

*Изкуствен интелект - зимен семестър, 2007/2008 учебна година*

***Лекция 8:  
Представяне и използване на знания***

# ***Работа със знания в системите с изкуствен интелект. Данни и знания. Архитектура на системите, основани на знания***

## ***Обща постановка***

Работата със знания е една от съществените отличителни черти на програмните системи с ИИ. Според едно от популярните определения *интелектът е способност за формулиране, натрупване и използване на знания* (Intelligence is applied knowledge). Терминът “знания” (“знание”) в системите с ИИ се използва за означаване на кодирания опит на агентите. Опитът е източникът на знания (информация) за решаването на задачи. Чрез определението “кодирани” се означава обстоятелството, че знанията са формулирани, записани и готови за използване.

Съществена за системите с ИИ е възможността за преход от работа с данни към работа със знания. Традиционните програмни системи работят с информация, организирана във вид на бази от данни (БД), докато за системите с ИИ и по-точно за т. нар. *системи, основани на знания* (СОЗ; Knowledge-based Systems, KBS) е характерна работата с *бази от знания* (БЗ). Има фундаментална разлика между възможностите, които предоставят тези два типа системи за съхранение и предоставяне на информация.

При БД е възможно извличане само на такава информация, която е представена в явен вид в базата. При БЗ е възможно извършването на разсъждения (извод), в резултат на което *може да се генерира нова информация, която не присъства в явен вид в базата*. В този смисъл по принцип БД не могат да представят и СУБД не могат да обработват непълна информация, докато за БЗ такова ограничение не съществува.

## ***Същност на знанието***

В общ енциклопедичен план знанието обикновено се определя като система от съждения с принципна и единна организация, основана на обективни закономерности.

В обективен смисъл (т.е. като резултат от определен познавателен процес) знанието се противопоставя на заблуждението.

В субективен смисъл знанието представлява мнение, вяра, убеждение (по реални причини) в истинността на наблюдението.

В СОЗ знанията представляват *съвкупност от твърдения, представящи мнението, вярата и убежденията на т. нар. когнитивен агент.*

Различия между данни и знания в СИИ:

- В степента на общност (и постоянност/изменяемост);
- По начина на използване;
- Наличие на класификационно-йерархична структура на знанията (основана на родово-видови отношения от тип обект – клас, клас – суперклас, тип – подтип, ситуация – подситуация и т.н.).

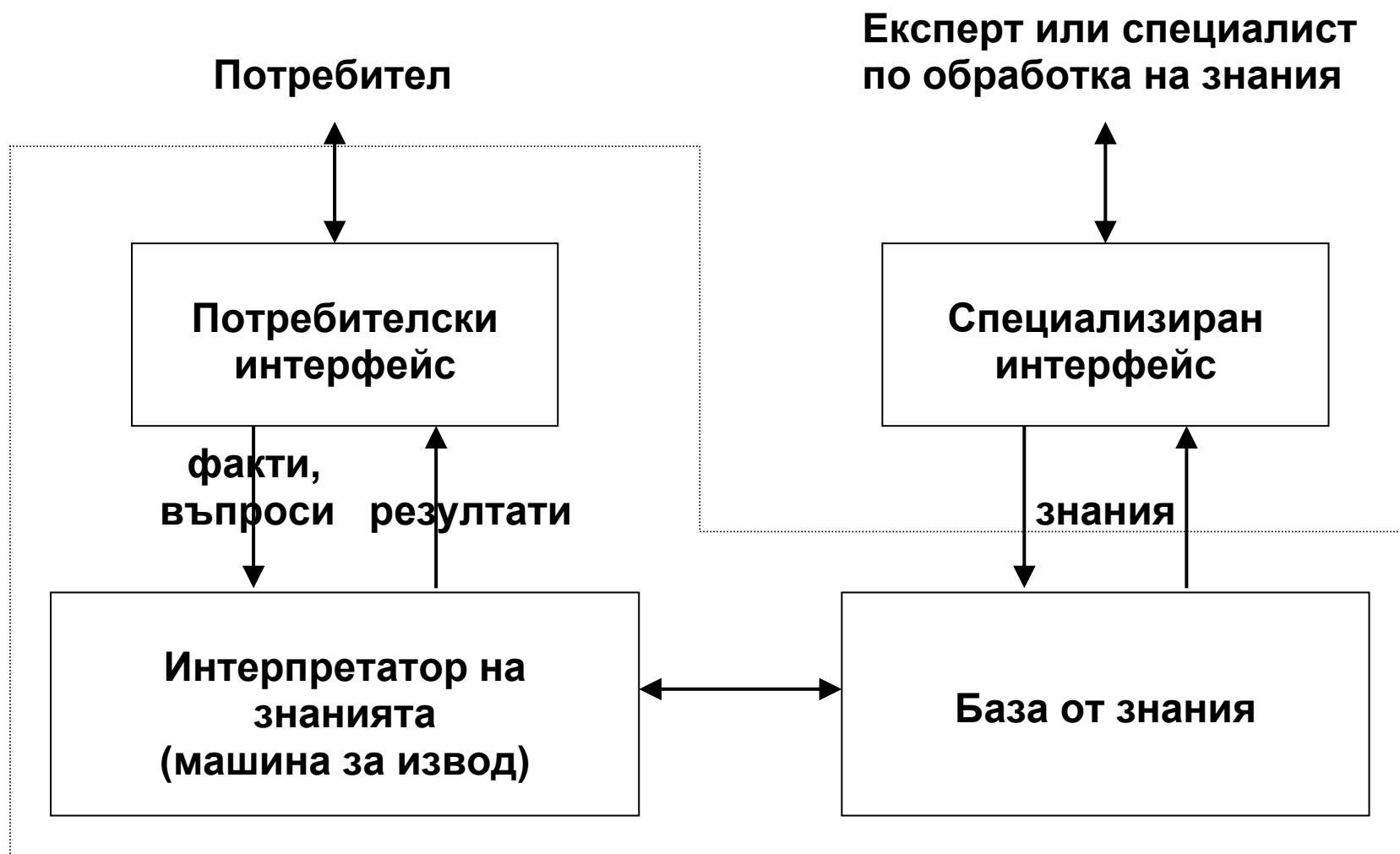
Няма рязка граница между данни и знания в СИИ.

## ***Архитектура на СОЗ***

Основава се на приемане (символен подход) или отхвърляне (конекционистки подход) на т. нар. хипотеза за представянето на знанията (knowledge representation hypothesis).

При символния подход:

- база от знания (knowledge base)
- машина за извод (inference engine) или интерпретатор на знанията (knowledge interpreter)



Архитектура на CO3 (символен подход)



### Типове знания в СОЗ:

- знания за обектите и фактите в предметната област (фактологически знания)
  - знания за връзките (релациите) между обектите и фактите
  - метазнания
- } **знания за предметната област**

Видове метазнания: стратегически знания, поддържащи знания и др.

***Изисквания към формализмите за представяне и използване на знания (ПИЗ):***

- ясна семантика (значението на правилно построените изрази да бъде добре определено, т.е. всеки правилно построен израз да може да се интерпретира еднозначно);
- коректност на правилата за извод (правилата за извод да бъдат такива, че ако явно зададените знания в БЗ са верни, то всички извлечени от тях знания също да са верни);
- естественост на представянето (по отношение на терминологията и структурата на знанията за съответната предметна област);
- модулност на представянето;
- ефективност (по отношение на изискванията за памет и време).

## ***Основни типове формализми за ПИЗ***

Съществуват два основни типа формализми за ПИЗ: формализми от декларативен тип (декларативни формализми – *какво?*) и формализми от процедурен тип (процедурни формализми – *как?*). При декларативните формализми основната тежест пада върху представянето на знанията; при процедурните формализми съществен е начинът на използване на знанията. При декларативните формализми знанията се представят явно, в декларативен формат, а при процедурните формализми знанията се съдържат в процедурите на някаква програма. Процедурните формализми дават възможност за по-голяма ефективност при използването на знанията, но при тях измененията в БЗ стават по-трудно, отколкото при декларативните.

## ***Представяне и използване на знания чрез процедури***

### ***Обща характеристика***

Типичен процедурен формализъм. Процедурите са обособени като формализъм за ПИЗ, защото всяка програма “носи” в себе си своите знания (част от знанията на своя автор) и в този смисъл те са представени чрез нея.

### ***Идея на представянето***

Най-често знанията се представят чрез системи от малки по обем процедури – демони, в които са кодирани знания за определени характерни ситуации и всяка от които се активира само при настъпване на съответната ситуация. За разлика от стандартните програмни системи, за които е характерна йерархична структура, за базите от процедури (демони) е типична хетерархична структура.

### ***Характерни области на приложение***

При управление на работи, управление на процеси, протичащи в реално време, и др. (в области, в които преобладават алгоритмични знания).

## ***Представяне и използване на знания чрез средства на математическата логика***

### ***Обща характеристика***

Математическата логика може да се разглежда като един от най-рано създадените формализми за ПИЗ, който има ясно изразен декларативен характер. Всяка логическа система е формален език, с помощта на който могат да се изразяват различни твърдения (знания) и да се извеждат нови твърдения с помощта на съответни правила за извод.

По-точно, представянето на знания в термините на дадена формална логическа система се извършва с помощта на правилно построени формули в тази система, а използването на знанията се осъществява посредством валидните за логическата система правила и методи за извод.

Най-често използваната формална логическа система за целите на ПИЗ е предикатното смятане от първи ред.

***Характерни особености на предикатното смятане от първи ред като формализъм за ПИЗ:***

- полуразрешимост
- монотонност (невъзможност за работа със знания, верни по подразбиране)

Некласически логики, които се използват като формализми за ПИЗ: немонотонни логики, модални логики, размита логика и др.



### ***Оценка на логическите системи като средства за ПИЗ***

В сила са всички общи предимства и недостатъци на декларативните формализми за ПИЗ. Най-съществени специфични предимства на логическите системи: ясна семантика, голяма изразителна сила.

## ***Представяне и използване на знания чрез системи от продукционни правила***

### ***Обща характеристика***

Декларативен в основата си формализъм с елементи на процедурност на по-ниско ниво. Негова основна характеристика е декомпозирането на знанията на малки части (правила) от типа условие – следствие (ситуация – действие).

### ***Архитектура на системите, основани на правила:***

- работна памет (контекст)
- база от правила
- интерпретатор на правилата

## ***Работна памет***

Съдържа данните за конкретно решаваната задача, които са установени (известни) към текущия момент. Произход на данните в работната памет:

- от потребителя (зададени по условие при дефинирането на задачата или получени като отговор на уточняващ въпрос, зададен от интерпретатора на правилата в процеса на обратен извод);
- от интерпретатора на правилата (получени в процеса на логическия извод).

### ***База от правила***

Съхранява знанията за предметната област, представени под формата на импликации от вида

(<име\_на\_правило> <лява\_страна> <дясна\_страна>)

Лявата страна на правилото се нарича негово *условие* (*предпоставка*), а дясната страна на правилото – негово *следствие* (*действие*). Най-често лявата страна на правилото представлява конюнкция от т. нар. елементи на условието. Дясната страна обикновено представлява поредица от елементи, описващи действия, които следва да се изпълнят (най-често това са твърдения - следствия, които се записват в работната памет в резултат на изпълнението на съответното правило). Структурата на елементите на условията и елементите на десните страни на правилата е сходна с тази на елементите на работната памет.

## ***Интерпретатор на правилата***

Програмна система, чието основно предназначение е да приложи описаните чрез правилата знания върху данните за конкретната задача.

Процесът на работа на интерпретатора на правилата е цикличен, като всяка стъпка от работния цикъл на интерпретатора се разделя на две фази: *избор на правило* и *изпълнение на избраното правило*.

При избора на правило, което да се изпълни на текущата стъпка, се извършват множество операции на *съпоставяне по образец*.

Стратегии на работа на интерпретатора на правилата:

- *прав извод* (forward chaining) или *извод, управляван от данните* (data-driven inference). Пример: OPS5;
- *обратен извод* (backward chaining) или *извод, управляван от целите* (goal-driven inference). Пример: Пролог.

*Конфликтно множество*: множеството от правила, чиито леви страни се удовлетворяват от съдържанието на работната памет на текущата стъпка от работния цикъл на интерпретатора.

*Стратегии за разрешаване на конфликтите*: стратегии за избор на правило от конфликтното множество (избор на първото възможно правило, което не предизвиква зацикляне; избор на най-актуалното правило; избор на правилото с най-сложното/най-простото условие и др.).

## ***Обща оценка на правилата като формализъм за ПИЗ***

Специфични преимущества:

- естественост на представянето на експертни знания;
- модулност на базата от знания;
- възможност за обяснение на резултатите от извода.

### Недостатъци и проблеми:

- проблеми при генериране на съдържателни обяснения на взетите решения (необходимост от използване на добре структурирана база от знания, която освен знанията за предметната област, представени чрез правила, включва и различни типове метазнания, подпомагащи генерирането на обяснения);
- недостатъчна изразителна сила;
- потенциална неефективност на работата на интерпретатора на правилата (решения: алгоритъм RETE, архитектури от тип “черна дъска”).



## ***Експертни системи***

### ***Основни характеристики на експертните системи***

*Експертната система* (ЕС) е компютърна програма, която съдържа знания и извършва логически извод относно специализирана предметна област с цел решаване на определени задачи или даване на подходящи съвети. По обхвата, обема, съдържанието, структурата и организацията си знанията на една ЕС са сравними с тези на хората – експерти в съответната предметна област.

Обикновено ЕС се използват за решаване на следните типове задачи:

- Интерпретация на данни (например звукови сигнали);
- Диагностика на повреди или заболявания;
- Структурен анализ на сложни обекти (например химични съединения);
- Конфигурация на сложни обекти (например компютърни системи);
- Планиране на последователности от действия.

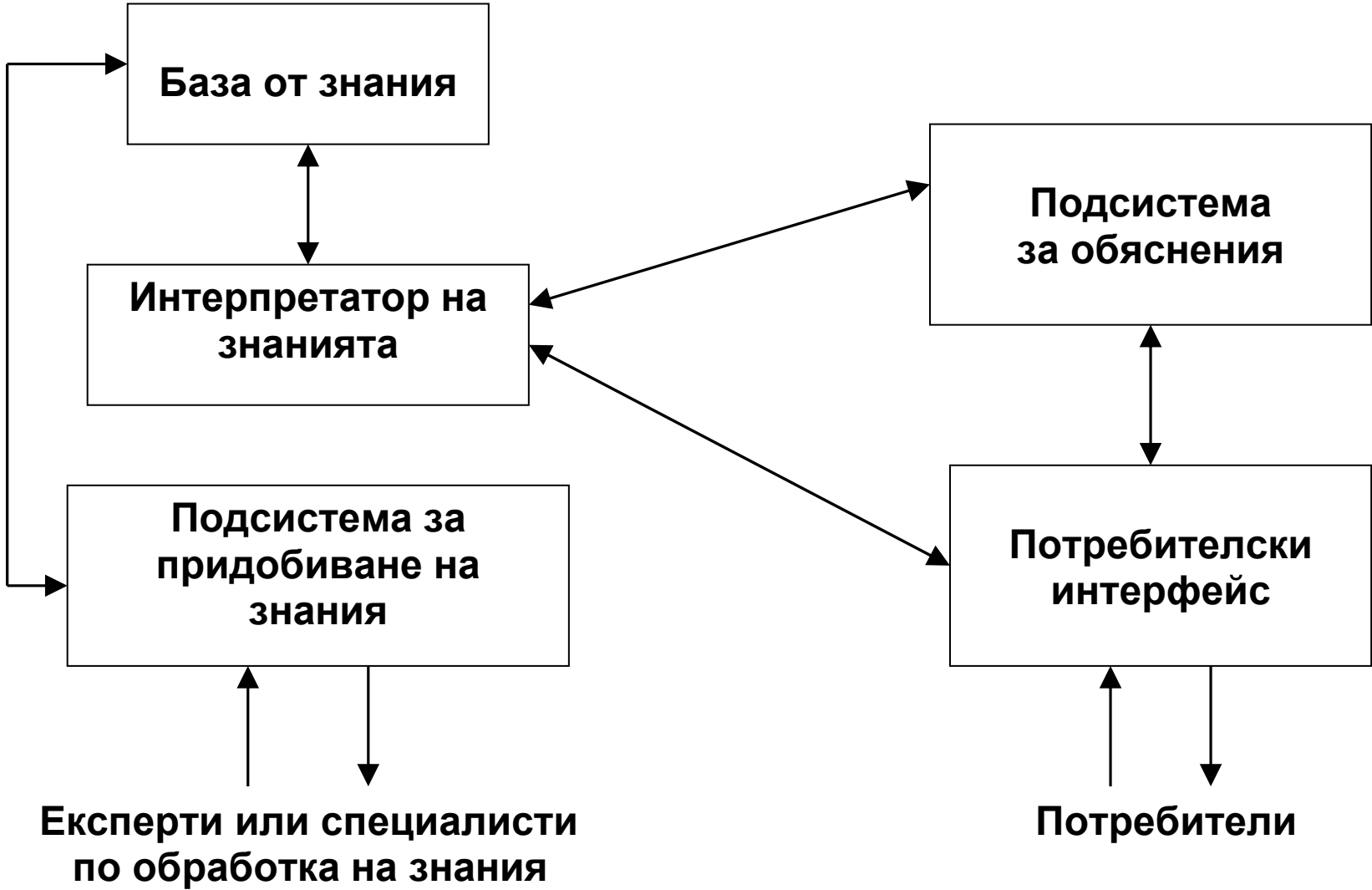
ЕС имат някои *особености*, които ги отличават от традиционните програмни системи:

- ЕС моделират начина, по който хората правят изводи в определена предметна област, а не самата област;
- ЕС извършват извод на базата на някакво представяне на човешки знания;
- ЕС обикновено решават задачи с помощта на евристични или приблизителни методи.

*Евристиките* са правила, извлечени от опита, които кодират определени знания за начина за решаване на даден проблем от определена област. *Евристичните методи* са приблизителни в смисъл, че не изискват точни данни и решенията могат да бъдат изведени от системата с определена степен на сигурност.

## ***Архитектура на ЕС***

ЕС съдържат традиционните за всяка СОЗ компоненти, към които са добавени още т. нар. *подсистема за обяснения* и *подсистема за придобиване на знания*.



***Инструментални средства за създаване на ЕС:***

- редактори на знания и среди за придобиване на знания;
- празни ЕС (ядра на ЕС, ES shells) и среди за представяне на знания.