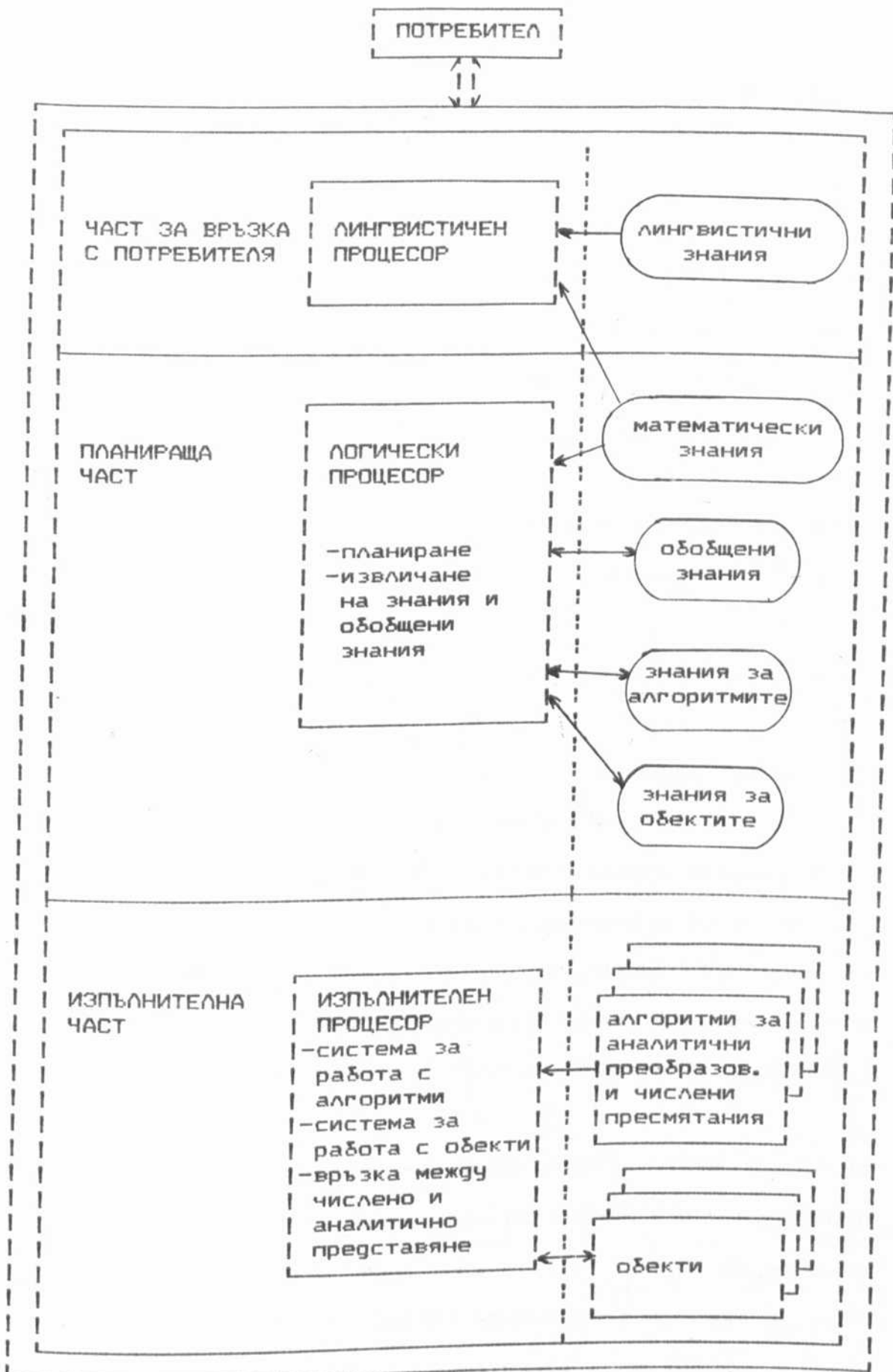
Споменатите и някои други подобни примери касаят основните аспекти на връзка между САП и ИИ:

- алгоритмите за аналитични преобразования (с използване на евристики, аналогии и т.н.)

- интелектуализацията на САП - чрез включване в САП на средства за анализ на математическите задачи, планиране на действията, използване на бази математически знания, развитие на езиците на системите с цел приближаване до естествения език и т. н.

Приложението на методите на ИИ в САП увеличава техните възможности и повишава интелигентността им. При това следва да се отбележи, че строго дефинираните математически обекти и операции в САП са добра предпоставка за успешното приложение на методите на ИИ в тях.

Общата схема на ИАИС е дадена на фиг.18. ИАИС се състои от три основни части. Те са: връзка с потребителя, планираща или анализираща част и изпълнителна част.



Фиг. 18. Обща схема на ИАИС.

2.2.1. Връзка с потребителя

Тази част на ИАИС трябва да осигури съществена част от удобствата на потребителя при работа със системата. На първо място това е възможността да се използва удобен език. Като се има предвид, че потребителят не е програмист и че системата е предназначена за решаване на математически задачи, то е необходимо математически ориентиран ограничен естествен език. Съответният лингвистичен процесор, който възприема такъв език, е относително прост, понеже езикът, съответстващ на предметната област – математиката е добре формализиран. При работа на лингвистичния процесор трябва да се използват два вида знания: знания за езика и математически знания.

В развитието на езика на САП могат да се различат следните нива:

а) Алгоритмичен език. Такъв е, например, езикът на системата REDUCE и някои други развити САП;

б) Детерминиран език, чрез който заявките към системата се задават в обобщен вид (относно съдържанието на заявката). Например: "Да се реши линейното диференциално уравнение от II ред....", "Да се пресметне неопределеният интеграл ..." и т.н. За анализ на заявката и избор на метод се използват база математически знания.

в) Език на потребителя – непрограмист. Това е силно ограничен математически ориентиран естествен език. Например: "Да се реши следното уравнение", "Да се намерят корените на уравнението", "Непрекъсната ли е функцията....." и т.н. В този случай освен математически знания, в база

знания трябва да има и знания за естествения език.

Освен наличието на удобен за потребителя език е необходима възможност за получаване на консултации от системата относно възможностите на системата и състоянието на задачата, която се решава.

При разработването на тази част е нужно да се предвиди възможност за доуточняване на задачата или заявката на потребителя с цел да се избегнат случаите, в които лингвистичният процесор не може да се справи с конкретната заявка.

2.2.2. Планираща част

Планиращата част е втората основна част, която има съществено отношение към интелектуализацията на този тип системи. Нейната основна задача е на базата на възприетата чрез лингвистичния процесор заявка на потребителя да извърши:

- анализ на заявката относно наличието на средства в системата, необходими за решаването на задачата;
- анализ на задачата относно съвместимостта на типовете на математическите обекти;
- планиране на основните етапи на процеса на решаването на задачата.

Основно място в планиращата част заемат подходящо представените знания. На основа на тези знания потребителската заявка се трансформира в план, който се изпълнява със средствата от изпълнителната част. Знанията се разделят на следните групи:

- математически знания - те се използват и от лингвистичния процесор при възприемането на заявката на потребителя. Обикновено математическите знания са представени като правила и във вид на формули за извършване на подходящи преобразования на изрази с цел опростяването им;

- знания за алгоритмите - това са знания за пълните възможности на изпълнителната част. В тях са отразени всички обработващи програми за аналитични преобразования и числени пресмятания;

- обобщени знания - те описват в общ вид понятията и възможните действия на цялата ИАИС. Необходими са при анализирането на задачата относно възможността за решаването и;

- знания за обектите - представляват факти относно текущото състояние на математическите обекти, които се използват при решаване на задачата. Служат при анализа на задачата по отношение на съвместимостта на математическите обекти.

2.2.3. Изпълнителна част

Основа на изпълнителната част представляват програмните средства за работа с математически обекти. Поради това при създаването на ИАИС може да се използва съществено вече създадена САП. Ще разгледаме по-подробно двата основни компонента на изпълнителната част - множеството от алгоритми и изпълнителния процесор.

Алгоритмите в ИАИС се представят с подпрограми, които

извършват съответни действия над математическите обекти. Ще ги разделим на три основни групи:

- числени алгоритми. Те се изграждат на основа на крайни или безкрайни (итеративни) последователности от действия, за които по математически път е доказано, че водят до търсеното решение. Въпреки, че на практика полученото решение рядко е точно, то в редица случаи удовлетворява нуждите на потребителя. Присъстващата неточност в решението може да се дължи на две причини - от една страна това е свързано с приближеното представяне на числата в конкретната ЕИМ, а от друга - със заместването на безкрайната последователност от действия, водеща до точното решение, с крайна;

- алгоритми за аналитични преобразования. Те се изграждат на основа на крайни последователности от действия, за които по математически път е доказано, че водят до търсеното решение. Част от тях представляват аналози на числените алгоритми, като с помощта на подходящо представяне на данните се осигуряват изчисления с неограничена точност. Примери за такъв вид алгоритми са алгоритмите за умножение на матрици.

Редица резултати на съвременната математика доведоха до създаване на група от специални алгоритми за аналитично решаване на задачи, които преди това са решавани само приближено с помощта на числени алгоритми, например факторизация на полиноми. Понастоящем се наблюдава бурно развитие на този вид алгоритми, дължащо се както на успехите на съвременната математика, така и на новите постижения в информатиката, дали възможност математическите идеи да бъдат

реализирани във вид на програми за ЕИМ;

– евристични алгоритми. В последно време значително развитие достигнаха детерминирани методи за аналитични преобразования. Тези методи са добре формализирани и обикновено водят до намиране на точното решение на разглежданата задача. Ако, обаче, за решаването на дадена задача не са известни детерминирани методи, то се прилагат недетерминирани (евристични) методи, основани на евристики, тестове, аналогии и т. н. Тяхното използване в повечето случаи води до намиране на точното решение. В същност, първите съществени резултати в областта на САП са свързани именно с използването на евристични методи [Кариманиан 53], [Нолан 53] [Слейджл 67]. Тяхното включване в планиращата част повишава ефективността и интелигентността на съответната система.

Ще отбележим, че евристичните алгоритми се изграждат на основа на крайни последователности от действия, за които от евристични съображения се предполага, че водят в редица от случаите до търсеното решение. Най-общо казано, прилагането на евристики влияе динамично на последователността от действия – в зависимост от входните данни, извършването на дадено действие може да бъде отменено, може да бъде избрано едно или друго действие, или една или друга последователност от действия. С помощта на тези алгоритми могат да се решават редица много сложни задачи, без обаче да е сигурно, че всяка такава задача ще бъде решена.

Сега ще се спрем на изпълнителния процесор. Въз основа на плана, създаден от логическия процесор, изпълнителният процесор извършва решаването на задачата, като създава

конкретната програма, която се изпълнява под неговото управление. В тази програма съществено се използват готови подпрограми за аналитични преобразования, логически извод и др., съдържащи се в базата знания на изпълнителната част. За това са необходими следните подсистеми:

- подсистема за работа с библиотеки от програми. Тя управлява разполагането на програмите, реализиращи съответните алгоритми, в оперативната памет и се грижи за осигуряване на връзката между програмите. Освен това, тя осигурява възможност за включване на нови програми в изпълнителната част;

- подсистема за обмен на обекти. В случаите, когато математическите обекти са с голям обем т. е. заемат много памет и е необходимо да се въвеждат и извеждат на външна памет, то необходимо е тази дейност да се обособи в отделна подсистема;

- подсистема или група програми за връзка между аналитични и числени пресмятания. Сложността на тази подсистема зависи изключително от различието в езиковите средства и начините за представяне, използвани при програмирането за аналитични преобразования и алгоритмите за числени пресмятания.

2.2.4. Планиране на действията и представяне на знанията в ИАИС за работа с матрици

Представената в т. 1.2 на настоящата глава специализирана аналитично-изчислителна система за работа с матрици е представител на ИАИС [Герев 87]. Тук ще разгледаме

конкретните механизми, по които се извършва планирането на действията и свързаното с тях представяне на знанията.

Планирането на действията е основна функция на планиращата част. В специализираната ИАИС за работа с матрици планирането на действията включва избор на конкретни алгоритми, прилагането на които осигурява решаване на поставената в заявката задача и избора на областта, в която е необходимо да се представят математическите обекти.

Ще разгледаме представянето на знанията за обектите и алгоритмите, с които работи системата.

Обектите, с които работи потребителят, са най-динамичната част на специализираната система. Представянето им се определя от изискванията на базисните алгоритми или програми, реализиращи операциите в изпълнителната част.

По-сложно е представянето на знанията за алгоритмите. Избрана е структура, която съдържа три нива, като знанията от всяко следващо ниво обобщават в известен смисъл знанията от по-долното ниво и в същото време са ориентирани по-пряко към математическия аспект на конкретните действия.

На най-ниско ниво са знанията за алгоритмите (последователност от една или повече програми, която реализира конкретната операция върху обекти от определен тип и при определено представяне). На това ниво са необходими характеристиките на входните и изходните аргументи и начина, по който се настройват адресите при изпълнение на конкретните програми.

Следващото ниво е предназначено за обобщените алгоритми. Обобщените алгоритми представляват описание на

математическата операция, без да се гържи сметка за типа и областта на обектите, които участват в нея. Основната информация представлява множество от алгоритми, измежду които се избира определен алгоритъм, осигуряващ операцията при зададен тип и област на обектите.

Обобщените знания от най-високо ниво са свързани с така наречените процедури. Под процедура разбираме последователност от математически операции, или в нашите термини – последователност от обобщени алгоритми. Например, такава последователност може да е умножение на две матрици, намиране на детерминантата на произведението и факторизирането ѝ. Такива процедури могат да се дефинират от потребителя, а извличането на необходимата информация става от системата. Представянето на процедурите се състои от две части. В първата част, всеки ред представлява отделен допустим вариант за типовете и областите на входните аргументи. Втората част е самата последователност от обобщени алгоритми и указатели към номерата на участващите в операцията обекти.

Ще представим накратко работата с така описаните знания в случай, че имаме процедура за умножение на две матрици, намиране на детерминантата и факторизирането ѝ. Обектите, с които работим, са от тип матрица, а елементите ѝ може да са цели числа, рационални числа, полиноми и т.н. Търсим ред в описанието на процедурата, в който да съвпадат типът и областта на представянето на двете матрици. След това, за всеки обобщен алгоритъм от описанието на процедурата (умножение, детерминанта и факторизация) избираме такъв алгоритъм, който да подхожда по тип и множество на избрания

вариант. За тази цел се ползват указатели за номер на аргумент за процедурата и за номер на аргумент за обобщения алгоритъм. Накрая, относителните адреси се заменят с физическите адреси на матриците в базата от знания за обектите.

3. Общи заключителни бележки

Създаването на интелектуализирана САП не означава непременно разработване на нова система. Може конкретна, работеща и използвана САП да бъде развита по подходящ начин, с отчитане на основните аспекти при създаването на ИППС.

Ще отбележим, че приложението на подхода на предметно-ориентираните празни ИППС (описан в Глава I, т.3) към ИАИС се оказва особено полезно и перспективно при създаването на специализирани САП, които могат да се разглеждат като проблемно-ориентирани ИППС.

В Института по математика с Изчислителен център при БАН се провеждат изследвания и конкретни разработки по създаване на интелектуализирани АИС с използване на системите REDUCE и SAC-2. С помощта на SAC-2 е разработена автоматизирана среда за аналитични преобразования, която поема грижата за унификация на математическите обекти по тип и свързването им със съответни програми. Създадената в сектор "Изкуствен интелект" система за работа с матрици [Геров 87], с използване на SAC-2 като инструментално средство, в значителна степен съдържа описаните по-горе елементи на интелектуализираната АИС. Създадената с помощта на REDUCE

система за работа с бази от символни преобразования [Спиридонова, Джамбазова 84] позволява създаването, поддържането и прилагането на бази от математически знания, записани с последователности от формули и процедури.

Направлението на работа в ИМ с ИЦ при БАН, свързано с интелектуализацията на АИС, намери отражение в тематиката на РГ 18 на Комисията по научни въпроси на изчислителната техника на Академиите на науките на страните-членки на СИВ. На тази тема беше посветен един раздел на изданието заключителен отчет на тази група [Геров, Капитонова, Спиридонова, Томов 84].

САП имат голямо научно и приложно значение. Те представляват широко поле за научно-изследователска и приложна дейност за специалисти от различни клонове на класическата математика и информатиката, при това, особено актуални са интелектуализацията на САП и създаването на специализирани интелектуализирани аналитично-изчислителни системи за редица области от математиката, физиката, техниката и др.

ГЛАВА VII. ИНТЕЛЕКТУАЛИЗИРАНИ ПРИЛОЖНИ ПРОГРАМНИ СИСТЕМИ ЗА НАУЧНО-ТЕХНИЧЕСКИ ЗАДАЧИ (ИППС ЗА НТЗ)

В настоящата глава са представени някои идеи и разработки от кандидатската дисертация на автора [Томов 75δ] и тяхното продължение и развитие като ИППС.

1. СИСТЕМЕН ПОДХОД ПРИ СЪЗДАВАНЕ НА МАТЕМАТИЧЕСКО ОСИГУРЯВАНЕ ЗА НАУЧНО-ТЕХНИЧЕСКИ ЗАДАЧИ

Използването на ЕИМ в изследователската и приложна дейност в областта на научно-техническите задачи води до необходимостта от създаване на съответно математическо осигуряване, което трябва не само правилно да реализира алгоритъма за решаване на задачата, но и да дава възможност за удобно и ефективно провеждане на изследователската и приложна дейност при по-сложни условия като многовариантност, диалог между човека и машината, данни, съдържащи грешки и др.

Системният подход при разработката на математическо осигуряване за научно-технически задачи дава възможност в значителна степен тези изисквания да бъдат удовлетворени [Томов 75а].

1.1. Характерни особености на научно-техническите задачи във връзка с решаването им на ЕИМ

Тези характерни особености се състоят в следното:

- а) Голям обем изчислителна работа и сравнително малък обмен с входните и изходните устройства.
- б) Неоъходимост от работа в диалогов режим при изпълнение на програмите. Преглед алгоритмичната сложност на повечето научно-технически задачи, често се налага намесата на човек при решаването им от ЕИМ. Обикновено след анализ на междинните резултати човекът взема решение за продължаване на изчислителния процес по един или друг начин или за неговото прекратяване.
- в) Изследване на много варианти на задачата в зависимост от изменение на параметрите, а понякога – и на алгоритъма. Обикновено вариантите са зависими помежду си и се получават един от друг чрез извършване на съответни изменения.
- г) Неоъходимост от библиотеки и архиви от данни, даващи възможност за записване във външната памет на цели варианти или части от тях. Осигуряване на удобен и ефективен достъп до данните.
- д) Неоъходимост от запомняне на междинни варианти и резултати.
- е) Неоъходимост от графичен, визуален и звуков вход и изход.

1.2. Основни елементи на математическото осигуряване за научно-технически задачи

Освен математическото осигуряване с общо назначение като например операционната система, за решаване на научно-технически задачи се ползува и съобразено с тяхната специфика математическо осигуряване, по-важните елементи на което са:

а) Езици за програмиране и съответни транслятори.

Особено подходящ алгоритмичен език за научно-технически задачи е ФОРТРАН, тъй като притежава силно развити средства за реализиране на алгоритмични и логически операции, за работа с подпрограми и функции, за вход и изход и др.

Алгоритмичният език PL/I от гледище на приложението му за решаване на научно-технически задачи може да се разглежда като разширение и развитие на ФОРТРАН, свързано с редица допълнения и усъвършенствувания като наличие на възможност за вариране на точността, работа с битове, оператори за обмен с големи възможности и др.

Някои удачно разработени автокодове също представляват удобни средства за програмиране на научно-технически задачи.

б) Пакети стандартни програми за типични и често употребявани изчислителни процеси - с общо математическо назначение, математическо оптимизиране, теория на графите и др.

в) Специализирани езици за определени класове научно-технически задачи.

г) Езици, транслятори и програмни системи за реализиране на диалогов режим и за работа с някои от

периферните устройства като дисплеи, плотери и др.

1.3. Същност и възможности на системния подход при създаване на математическо осигуряване за научно-технически задачи [Томов 75a]

Обикновено, при разработката на математическо осигуряване за научно-технически задачи най-голямо внимание се отделя на точното и оптимално програмиране на изчислителния алгоритъм, без да се полагат особени грижи за създаване на програмни средства, които да са съобразени с изискванията от т. 1. 1 на тази глава. При това, контролът на обработената информация най-често се извършва частично и непълно.

Всичко това прави разработеното математическо осигуряване до голяма степен неефективно и неудобно за изследователска и приложна дейност.

Прилагането на изискванията и методите на системното програмиране обаче може да доведе до отстраняването на значителна част от тези недостатъци.

Във връзка с това, математическото осигуряване за решаването на една или клас научно-технически задачи е подходящо да се разглежда като програмна система, съставена от два взаимно свързани модула:

- Обработващ модул, реализиращ алгоритъма за решаване на съответната научно-техническа задача.

- Операционен модул, реализиращ както някои общи изисквания на математическото осигуряване, така и необходимите възможности на програмната система за

удовлетворяване на условията от т. 1.1 на настоящата глава.

Във връзка с разработката на операционния модул могат да бъдат отбелязани някои основни негови елементи:

а) Система за синтактичен контрол на обмена на информация между оперативната памет и периферните устройства. Основно значение в този случай има контролът, произхождащ от съответния транслятор.

б) Система за синтактичен и семантичен контрол на обмена, отчитаща специфичните особености на задачата, отразяващи се върху структурата и смисъла на обменяната информация.

При а) и б) от значение е разработката на подходяща система от индикации за констатираните грешки.

в) Изменение на трансляторите с цел изпълнението на преведената програма да не спира при констатирането на първата поред грешка в обменяната информация, а да се извършва прекъсване, което да дава възможност на програмната система да обработи грешката и да продължи обмена с цел да се констатират и други възможни грешки.

г) Програмни средства за работа в многовариантен режим, състоящ се в автоматично изпълнение на множество варианти, които се получават един от друг чрез въвеждане само на необходимите изменения и са свързани в дървовидна структура.

д) Подходящи формати на входните данни, даващи възможност за лесно преминаване от един вариант към друг.

Целесъобразно е всеки формат да съдържа съответни спецификатори, идентифициращи вида на включените в него данни. Това дава възможност данните да се въвеждат независимо една от друга, да бъдат изменяни съгласно

принципа на наслагването и да се опрости първоначалната им обработка.

е) Програмни средства за образуване и използване на база от данни.

Поради естеството на научно-техническите задачи, обикновено не се извършва последователна обработка на данните, а е необходимо базата да осигурява възможност за работа с отделни групи и подгрупи от данни, а също така и с техни обединения.

В по-прости случаи е целесъобразно създаването на библиотека от данни, която дава възможност за съхранение и обработка на комплекти от данни за отделни варианти.

ж) Работни файлове, даващи възможност за записване във външната памет на междинни резултати и варианти и за тяхното ползване след това.

з) Средства за контрол и анализ на изчислителния процес по време на изпълнение и подходящи индикации. Осигуряване на съответна автоматична реакция на програмната система при извършване на прекъсване с цел да се даде възможност за намеса на човека в изчислителния процес в случай, че е констатирана в хода на пресмятанията някоя предварително определена ситуация.

и) Езици и съответни транслатори за работа в диалогов режим по време на изпълнението.

к) Възможност за ползване в текущия вариант на някои резултати, получени в предишните варианти.

л) Възможност за автоматично поправяне на някои от констатираните грешки и даване на съответни предупредителни индикации.

Качествата и възможностите на операционния модул са тясно свързани с възможностите на апаратната част и операционната система на съответната ЕИМ.

Системният подход е приложим не само за научно-технически задачи, но и за широк кръг изследователски задачи и приложни задачи от областта на математиката, физиката, химията, биологията, моделирането, изследването на операциите и др.

За разработка на математическо осигуряване с описаните възможности се изисква образуването на смесени колективи от специалисти по съответната задача и по математическо осигуряване.

Основните принципи на описания системен подход са предложени при разработването на програмната система за вариантен анализ на вентилационни системи.

като по време, така и по обем.

2. ИНТЕЛЕКТУАЛИЗАЦИЯ НА ПРИЛОЖНИ ПРОГРАМНИ СИСТЕМИ ЗА НАУЧНО-ТЕХНИЧЕСКИ ЗАДАЧИ

обучение и самонаблюдение.

Не е трудно да се види, че развитието на идеята на системния подход при създаване на математическо осигуряване за научно-технически задачи води до интелектуализация на това математическо осигуряване и до създаване на ИППС за научно-технически задачи.

Операционният модул при системния подход може да се разглежда като първообраз на интерфейсия блок и логическия процесор при ИППС. Аналогично, развитието на обработващия модул води до изпълнителния процесор.

Освен възможностите, които осигурява прилагането на

системния подход при създаването на математическо осигуряване за научно-технически задачи, интелектуализацията на това математическо осигуряване довежда до придобиването на нови възможности и повишава неговата ефективност и удобствата за потребителя.

Ще отбележим само някои от най-важните качества, които математическото осигуряване за научно-технически задачи добива в резултат на интелектуализацията:

- интерфейс "човек-машина" на език, близък до естествения, при това специализиран за съответната научно-техническа област;
- автоматизиране (изцяло или частично) на избора на метод за решаване на задачата, планирането на действията и управлението на процесите;
- автоматизиране (изцяло или частично) на многовариантния анализ както по данни, така и по алгоритми (генериране на различните варианти чрез изменение както на данните, така и на алгоритмите);
- обучение и самообучение;
- визуален и звуков вход и изход.

3. ПРИЛОЖНА ПРОГРАМНА СИСТЕМА ЗА ВАРИАНТЕН АНАЛИЗ НА ВЕНТИЛАЦИОННИ СИСТЕМИ И НЕЯНОТО ИНТЕЛЕКТУАЛИЗИРАНЕ

При проектиране, реконструкции, аварии и оперативен контрол на вентилационните системи обикновено възниква необходимост от пресмятане и изследване на потокоразпределението за различни варианти на системата. При това, главно поради сложността и големия обем на

изчислителната работа, характерни за анализа на съвременните вентилационни системи, а също и поради необходимостта от вързо извършване на изследвания по време на аварии, се налага да бъдат използвани ЕИМ и във връзка с това – да се разработва съответно математическо осигуряване.

3.1. Пресмятане на потокоразпределението във вентилационни системи

Вентилационната мрежа се представя като ориентиран свързан граф, за всеки клон на който е зададено аеродинамичното съпротивление, а също така фиксирания дебит или характеристиката на източника на тяга, ако има такава. За пресмятане на потокоразпределението се съставят възловите и контурните уравнения на Кирхоф и получената нелинейна система от алгебрични уравнения, в която неизвестни са дебитите в клоновете на графа, се решава по итерационния метод на Крос [Сул 60], [Чанг, Хартман 66]. За ускоряване на сходимостта, базата от независими контури в графа, относно която са съставени нелинейните уравнения на системата, се избира така, че общите клонове на независимите контури да са с възможно най-малки аеродинамични съпротивления. За целта чрез една модификация на метода на Крускал, в графа се определя минимално по отношение на аеродинамичните съпротивления дърво. Паралелно с генерирането на дървото, за всеки възел се определя възелът – предшественик, нивото и клонът – предшественик. Познаването на предшествениците дава възможност за всеки клон j от графа, който не принадлежи на дървото, лесно да бъдат определени веригите μ и ν от

клоновете на гървото, свързващи съответно началния и крайния възел на j с корена на гървото. Клонът j и неоштите клонове на μ и ν определят независимия контур през j [Оре 68], [Томов 75].

По посочения по-горе метод за пресмятане потокоразпределението са разработени алгоритъм и програма VENT за пресмятане на естественото и регулираното потокоразпределение във вентилационни системи [Томов, Велчев 75]. Преимуществата на програмата VENT в сравнение с описаните в [Чанг, Сапърщайн 70] и [Нова технология... 69] програми са: значително по-бърз алгоритъм за образуване на базата от независими контури, спиране на итерационния процес при достигане на зададен толеранс на разликите между неизвестните при две последователни итерации и отпечатване на максималната по абсолютна стойност несвързка на нелинейните уравнения на Кирхоф за неизвестните от последната итерация, обстоен анализ на входните данни във връзка с констатирането на допуснати в тях грешки и др.

Разгледаните алгоритми дават възможност за анализ на големи и сложни вентилационни системи, което без използването на ЕИМ практически е невъзможно, а също така за по-точни и задълбочени изследвания, което води до вземане на по-ефективни решения при регулиране и управление на вентилационните системи.

Многобройните приложения на програмите в практиката недвусмислено потвърдиха тези изводи.

Провежданият обстоен анализ на алгоритмите, програмните реализации и операционните възможности на програмите за пресмятане на покоторазпределението, както и опитът от

експлоатацията на тези програми показаха и някои техни недостатъци, по-важни от които са:

- Липса на ефективни и удобни за потребителя възможности и средства за ползуване на програмите в многовариантен режим, възникващ при анализа на вентилационните системи;

- Неоптималност на някои алгоритми по отношение на изразходваното машинно време;

- Липса на възможности за ползуване на база от данни;

- Липса на възможности за ползуване на работни файлове за съхраняване на междинни резултати и варианти;

- Незадоволителен контрол при обмен и обработка на информацията.

Подобни недостатъци притежава голяма част от математическото осигуряване за решаване на научно-технически задачи.

Анализът на тези недостатъци показва, че е необходимо да се прилага системен подход при създаването на математическо осигуряване за научно-технически задачи [Томов 75а].

3.2. Програмна система за вариантен анализ на вентилационни системи

При разработването на програмната система са приложени основните принципи на системния подход за създаване на математическо осигуряване на научно-технически задачи.

Анализът на вентилационната система обикновено се свежда до пресмятане и изследване на потокоразпределението

за различни варианти на структурата на мрежата и значенията на параметрите на системата.

Програмната система дава възможност за анализ на вентилационни системи в многовариантен режим, за работа с библиотеки от данни и за ползуване на работна лента, като при това извършва обстоен анализ на обработваната информация.

Програмната система е разработена на ФОРТРАН (за транслятора ФОР 32 за ЕИМ МИНСК-32, а по-късно е прехвърлена на ЕС 1040 и IBM PC/XT) и се състои от два взаимно свързани модула:

- Обработващ, който извършва пресмятането на потокоразпределението във вентилационната система;
- Операционен, който реализира многовариантния режим, работата с библиотеката и работната лента, основната част от контрола на информацията и др.

Обработващият модул се състои от програмата VENT за пресмятане на естественото и регулираното потокоразпределение в мрежата.

Операционният модул на програмната система дава възможност за:

- Пресмятане, с помощта на обработващия модул, на потокоразпределението за различни варианти на вентилационната система, получени в резултат на изменение структурата на мрежата, аеродинамичното съпротивление на клоновете, естествените тяги, местата и характеристиките на вентилаторите, местата и стойностите на фиксираните дебити и др;

- Работата с библиотека на носител магнитна лента, в

която за различни вентилационни системи се съхранява необходимата за тяхното анализиране информация – структура на мрежата, параметри на системата и др. Обикновено в библиотеката се записват основни варианти за дълговременно съхраняване и ползуване при реконструкции, управление, аварии и др. Библиотеката дава възможност чрез програмната система да се третират множество вентилационни системи;

– Записване на работна лента на някои варианти и ползуването им след това (след анализ от потребителя на системата) като база за образуване на други варианти.

Вариантите, които могат да се третират чрез програмната система, са свързани помежду си в дървовидна структура. Формирането на един вариант от друг се извършва съгласно принципа на премълчаването – въвеждат се поправки само за параметрите, които трябва да бъдат изменени, като останалите запазват предишните си стойности.

Базис наричаме вариант, който се формира като всички данни се въвеждат от перфокарти или от библиотеката, или от работна лента.

Всеки небазисен вариант се образува чрез въвеждане на изменения относно предишния вариант. В частност, предишният вариант може да бъде базисен вариант.

Вариантите, разположени по възлите на верига, свързваща корена на дървото на вариантите с някои от свободните му върхове, се образуват последователно един от друг чрез въвеждане на съответните поправки, като се започва с базисен вариант. От произволен възел на такава верига може да се образуват разклонения, които също представляват вериги. Обикновено вариантът, от който се въвежда разклонение, се

третира като базисен, което в повечето случаи предполага, че при неговото образуване е бил записан на магнитна лента.

Дървото на вариантите се представя като линейна поредица от варианти в реда на тяхното образуване и се оформя като отделен пакет от перфокарти. Поправките за всеки вариант се обособяват с помощта на разделители в отделна група от перфокарти, в която подреждането им, с някои изключения, може да бъде произволно.

За всеки текущ вариант се анализират въведените поправки, с цел да се установи, дали те предизвикват промяна на графа или на минималното дърво от предишния вариант. Ако такава промяна няма, базата от независими контури от предишния вариант се ползува наготово в текущия вариант и решението на предишния вариант се приема за начално приближение на решението на текущия вариант. Във връзка с това за всеки вариант, записан в библиотеката или на работна лента се съхраняват не само необходимите данни, но и информация за независимите контури и решението на варианта.

Всеки вариант в библиотеката се идентифицира чрез своето име. Програмната система осигурява автоматично извършване на записа, търсенето, четенето и контрола на информацията в библиотеката.

Поради това, че в редица случаи като аварии, оперативен контрол и др., е особено необходим бърз и надежден достъп до информацията в библиотеката, тя е дублирана на две магнитни ленти. В случай, че някоя порция информация не може да бъде прочетена от първата магнитна лента, автоматично се прави опит за нейното прочитане от втората магнитна лента.

Структурата на информацията върху работната лента и

начинът на работа при нея са както при библиотеката, но без дублираща лента.

При определянето на структурата на входната информация са взети предвид редица съображения, свързани с техническото естество на задачата, вариантният режим на работа, прилагането на принципите на премълчаването и наслагването при въвеждане на изменения, стандартното представяне на данните с цел облекчаване на тяхната предварителна обработка и подготовка, коригиране на констатирани грешки и др.

За всеки тип входни данни е определен съответен формат, който съдържа спецификатор на типа (обикновено мнемоничен код) и информационна част (стойности, признаци, имена и др.). Това дава възможност, за всеки вариант, входните данни да са до голяма степен независими от реда на тяхното въвеждане.

Например, форматът T M M1 M2 R служи за въвеждане на клон от графа. T е спецификатор, M, M1, M2 – са съответно номерата на клона, на началния и крайния му възел, а R представлява аеродинамичното съпротивление на клона.

По подобен начин са разработени и други формати, предназначени за задаване на базисния вариант, на естествените тяги, фиксираните дебити, местата и характеристиките на вентилаторите, името на варианта, за обмен на информация с библиотеката и работната лента, за изтриване на информация, за коментар и др. Общият брой на форматите е 24.

Отпечатват се във вид на таблица пълният комплект входни данни за варианта и полученото решение – дебитите и загубите на налягането във всеки клон, работните режими на

вентилаторите, депресиите, които трябва да се регулират в клоновете с фиксирани дебити и др.

При своята работа, за всеки вариант, програмната система извършва контрол на информацията и дава съответни индикации във връзка с откритите формални грешки във входните данни, спазването на техническите ограничения на задачата, образуването на вариантите, свързаността на графа, работата с библиотеката и работната лента, завършването на някои дейности и др.

Един вариант се счита за некоректен, ако съдържа грешки или се образува относно некоректен вариант. За такъв вариант обработващият модул на програмната система не се изпълнява, а само се извършва анализ за наличието на грешки и след това се преминава към следващия вариант.

Програмната система самостоятелно извършва поправки на някои грешки и дава съответни предупредителни индикации. Общият брой на индикациите е 43.

Програмната система се състои от 5 програмни части, които съдържат общо около 1200 оператора.

Максималният брой на клоновете на графа за един вариант е 400. Допускат се до 30 различни характеристики на вентилатори, всяка от които се задава с не повече от 20 точки. Общият брой на клоновете с фиксирани дебити и с вентилатори не трябва да надминава цикломатичното число на графа.

3.3. Общ проект за интелектуализиране на приложна програмна система за вариантен анализ на вентилационни системи (ППС за ВАВС)

Както отбелязахме по-горе (т.3.2 от настоящата глава), при създаването на ППС за ВАВС е приложен системния подход.

Върху нея би могло да се извърши интелектуализиране, в резултат на което тя би се превърнала в ИППС за ВАВС. Това би довело до разширяване на възможностите, повишаване на ефективността и подобряване на удобствата за работа на ИППС за ВАВС.

Ще отбележим някои от основните възможности и качества, които системата придобива в резултат на нейното интелектуализиране:

а) Ограничен естествен език, специализиран в областта на вентилационните системи. Той ще замества досегашния форматен език на неинтелектуализираната система.

Ще илюстрираме възможностите на ограничения естествен език с един елементарен пример. Операцията "въвеждане на нов клон в системата" ще може да се задава на системата по различни начини като:

"Внеси клон №107 начало 7 край 12"

"Въведи клон (7, 12)" - в този случай системата сама определя и поставя номер на клона и го съобщава на потребителя.

Ще отбележим, че синтаксисът на ограничения естествен език включва най-простите структури на описателни, повелителни и др. изречения, а речникът е крайно ограничен.

Общата структура на изреченията (описателни и

повелителни) е от вида

<подлог> <сказуемо> <допълнение> <пояснение>

където подлогът, допълнението и пояснението са групи съществителни или предложни групи. Сказуемото е глаголна група.

В повелителните изречения подлогът обикновено представлява обръщение и най-често липсва, а сказуемото е глагол във второ лице.

Аналогично се третират и въпросителните изречения.

Очевидно гъвкавостта на езика е голямо удобство, особено при аварийни ситуации и при наличието на гласова връзка "човек-машина".

δ) Автоматично (изцяло или частично) генериране и избиране на варианти чрез вариране на данните. Тук може да се приложи станалия вече класически метод на изкуствения интелект "търсене в пространството на състоянията".

Необходимо е да се определят поправките и критериите (от техническа гледна точка) съответно за генериране и избиране на варианти на вентилационната система. Въз основа на тези поправки и критерии могат да се дефинират двете основни функции: генератор на перспективни варианти и оценяваща даден вариант функция.

Спирането на търсенето в пространството от варианти става при достигането на целева стойност на оценяващата функция. Вариантът, за който е постигната тази стойност, се приема за решение на задачата. Спирането на търсенето може да стане и без да е достигната целева стойност на оценяващата функция, ако се е изчерпал ресурсът (по време или по памет). В такъв случай за решение (очевидно

приблизително) се приема вариантът, за който оценяващата функция има стойност, най-близка в някакъв смисъл до целевата.

Ще отбележим някои от най-често срещаните се поправки на данните, водещи до различни варианти на вентилационната система:

- промяна на аеродинамичното съпротивление на някои клонове;
- въвеждане, премахване или промяна на фиксиран дебит за даден клон;
- въвеждане на нов клон в мрежата;
- премахване на клон от мрежата;
- въвеждане, преместване или премахване на вентилатори в различни клонове на вентилационната система;
- промяна на характеристиките на вентилатор;
- въвеждане, преместване или премахване на врати в различни клонове на вентилационната система;
- отваряне или затваряне на врати в някои клонове.

Оценяващата функция оценява даден вариант по различни показатели като:

- изразходвана електроенергия за вентилация на мината;
- осигуряване на нужното проветряване в дадени клонове (в които има хора или съоръжения);
- премахване (или свеждане до минимум) на проветряването с чист въздух на район от мината, в който има пожар с цел неговото изолиране и ликвидиране на пожара;
- други.

Вариантът на параметрите на системата с цел генериране на различни варианти става по определени правила,

произтичащи от теорията на вентилационните системи и диспечерската практика с тях. Тези правила отразяват знанията за структурата и управлението на вентилационната система при различни ситуации. Подходящо е те да бъдат организирани в база от знания за предметната област "структура и управление на вентилационните системи".

Обикновено правилата са от вида

Ако < условие > то < действие >.

Тук < условие > представлява описание на ситуацията (във вентилационната система), при наличието на която в < действие > е указано кои параметри и как да се варират с цел генерирането на подходящи варианти за изследване.

За всеки вариант, системата пресмята основни характеристики на вентилацията като дебитите на въздуха и регулиращата депресия за всеки клон, работните режими на вентилаторите и др.

Въз основа на тези характеристики и на вложените данни за всеки вариант, оценяващата функция пресмята оценка на варианта. Тази оценка представлява числена или логическа стойност.

Тя показва дали е постигната целта на многовариантния анализ или не, а при спиране на анализа поради изчерпване на ресурса се използва при определяне на най-подходящото приблизително решение. Ако за даден текущ вариант оценката показва, че целта не е достигната, пресметнатите характеристики на вентилацията и вложените данни за този вариант се използват в условията на правилата с цел генериране на нови варианти и продължаване на многовариантния анализ. Това всъщност е вложение на правила

едно в друго.

В базата знания се включват и правила, в които < условие > е по същество описание на ситуация във вентилационната система (или в част от нея), а в < действие > се поставя оценяващ алгоритъм, който използва характеристиките на разглеждания (текущ) вариант. Тези правила се ползват от оценяващата функция.

в) Може да има правила както от генериращ, така и от оценяващ тип, които да предизвикат промени в алгоритмите за изчисления, генериране, оценка и др.

Очевидно, генераторът на перспективните варианти и оценяващата функция е подходящо да бъдат разработени като продукционни системи, ползващи правилата от базата знания за структурата и управлението на вентилационната система.

В заключение ще отбележим, че интелектуализацията на МО за НТЗ е важен и перспективен етап в неговото развитие и представлява широко поле за бъдещи научни изследвания и приложения.

СПИСЪК НА СЪКРАЩЕНИЯТА

АИС	- аналитично/и изчислителна/и система/и
БД	- база данни
БЗ	- база знания
ВАВС	- вариантен анализ на вентилационни системи
ЕС	- експертна/и система/и
ЕСА	- експертна система за атестиране на кадри
ЕСИТ	- интелектуализирана въпросно-ответна система за анализ на историко-географски текстове за гревна Тракия
ЕСОК	- експертна система за оценка на кадри
ИАИС	- интелектуализирана/и аналитично-изчислителна/и система/и
ИВОС	- интелектуализирана/и въпросно-ответна/и система/и
ИЗ	- интерпретатор на знания
ИИ	- изкуствен интелект
ИМО	- интелектуализирано математическо осигуряване
ИППС	- интелектуализирана/и приложна/и програмна/и система/и
ИС	- информационна система
НТЗ	- научно-технически задачи
ОЕЕ	- ограничен естествен език
ПОПЕС	- предметно-ориентирана/и празна/и експертна/и система/и

- ППС – приложна/и програмна/и система/и
- САП – система/и за аналитични преобразования
- УБЗ – управление на база знания
- LIPES – експертна система за получаване на липозоми
- PREFES – експертна система за получаване на прлотопласти
- RIRL – език за представяне на правила

Л И Т Е Р А Т У Р А

1. [Ангелова, Димитров 86]
Angelova, M. and Dimitrov, D. S. Liposome Electroformation, In: Faraday Disc. Soc. 81 (1986) (in print).
2. [Бар, Фегенбаум 81]
Barr, A. and Feigenbaum, E. The Handbook of Artificial Intelligence. Vol. 1. Pitman, London, 1981 г., 409 p.
3. [Бар, Фегенбаум 82]
Barr, A. and Feigenbaum, E. The Handbook of Artificial Intelligence, vol. II, William Kaufmann Inc., Los Altos, California, 1982, 428 p.
4. [Бонг и др. 64]
Bond, E. and al. FORMAC - an experimental Formula Manipulation Compiler, Procc. of the 19th ACM National Conf., 1964.
5. [Браун 64]
Brown, W. The ALPAK system for non-numerical algebra on a digital computer, Bell System Technical Journal, 42(1963), No 5; 43(1964), No 2.
6. [Буханан и др. 69]
Buchanan, B. et al. Heuristic DENDRAL: a program for generating explanatory hypotheses in organic chemistry, In: Machine Intelligence, Edinburgh, 1969, vol. 4,

pp. 209-254.

7. [Буханан, Фегенбаум 78]

Buchanan, B. and Feigenbaum, E. Dendral and Meta Dendral: Their applications dimension, Journ. of AI, vol II, 1978, No 5, pp. 5-24.

8. [Буханан, Шортлиф 84]

Buchanan, B. G. and Shortliffe, E. H. (editors), Rule-Based Expert Systems, Addison-Wesley, Mass., 1984.

9. [Ван Мелс 79]

Van Mells, W. A domain independent production rule system for consultation programs, Proc. IJCAI-79, Tokyo, 1979, pp. 923-925.

10. [Ван Хулцен, Калме 82]

Van Hulzen, J.A. and Calmet, J. Computer Algebra Systems, Computing Supplementum 4, Springer Verlag, 1982, pp. 221-243.

11. [Гергт, Тарасов, Ширков 80]

Гергт, В. П., Тарасов, С. В., Ширков, Д. В. Аналитические вычисления на ЭВМ в приложениях к физике и математике. Успехи физических наук, 1980, т. 130. вып. 1., с. 113-147.

12. [Геров 80]

Геров, А. ЛISP - справочник по програмиране. ЕЦММ при БАН, 1980, 34 стр.

13. [Геров 87]

Геров, А. Специализирана аналитично-изчислителна система за работа с матрици. Автореферат, София, 1987, 30 стр.

14. [Геров, Григоров 83]

Геров, А., Григоров, А. Аналитично-вычислительная система для работы с матрицами. Конф. по системам для автоматизированного обслуживания, Варна, 3-8 октября 1983 (под печат).

15. [Геров, Дечева 83]

Геров, А., Дечева, Е. Аналитическое вычисление собственных значений и собственных векторов матриц. Сб. трудов Межд. конф. "Автоматизация научных исследований", Пловдив, 15-20 октября 1983, с. 232-239.

16. [Геров, Капитонова, Спиридонова, Томов 84]

Геров, А., Капитонова, И., Спиридонова, М., Томов, В. Интеллектуальные системы аналитических преобразований. В "Представление знаний в человеко-машинных и робототехнических системах", том. С "Прикладные человеко-машинные системы, ориентированные на знания", Рабочая группа 18 КНВВТ, М., ВЦ АН СССР, ВИНТИ, 1984, стр. 112-136.

17. [Геров, Куманов 84]

Геров, А., Куманов Н. Автоматизированная среда для аналитических преобразований. Proc. of the Intern. Conf. on AIMSA, Varna, 17-20 Sept., 1984, ЦИНТИ, MP16.

18. [Григоров 87]

Grigorov, A. Natural Language Parsing and Generation in the ESIT Expert System, Int. Conf. on Advanced Dialog Systems and Natural Language Processing, Suwalki, June 14-21, 1987 (in print).

19. [Григоров 87a]

Григоров, А. Представление знаний в экспертной системе ЕСИТ, Первый научный семинар с международным участием

"Проблемы и применения искусственного интеллекта",
Варна, 21-25 сентября 1987 (пог печат).

20. [Делияни, Ковалски 81]
Deliyanni, A. and Kowalski, R. A. Logic and Semantic Networks, Comm. ACM. Vol. 22, No. 3, pp. 184-192.
21. [Деранзарт 79]
Deransart, P. The Language LISP Does not Exist? Sigplan Notices No. 5, 1979.
22. [Дейвис, Ленат 82]
Davis, R., Lenat, D. Knowledge-Based Systems In AI, McGraw-Hill Inc., 1982, pp. 227-490.
23. [Димитров 83]
Dimitrov, D. S. Progress in Surface Science, 12 (1983) pp. 295.
24. [Димитров 86]
Dimitrov, D. Biophysical Engineering in Biotechnology, In: Proc. 4-th Int. Symp. on Biotechnology, Varna, 1986 (in print).
25. [Димитров, Джейн 84]
Dimitrov, D. S. and Jain, R. K. Biochim. Biophys. Acta, № 779, 1984, pp. 437.
26. [Димитров и др. 87]
Dimitrov, D. et al. Expert Systems for Mechanical Parts Design. In: "Artificial Intelligence: Methodology, Systems, Applications", Bulgaria, (in print).
27. [Димитров, Томов, Шарков, Ангелова 87]
Dimitrov, D., Tomov, V., Sharkov, G., Angelova, M. Liposome Production and Protoplast Electrofusion Expert Systems". In: "Artificial Intelligence:

Methodology, Systems, Applications", Ph. Jorrand, V. Sgurev (editors), North - Holland, Amsterdam, 1987.

28. [Дуга и гр. 79]
Duda, R. et al. Model design in the PROSPECTOR consultant system for mineral exploration, In: Expert Systems in the Microelectronic Age, Edinburgh University Press, 1979, pp. 153-167.
29. [Евтимов 80]
Евтимов, В. Конъюнктура и технология. Анализ 80, ИП-048-ИО-1-80-1, Интерпрограма, София, март 1980.
30. [Кариманиан 53]
Kahrimanian, H. Analytical differentiation on a digital computer, M.A. Thesis, Temple University, Philadelphia, 1953.
31. [Калме, Ван Хулцен 82]
Calmet, J., van Hulzen, J. A. Computer Algebra Applications, Computing Supplementum 4, Springer-Verlag, 1982, pp. 245-258.
32. [Клоксин, Мелиш 82]
Clocksin W., Mellish C. Programming in Prolog, Springer Verlag, Berlin, 1982, 576 p.
33. [Ковалски 79]
Kowalski, R. Logic for Problem Solving, North-Holland, 1979, 287 p.
34. [Колинз, Лоос 80]
Collins, G., Loos, R. SAC-2 Documentation, 1980.
35. [Ленат и гр. 83]
Lenat, D. et al. A program that learns new heuristics and domain concepts, Artificial Intelligence, vol. 21,

- 1983, pp. 61-98.
36. [Макгермот 80]
McDermott, J. R: A Rule-Based Configurer of Computer Systems, Report CMU-CS-80-119, April 1980, 55 p.
37. [Маккарти 60]
McCarthy, J. LISP Programmer's Manual, March, 1960.
38. [Маккарти 60a]
McCarthy, J. Recursive Functions of Expressions and Their Computation by Machine, Part I, Comm. ACM 3 (1960) 4, pp. 184-195.
39. [Маккарти 62]
McCarthy, J. et al. LISP 1.5 Programmer's Manual, MIT Press, Cambridge, Mass., 1962.
40. [Маурер 76]
Маурер, У. Введение в программирование на языке ЛИСП. Пер. с англ. Изд. "Мир", Москва 1976.
41. [Мински 74]
Минский, М. Фреймы для представления знаний. Москва, Энергия 1974 г. 151 с.
42. [Нова технология ... 69]
Nouvelles techniques d'etude et de controle de la ventilation miniere, Institut d'Hygiene des Mines a Hasselt et l'Universite Catholique de Louvain, 1969.
43. [Нолан 53]
Nolan, J. Analytical differentiation on a digital computer, M.A. Thesis Cambridge, Mass., MIT, 1953.
44. [Нордстром 70]
Nordstrom, M. et al. LISP F1 - a Fortran implementation of LISP 1.5, Computer Science Dept., Uppsala univ.,

Sweden, 1970.

45. [Нишева, Тонев 84]
 Нишева, М., Тонев, Т. Основные возможности аналитично-вычислительной системы для работы с непрерывными графами. Сборник трудов Межд. конф. "Автоматизация научных исследований", Пловдив, 15-20 октября, 1984, с. 398-404.
46. [Оре 68]
 Оре, О. Теория графов, "Наука", Москва, 1968.
47. [Попов 78]
 Попов, Г. Ръководни кагри. Критерии и методи за оценка. София, Партиздат, 1978.
48. [Поспелов 86]
 Поспелов, Д. А. Ситуационное управление: теория и практика. М. Наука, 1986, 288 стр.
49. [Представяне 84]
 Представление знаний в человеко-машинных и роботехнических системах. Том А. Фундаментальные исследования в области представления знаний. ВЦ на АН на СССР, 1984 г. Сб. статьи под ред. на Д. А. Поспелов.
50. [Робертс, Голдстейн 77]
 Roberts R., Goldstain P. The FRL Manual. MIT - 1977 г. 29 p.
51. [Сангевол 73]
 Sandewall, E. Programming in an Interactive Environment: the "Lisp" Experience. Computing Surveys, Vol. 10, No. 1, March 1973.
52. [Сгурев и др. 87]
 Sgurev, V. et al. Knowledge Acquisition and Man-Machine

Interface in the Expert System DIGS. In "Artificial Intelligence II: Methodology, Systems, Applications", pp.

53. [Слейджл 67]
Слейджл, Д. Эвристическая программа, решающая задачи символического интегрирования в объеме первого курса университета. В сб. "Вычислительные машины и системы", М., Мир, 1967, с. 139-153.
54. [Спиридонова 87]
Spiridonova, M. Some Extensions and Applications of REDUCE System. Proceedings of EUROCAL'87 (in print).
55. [Спиридонова, Джамбазова 84]
Spiridonova, M., Djambazova, M. A Knowledge Based Extension of REDUCE 2 System, Proc. of the Intern. Conf. on AIMS, Varna, 17-20 Sept., 1984, ЦИТИ, МР16.
56. [Спиридонова, Ищев, Филипов 83]
Спиридонова, М., Ищев, К., Филипов, Ф. Применение структурного подхода при аналитическом решении волновых задач. Конф. по системам для автоматизированного информационного обслуживания, Варна, 3-8 октября, 1983 (под печат).
57. [Спиридонова, Казасов 83]
Спиридонова, М., Казасов, Х. Аналитично извършване на Лапласово преобразование с помощта на ЕИМ. АСУ, 1983, № 3, с. 33-41.
58. [Спиридонова, Хаджиков, Динева 83]
Спиридонова, М., Хаджиков, Л., Динева, Л. Исследование реакции динамической модели со сосредоточенными массами при помощи системы аналитических преобразований REDUCE-

2. Конф. по системам для автоматизированного информационного обслуживания, Варна, 3-8 октября, 1983. (под печат).
59. [Стефик и гр. 82]
Stefic M. et al. The organizations of expert systems, a tutorial. Artificial Intelligence, 1982, vol. 18, pp. 135-173.
60. [Стиил 77]
Steele, G. Fast Arithmetic in MACLISP. Proc of the 1977 MACSYMA Users Conference, NASA CP-2012, 1977.
61. [Стоян 79]
Stoyan, H. The TU-LISP System for DOS and OS. Рабочее совещание по системам и методам аналитических вычислений на ЭВМ и их применении в теоретической физике. Дубна, 15-21 сент. 1979.
62. [Стоян 80]
Стоян, X. ЛИСП - ръководство по програмиране. Пр. от немски. ЕЦММ при БАН, 1980, 301 стр.
63. [Сул 60]
Soule, J. L. Application de la theorie des reseaux mailles aux problemes d'aerage minier, Annales des Mines, VI, 1960, pp. 339-356.
64. [Тейтелман 74]
Teitelman, W. INTERLISP Reference Manual, Xerox-Palo Alto Research Center, Palo Alto, Calif., 1974.
65. [Томов 75]
Томов, В. Алгоритъм и програма за определяне на база от независими контури в ориентиран свързан граф. Доклади на Националната конференция по "Проблеми на

математическото осигуряване, свързани с алгоритмичния език ФОРТРАН и неговите приложения", София, 1975, с. 64-68.

66. [Томов 75a]

Томов, В. Системен подход при създаване на математическо осигуряване за решаване на научно-технически задачи, Доклади по покана на Четвърта пролетна конференция на БМД, Перник, 1975, с. 79-86.

67. [Томов 75б]

Томов, В. Математическо осигуряване за пресмятане и анализ на потокоразпределението във вентилационни системи, Автореферат, София, 1985, 21 стр.

68. [Томов, Велчев 75]

Томов, В., Велчев И. Алгоритъм и програма за пресмятане на регулираното разпределение на въздуха във вентилационни системи, Сборник научни трудове на семинара по "Проблеми на съвременните методи и средства за управление на миннодобивната промишленост". София, 1975, с. 59-67.

69. [Томов, Геров 87]

Томов, В., Геров, А. Интеллектуални прикладни програмни системи, Первый научный семинар с международным участием "Проблемы и применения искусственного интеллекта", Варна, 21-25 септември 1987, с. 58-66.

70. [Томов, Геров 87a]

Томов, В., Геров, А. Експертни системи и приложения. В сб. "Математика и математическо образование. Доклади на XVI пролетна конференция на СМБ, Слънчев бряг, 6-9

април 1987 г.", София, Издателство на БАН, 1987 г., стр. 103-112.

71. [Томов, Геров, Григоров, Колчев 88]
 Томов, В., Геров, А., Григоров, А., Колчев, К. Предметно-ориентированные пустые интеллектуальные прикладные программные системы. Межд. конф. "Методы искусственного интеллекта и их применения", Пхенян, 1-11 октомври 1988 (пог печат).
72. [Томов, Геров, Спиридонова]
 Томов, В., Геров, А., Спиридонова, М. ЛИС Попогоди езици за програмирање. Сборник доклади на Национални семинар по информатика към СМБ, т. II (пог печат).
73. [Томов, Геров, Спиридонова 85]
 Томов, В., Геров, А., Спиридонова, М. Системи за аналитични преобразования. В сб. "Математика и математическо образование. Доклади на XIV пролетна конференция на СМБ, Слънчев дряг, 6-9 април 1985 г.", София, Издателство на БАН, 1985, стр. 92-103.
74. [Томов и др. 87]
 Томов, В. и др. Представяне и използване на знанията. В "Отчетни материали за работата по договор No 29/ 25. 03. 1987г. между Институт по математика с ИЦ и КН за периода април - декември 1987 г.", 82 стр.
75. [Томов, Сахно 85]
 Tomov, V., Sahnó, St. Experimental System for Recognition and Classification of Archaeological Ceramics. In "Artificial Intelligence: Methodology, Systems, Applications" W. Bibel, B. Petkoff (editors), North-Holland, Amsterdam, 1985, pp. 191-199.

76. [Томов, Спиридонова, Геров 84]
 Томов, V., Spiridonova, M., Gerov, A. Intelligent Computer Algebra Systems. International Conference on Artificial Intelligence - Methodology, Systems, Applications, Varna, 17-20 Sept. 1984, ЦИНТИ, МР16.
77. [Томов, Тачева, Григоров 87]
 Томов, V., Tacheva, M., Grigorov, A. ESIT - An Expert System for Analysis of Antique Historical - Geographical Texts about Ancient Thrace, In "Artificial Intelligence II: Methodology, Systems, Applications", - Ph. Jorrand, V. Sgurev (editors), North-Holland, Amsterdam, 1987, pp. 235 - 242.
78. [Томов, Шарков 88]
 Томов, В., Шарков, Г. Интеллектуальные прикладные программные системы для мембранной биофизики. Седмой семинар "Проблемы информатики и ее применения в управлении, обучении и научных исследованиях", 5-11 юни 1988, Гявечица (под печат).
79. [Уайл 78]
 While, J. A Historical Perspective on Maclisp. Seminario Introdusione Alla Manipulatione Simbolica e Algebrica, Roma 20-24 Marzo 1978, I Parte.
80. [Уанг, Сапършайн 70]
 Wang, Y.J. and Saperstein L.W. A Computer - Aided Solution of Complex Ventilation Networks, Trans. Soc. Eng. A.I.M.E. Sept. 1970.
81. [Уанг, Хартман 66]
 Уанг, И. Ж., Хартман, Х.Д. Решение с ЕИМ на тримерни руднични вентилационни мрежи с няколко вентилатора и с

естествена тяга. Университет на щата Пенсилвания, 1966
(превод от английски език).

82. [Уинстон 84]
Winston, P. H. Artificial Intelligence. Addison-Wesley, Mass., 1984.
83. [Уинстон 84a]
Winston, P. H. and Horn, B.K.P. LISP - second edition. Addison-Wesley, Mass., 1984.
84. [Файкс, Келер 85]
Fikes, R. and Kehler, T. Communications of the ACM, 28 (1985) 905.
85. [Хейс-Рот и др. 83]
Hayes-Roth, F. et al. (editors) Building Expert Systems. Addison - Wesley, New-York, 1983, 547 p.
86. [Хърн 85]
Hearn, A. REDUCE User's Manual, Version 3.2, Rand publication, April, 1985
87. [Цимерман 82]
Zimmermann, U. Biochim. Biophys. Acta, 694 (1982) 227.
88. [Шарков, Димитров 87]
Sharikov, G. and Dimitrov, D. Expert systems in membrane biophysics and engineering - Knowledge representation and reasoning. In: Artificial Intelligence and Information-Control Systems of Robots-87, Plander I. (editor), North-Holland, Amsterdam, 1987, pp. 455 - 459.
89. [Шортлиф 76]
Shortliffe E. Computer-Based Medical Consultations: MYCIN. Elsevier, New-York, 1976, 47 p.

ПРИЛОЖЕНИЯ ЗА ВНЕДРЯВАНЕ

Във връзка с хабилитационния труд са направени 10 внедрявания.

Следва списък на съответните документи и самите документи.

1. Акт за внедряване на транслятор ЛИСП 1. 6 за ЕС ЕИМ, утвърден от ръководителя на Лаборатория по информатика с ИЦ при СУ "Климент Охридски" на 18 септември 1981 г. (стр. 185).
2. Справка за внедряване на транслятор ЛИСП 1. 6 за ЕС ЕИМ, подписана от директора на Централна лаборатория по автоматизация и научно приборостроене на 27 август 1982 г. (стр. 186).
3. Акт за внедряване на система за аналитични преобразования REDUCE 2 за ДПС ЕС ЕИМ, утвърден от ръководителя на Лаборатория по информатика с ИЦ при СУ "Климент Охридски" на 28 октомври 1981 г. (стр. 187).
4. Справка за внедряване на транслятор LISP F3 (INTERLISP) за ЕС ЕИМ, подписана от директора на БСНИПИ "ИНТЕРПРОГРАМА" на 15 юли 1982 г. (стр. 188).
5. Протокол за предаване на транслятор LISP F3 за ЕС ЕИМ на Института по техническа кибернетика при Словашката академия на науките, представян от г-р В. Британяк и подписан на 21. 03. 1983 г. (стр. 189).

6. Протокол за предаване на транслятор LISP-A F3, адаптиран за машините CM-4 на Института по техническа кибернетика при Словашката академия на науките, представян от д.е.н., к.ф.-м.н. Й. Миклошко и подписан на 30.03.1983 г (стр. 190).
7. Справка за внедряване на аналитично-изчислителна система за работа с матрици в катедра "Техническа механика" при ВМЕИ-Варна, подписана от Ректора на ВМЕИ-Варна на 29.09.1986 г. (стр. 191).
8. Акт за внедряване на REDUCE 2 за ОС и на транслятор STANDARD LISP, утвърден от ръководителя на ЛИИЦ при СУ "Климент Охридски" на 20.12.1982 г. (стр. 192).
9. Служебна бележка за успешно използване на интелектуализираната приложна програмна система PREFES в секция "Биофизика на мембранните взаимодействия", подписана от директора на Централна лаборатория по биофизика при БАН на 25.05.1988 г. (стр. 193).
10. Акт за внедряване на система за аналитични преобразования SAC - 2 за ДДС ЕС ЕИМ, утвърден от ръководителя на Лаборатория по информатика с ИЦ при СУ "Климент Охридски" на 23 декември 1983 г. (стр. 194).

А К Т З А В Н Е Д Р Я В А Н Е

ОБЕКТ НА ВНЕДРЯВАНЕ: Транслатор ЛИСИ 1.6 за ДОС ЕС ЕИМ, получен от
ТУ-Дрезден.

ВНЕДРИТЕЛ: Лаборатория по информатика с ИИ при СУ "Кл.Охридски"

АВТОР: Адаптиран от колектив в състав: ст.н.с. В.Томов, н.с. Ал.Геров
и н.с. М.Спиридонова.

НАЧАЛО НА ЕКСПЛОАТАЦИЯ: Януари 1981 г.

АТЕСТАТ НА РАЗРАБОТКАТА: ЛИСИ е алгоритмичен език, предназначен за
обработка на символна информация, представена
предимно чрез списъци и низове. Той е удобен
за решаване на редица задачи, главно от облас-
тта на изкуствения интелект, аналитични преос-
разования, диалогови информационни системи,
управление на работи, доказване на теореми,
обработка на естествени езици и др.

Транслаторът може да се използва, както
в режим на интерпретация, така и в режим на
компиляция. Той съдържа над триста вградени
функции, написани на Асемблер.

ВИД НА ПРЕДАВАНЕТО: Дистрибутивна лента и документация.

София, 18 септември 1981

ПРЕДСТАВИТЕЛ НА АВТОРА:



ПРЕДСТАВИТЕЛ НА ВНЕДРИТЕЛЯ:

БЪЛГАРСКА
АКАДЕМИЯ НА НАУКИТЕ
ЦЕНТРАЛНА ЛАБОРАТОРИЯ ПО
АВТОМАТИЗАЦИЯ И НАУЧНО
ПРИБОРОСТРОЕНЕ

Тел. 73-41
бул. Ленин № 72

№ 1002

186

София, 27.08.1982г.

До ЕЦММ при БАН
Т У К

СПРАВКА ЗА ВНЕДРЯВАНЕ

Транслаторът LISP 1.6 за ДОС ЕС ЕЦММ е внедрен в ЦИАНИ - ЕЦ по физика от колектив в състав ст.н.с. В. Томов, н.с. Ал. Геров и н.с. М. Спиридовона, съвместно с наши специалисти и се използва за научно изследователска дейност.

Директор на ЦИАНИ:
(ст.н.с. Л. Антонов)



А К Т З А В Н Е Д Р Я В А Н Е

ОБЕКТ НА ВНЕДРЯВАНЕ: Система за аналитични преобразования *REDUCE 2* получена от Техническият университет в Дрезден

ВНЕДРИТЕЛ: Лаборатория по информатика с Изчислителен център при СУ "Кл.Охридски"

АВТОР: Системата е адаптирана от колектив: ст.н.с. В.Томов, н.с. М. Спиридонова, н.с. Ал.Геров

НАЧАЛО НА ЕКСПЛОАТАЦИЯТА: Май 1981 г.

АТЕСТАТ НА РАЗРАБОТКАТА: Системата за аналитични преобразования *REDUCE 2* е предназначена за формално манипулиране с математически изрази. Нейните възможности включват: развитие и приведение на полиноми и рационални функции; намиране на най-голям общ делител на 2 полинома; аналитично диференциране; субституции; автоматично и контролирано от потребителя опростяване на изрази; аналитични пресмятания с матрици; специални средства за приложение във физиката на високите енергии. Езикът на системата е алголоподобен. Той е удобен за описание на аналитични преобразования и е лесен за изучаване. Допустими са 2 режима на работа със системата – алгебричен и символен, последният от които позволява пълно използване на езика ЛИСП (в случая – ЛИСП 1.6, на който е програмирана системата). *REDUCE 2* работи под управление на операционната система ДОС/ЕС и може да се използва както в пакетен, така и в диалогов режим.

ВИД НА ПРЕДАВАНЕТО: Дистрибутивна лента и документация.

София, 28.10.1981 г.

ПРЕДСТАВИТЕЛ НА АВТОРА:

(Своеручен подпис)

ПРЕДСТАВИТЕЛ НА ВНЕДРИТЕЛЯ:

(Своеручен подпис)

Советско-Болгарский научно-
исследовательский и проектный
институт „ИНТЕРПРОГРАММА“
СОФИЯ
Бульвар Ал. Стамболийского 62/64
Телефон 8 41 81

Българо-съветски научно-
изследователски и проектантски
институт „ИНТЕРПРОГРАМА“
СОФИЯ
Булевард Ал. Стамболийски 62/64
Телефон 8 41 81

Исх. № 4-1056 от 15.07.82г.

на Ваш № от

ДО
ЕЦММ ПРИ БАН
ТУК

СПРАВКА ЗА ВНЕДРЯВАНЕ

Транслаторът LISP F3 (INTERLISP) за ЕС ЕИМ е внедрен в БСНИПИ "Интерпрограма" от колектив в състав ст.н.с. В.Томов, н.с. Ал.Геров и н.с. С.Върбанов, съвместно с наши специалисти и се използва за научноизследователски дейности.

ЗАМ.ДИРЕКТОР:



/В.С. СПИРИДОНОВ/

БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ

ИНСТИТУТ ПО МАТЕМАТИКА
С ИЗЧИСЛИТЕЛЕН ЦЕНТЪР1113 София, ул. „Акад. Г. Бончев“, бл. 8
или 1090 София, П. К. 373
тел. 71-31

№ 19

София, 21.05.1983

ПРОТОКОЛ

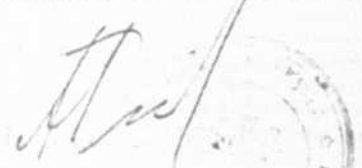
обмена программными системами

21 мая 1983 года сотрудники Института технической кибернетики Словацкой академии наук и Института математики с ВЦ Болгарской академии наук обменялись следующими программными системами:

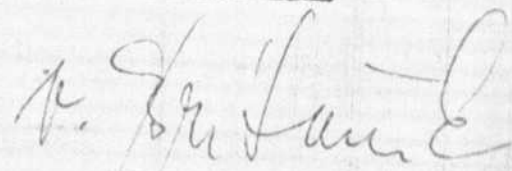
Институт математики с ВЦ БАН, представленный А. Геровым, передал систему ЛИСИ ФЗ для ЕС ЭВМ.

Институт технической кибернетики САН, представленный В. Бритањком, передал текст программ системы ЛИСИ 1.10.

Системы были выменены на основе сотрудничества рабочей группы РТ-18 в рамках КНВВТ Академий наук социалистических стран.

За ИМ с ВЦ БАН:


д-р Ал. Геров,
научный сотрудник.

За ИТК САН:


д-р В. Бритањк
научный сотрудник.

SLOVENSKÁ AKADEMIA VIED BRATISLAVA
ÚSTAV TECHNICKÉJ KYBERNETIKY

190

Братислава, 30.3.83

ПРОТОКОЛ

обмена программными системами

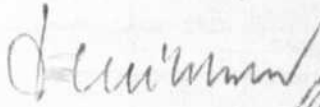
30 марта 1983 года сотрудники Института технической кибер-
нетики Словацкой академии наук и Института математики с ВЦ Бол-
гарской академии наук обменялись следующими программными систе-
мами:

ИТК САН, представленный д-ром Р.Фибби, передал Институту
математики с ВЦ БАН систему LISP 1.10 для использования
на PDP-11/4 подоперационной системы RSX-11/M V-4.0

Институт математики с ВЦ БАН, представленный Г.Ангеловой,
передал систему LISP-A F 3 для ЭВМ SM-4.

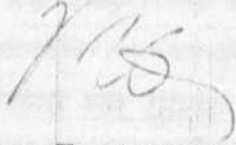
Системы были выменены на основе сотрудничества рабочей
группы РГ-18 в рамках КНВВТ Академий наук социалистических стран

За ИТК САН:


д.е.н. Я. МИКЛОШКО,
к.ф.-м.н.
зав. отделением



За ИМ с ВЦ БАН:


д-р Г. АНГЕЛОВА,
к.ф.-м.н.
научный сотрудник

МАШИННО ЕЛЕКТРОТЕХНИЧЕСКИ ИНСТИТУТ



Телефони:

Номератор 88-01-61 до 65

Телекс 77401

2757-КД
Варна № 1 / 198 г.

29.09.86

ДО

ЕИММ - Б А Н

С О Ф И Я

ул.Акад.Г.Бончев № 8

Уведомяваме Ви, че аналитично-изчислителната система за работа с матрици, предадена ни безвъзмездно от сектор "Изкуствен интелект", БАН, където е разработена, през 1985 год. беше успешно генерирана и частично експериментирана в катедра "Техническа механика" при ВМЕИ-Варна.

Запознаването и работата със системата позволяват да се заключи, че тя дава възможност с помощта на компютър да се извършват обемисти и труднопоглъщащи аналитични преобразования на обекти от линейната алгебра (матрици и вектори) и осигурява използването на числени процедури, написани на ФОРТРАН, предназначени за работа с матрици и вектори. Особено внимание е обърнато на средствата за аналитично намиране на обратна матрица и точно решаване на линейна задача и намирането на собствени вектори. Несъмнено, използването на системата ще даде възможност да се повиши ефективността на научно-изследователската и научно-приложната дейност в редица области от науката и техниката. Също така тя може да бъде използвана за селективно обучение на висококвалифицирани кадри както в рамките на тристепенното редовно обучение, така и в системата на следдипломната квалификация.

Изразяваме готовност за сътрудничеството в областта на изкуствения интелект да продължи и занапред.

РЕКТОР на ВМЕИ-ВАРНА:

/проф. д-н. инж. Д. Дончев/



А К Т ЗА В Н Е Д Р Я В А Н Е

ОБЕКТ НА ВНЕДРЯВАНЕТО: Система за аналитични преобразования REDUCE
вариант от 1979 г. за ОС/ЕС, получен от ОИЯИ - Дубна

ВНЕДРИТЕЛ: Лаборатория по информатика с изчислителен център при СУ
"Кл.Охридски"

АВТОР: Системата е усвоена от колектив: ст.н.с. В.Томов - ръководите
н.с. М.Спиридонова, асп. Здр.Василев

НАЧАЛО НА ЕКСПЛОАТАЦИЯТА: март, 1982 г.

АТЕСТАТ НА РАЗРАБОТКАТА: Системата за аналитични преобразования

REDUCE-2 е предназначена за формално манипулиране с математи-
чески изрази. Нейните възможности включват: развитие и приведе-
ние на полиноми; действия с полиноми и рационални функции; на-
миране на най-голям общ делител на 2 полинома; аналитично дифе-
ренциране; субституции; автоматично и контролирано от потребител
ля опростяване на изрази; аналитични пресмятания с матрици; спе-
циални средства за приложение във физиката на високите енергии
Езикът на системата е алголоподобен. Той е удобен за описание
на аналитични преобразования и е лесен за изучаване. Допустими
са 2 режима на работа със системата - алгебричен и символен,
последният от които позволява пълно използване на езика ЛИСП
(в случая - STANDARD LISP, на който е програмирана системата
Вариантът на REDUCE-2 за ОС/ЕС има някои допълнителни въз-
можности и по-добри експлоатационни качества в сравнение с ва-
рианта за ДОС/ЕС.

С REDUCE-2 може да се работи както в пакетен, така и в диалог
режим.

БИД НА ПРЕДАВАНЕТА: Дистрибутивна лента и документация.

София, 20.12.1982 г.

ПРЕДСТАВИТЕЛ НА АВТОРА:


/ст.н.с. В.Томов/

ПРЕДСТАВИТЕЛ НА ВНЕДРИТЕЛЯ


/доц. Ст.Бъчваров/

193

БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ЦЕНТРАЛНА ЛАБОРАТОРИЯ ПО БИОФИЗИКА

1113 — София, ул. Акад. Г. Бончев, бл. 6
тел. 72-86-10

София, 25.05 1978 г.

СЛУЖЕБНА БЕЛЕЖКА

Настоящата се издава от Централна лаборатория по биофизика - БАН уверение на това, че разработената съвместно със специалисти от Института по математика с ИЦ при БАН под научното ръководство на ст. н. с. кмн Валентин Томов интелектуализирана приложна програмна система PREFES се използва успешно в научно-изследователската работа на секция "Биофизика на мембранните взаимодействия".



ДИРЕКТОР:

A handwritten signature in black ink, appearing to be "Ал. Гидиков".

(проф. Ал. Гидиков)

А К Т ЗА В Н Е Д Р Я В А Н Е

ОБЕКТ НА ВНЕДРЯВАНЕ: Система за аналитични преобразования SAC - 2,
получена от Университета в Карлсруе, ФРГ.

ВНЕДРИТЕЛ: Лаборатория по информатика с Изчислителен център при СУ
"Кл.Охридски".

АВТОР: Системата е адаптирана от колектив: ст.н.с. В.Томов, н.с. Ал.Гергел,
н.с. С.Върбанов, асп. М.Нишева.

НАЧАЛО НА ЕКСПЛОАТАЦИЯТА: Февруари 1982 г.

АТЕСТАТ НА РАЗРАБОТКАТА: Системата за аналитични преобразования SAC - 2
е предназначена за формално манипулиране с алгебрични изрази. Нейните
възможности включват: операции над произволно големи цели и
рационални числа и над реални и комплексни числа с единична точност;
елементарни операции над полиноми на няколко променливи с цели и
рационални коефициенти: вход, изход, аритметични операции, субсти-
туция и оценяване, диференциране и интегриране, интерполация, гене-
риране на случайни полиноми, различни хомоморфизми и изоморфизми,
превръщания на представянето и др.; намиране на най-голям общ дели-
тел и резултанта на полиноми; факторизация на полиноми; изолиране
на реални корени на полиноми. Реализирани са и някои алгоритми за
модулна аритметика, за извършване на преобразования от линейната
алгебра и за работа с алгебрични числа. Езикът на системата е алго-
лоподобен. Той е лесен за изучаване и използване. SAC - 2 работи
под управлението на операционната система ДОС/ЕС и може да се
използува в пакетен режим.

ВИД НА ПРЕДАВАНЕТО: Дистрибутивна лента и документация.

София, 23.XII.1983 г.

ПРЕДСТАВИТЕЛ НА АВТОРА:



ПРЕДСТАВИТЕЛ НА ВНЕДРИТЕЛЯ:

С Ъ Д Ъ Р Ж А Н И Е

ХАРАКТЕРИСТИКА НА ХАБИЛИТАЦИОННИЯ ТРУД.	1
УВОД.	9
БЛАГОДАРНОСТИ	14
ГЛАВА I. Основни концепции на ИППС.	17
1. Определение и обща структура на ИППС.	17
2. Основни видове ИППС	21
3. Предметно-ориентирани празни ИППС. Проблемно-ориентирани ИППС.	32
ГЛАВА II. Представяне и обработка на знанията	35
1. Общи сведения	35
2. Три типа знания	39
2.1. Знания за обекти и факти.	39
2.2. Знания за връзките между обектите и фактите	40
2.3. Метазнания.	41
3. Представяне на знанията.	42
3.1. Семантични елементи	42
3.2. Връзки между семантичните елементи.	43
4. Два основни типа представяне на знания.	44
5. Основни схеми за представяне на знанията.	47

5.1. Предикатна логика от I ред.	47
5.2. Представяне чрез процедури.	47
5.3. Семантични мрежи.	48
5.4. Системи продукции	49
5.5. Фреймове.	49
ГЛАВА III. Инструментални средства.	51
1. Лиспоподобни езици за програмиране.	51
1.1. Описание на езика ЛИСП.	51
1.2. Стил на програмиране.	54
1.3. Версии на езика ЛИСП.	58
1.4. Транслатори от ЛИСП, внедрени у нас.	60
2. Други езици и системи	63
ГЛАВА IV. Експертни системи като ИППС	70
1. ЕС и приложения – общи сведения	70
1.1. Основни характеристики на ЕС.	71
1.2. Област на приложение.	72
1.3. Класификация на ЕС.	73
1.4. Обща структура на ЕС.	76
1.5. Развитие на ЕС.	84
2. Интелектуализирани приложни програмни системи за мембранната биофизика	87
2.1. Основна концепция	87
2.2. ЕС за получаване на липозоми и електросливане на протопласти.	90
2.2.1. Естество на проблема и необходимост от съвет.	91
2.2.2. Експертните системи PREFES и LIPES.	93
3. ЕСОК – предметно ориентирана празна ЕС за оценка	

на кадри.100
ГЛАВА V. Интелектуализирани въпросно-ответни системи (ИВОС).	
1. Предметно-ориентирана празна ИВОС за анализ на историко-географски текстове.107
2. Проблемно-ориентирана ИВОС за географски и логически анализ на историко-географски текстове за гревна Тракия110
3. Представяне на историческите знания в системата ЕСИТ.113
4. Действие на логическия и изпълнителен процесор на системата ЕСИТ119
ГЛАВА VI. Интелектуализирани аналитично-изчислителни системи (ИАИС).	
1. Системи за аналитични преобразования.123
1.1. Основни сведения за някои САП126
1.2. Специализирани САП.128
2. Развитие на САП132
2.1. Изисквания към съвременните САП132
2.2. Интелектуализирани аналитично-изчислителни системи.134
2.2.1. Връзка с потребителя.137
2.2.2. Планираща част.138
2.2.3. Изпълнителна част139
2.2.4. Планиране на действията и представяне на знанията в ИАИС за работа с матриц.142
3. Общи заключителни бележки145

ГЛАВА VII. Интелектуализирани приложни програмни системи за научно-технически задачи (ИППС за НТЗ)	147
1. Системен подход при създаване на математическо осигуряване за научно-технически задачи	147
1.1. Характерни особености на научно-техническите задачи във връзка с решаването им на ЕИМ.	148
1.2. Основни елементи на математическото осигуряване за научно-технически задачи	149
1.3. Същност и възможности на системния подход при създаване на математическо осигуряване за науч- но-технически задачи.	150
2. Интелектуализация на приложни програмни системи за научно-технически задачи	153
3. Приложна програмна система за вариантен анализ на вентилационни системи и нейното интелектуали- зиране.	154
3.1. Пресмятане на потокоразпределението във венти- лационни системи.	155
3.2. Програмна система за вариантен анализ на венти- лационни системи.	157
3.3. Общ проект за интелектуализиране на приложната програмна система за вариантен анализ на венти- лационни системи.	163
СПИСЪК НА СЪКРАЩЕНИЯТА.	168
ЛИТЕРАТУРА.	168

ПРИЛОЖЕНИЯ ЗА ВНЕДРЯВАНЕ.183
СЪДЪРЖАНИЕ.195