

Expertní systémy (ES)

<http://www.uai.fme.vutbr.cz/~jdvorak/Opory/ExpertniSystemy.pdf>

Definice ES (Feigenbaum): expertní systémy jsou počítačové programy, simulující rozhodovací činnost experta při řešení složitých úloh a využívající vhodně zakódovaných, explicitně vyjádřených znalostí, převzatých od experta, s cílem dosáhnout ve zvolené problémové oblasti kvality rozhodování na úrovni experta.

Charakteristické rysy ES:

- oddělení znalostí a mechanismu jejich využívání (tím se expertní systémy odlišují od klasických programů),
- (rozhodování za neurčitosti),
- (schopnost vysvětlování).

Typy ES:

- **problémově orientovaný ES:** báze znalostí obsahuje znalosti z určité domény,
- **prázdný ES (shell):** báze znalostí je prázdná,
- **diagnostický ES:** jeho úkolem je určit, která hypotéza z předem definované konečné množiny cílových hypotéz nejlépe koresponduje s daty týkajícími se daného konkrétního případu,
- **plánovací ES:** obvykle řeší takové úlohy, kdy je znám cíl řešení a počáteční stav a je třeba s využitím dat o konkrétním řešení případu nalézt posloupnost kroků, kterými lze cíle dosáhnout.

Nástroje pro tvorbu expertních systémů:

- prázdné expertní systémy: EXSYS, FLEX, G2, HUGIN, M4, ...
- speciální programová prostředí: CLIPS, OPS5, Lisp, Prolog, ...
- obecná programová prostředí: Pascal, Delphi, C, C++Builder, ...

Aplikace ES

Aby mělo smysl použít ES pro řešení nějakého problému, musejí být splněny dvě podmínky:

1. Musí se jednat o problém složitý rozsahem nebo neurčitostí vztahů, pro nějž exaktní metoda řešení buď není k dispozici, nebo není schopna poskytnout řešení v požadované době.
2. Efekty plynoucí z použití expertního systému musejí převyšovat vynaložené náklady. To znamená, že by mělo jít o problém s opakovanou potřebou řešení a značnými finančními dopady, pro nějž lidské experti jsou drazí nebo omezeně dostupní.

Výhody ES:

- schopnost řešit složité problémy,
- dostupnost expertíz a snížené náklady na jejich provedení,
- trvalost a opakovatelnost expertízy,
- trénovací nástroj pro začátečníky,
- uchování znalostí odborníků odcházejících z organizace.

Nevýhody ES:

- nebezpečí selhání ve změněných podmínkách,
- neschopnost poznat meze své použitelnosti.

Historie vývoje ES

Poté, co při řešení praktických problémů selhaly obecné metody řešení, byla pochopena nutnost využívat specifické (expertní) znalosti z příslušné problémové domény.

Etapy vývoje:

1965–1970	počáteční fáze (Dendral)
1970–1975	výzkumné prototypy (MYCIN, PROSPECTOR, HEARSAY II)
1975–1980	experimentální nasazování
1981–	komerčně dostupné systémy

Reprezentace znalostí v pravidlových ES

Znalosti v pravidlových systémech jsou reprezentovány pomocí pravidel, která mohou mít například takovéto tvary:

IF předpoklad THEN závěr

IF situace THEN akce

IF podmínka THEN závěr AND akce

IF podmínka THEN důsledek1 ELSE důsledek2

Příklady pravidel:

IF auto_startuje = ne AND světla_svítlí = ne THEN diagnóza = vybitá_baterie

IF barometr = stoupá THEN možnost_deště = nízká AND možnost_slunečna = vysoká

Pravidlo může také obsahovat neurčitosti. Např. v pravidle

IF výška_osoby(X) = velká THEN hmotnost_osoby(X) = velká.

Životní cyklus ES:

1. Analýza problému.
2. Specifikace požadavků.
3. Předběžný návrh.
4. Počáteční (rychlé) prototypování a vyhodnocování.
5. Konečný návrh.
6. Implementace (získávání a reprezentace znalostí).
7. Validace a verifikace (testování).
8. Změny návrhu.
9. Údržba.

Etapy 6, 7 a 8 se iteračně opakují pro jednotlivé části (subsystémy) expertního systému.

Expertní systémy (ES)

<http://www.ikaros.cz/uvod-do-problematiky-expertnich-systemu>

Co je ES?

Pojem *expertní systém* byl poprvé použit na přelomu 70. a 80. let 20. století. Přesná definice tohoto pojmu nebyla dosud stanovena. Expertní systém bývá charakterizován jako *počítačový program, který simuluje rozhodovací činnost expertů při řešení složitých, úzce problémově zaměřených úloh*. Jedná se tedy o systém, který nevyužívá znalostí nabytých vlastní činností, ale využívá znalostí (myšlení a rozhodování) špičkových odborníků v dané oblasti. Avšak bez lidských omylů. Cílem činnosti expertního systému je dosáhnout co nejlepší odezvy na reálná data.

Představa, jak by měly být expertní systémy konstruované a využitelné, se v průběhu času mění. Zejména v 80. letech se usilovalo o vytvoření několika opakovatelně a široce využitelných, problémově nezávislých expertních systémů se silným stupněm všeobecně přijaté standardizace. Časem se však ukázalo, že každá aplikační oblast a každá kategorie úloh vyžadují odlišnou reprezentaci znalostí i odlišný řídicí mechanismus, a tak většina dnešních expertních systémů má charakter speciálních, problémově orientovaných systémů.

Třemi základními částmi ES je **báze znalostí, řídicí mechanismus** a **báze dat**.

Báze znalostí

Veškeré znalosti experta, které jsou potřebné k řešení daného problému, jsou soustředěné v bázi znalostí. Tato báze je koncepčně podobná databázi.

V bázi znalostí je zapsáno velké množství různých znalostí – od nejobecnějších k úzce odborným. Specialitou jsou **soukromé znalosti** (též označovány jako heuristiky či nejisté znalosti). Jde o exaktně nedokázané znalosti, které expert získává postupně v průběhu praxe a o nichž ví, že mu pomáhají při řešení určitých problémů. Tyto nejisté znalosti však nezaručují nalezení správného řešení. Heuristiky odlišují znalosti experta od znalostí průměrného pracovníka (resp. laika).

Ideální požadavky na bázi znalostí:

Vzhledem k tomu, že se znalosti experta stále (a rychle) vyvíjejí a rozrůstají, je nezbytné, aby báze znalostí byla vysoce **modulární** (tj. aby bylo možné poznatky v bázi znalostí kdykoliv doplnit anebo aktualizovat, tak, aby báze stále odpovídala úrovni nejnovějších poznatků v dané oblasti).

Expertní systém musí být schopen **využívat soukromé (nejisté) znalosti**, stejně jako člověk-expert. Existence těchto nepodložených, zkrácených či neúplných informací v bázi znalostí se označuje jako **nejistota v bázi znalostí**. Každému jednotlivému nejistému elementu reprezentace znalostí jsou přiřazovány numerické parametry (váhy, míry, stupně důvěry apod.), které vyjadřují míru jejich nejistoty.

Každý člověk při řešení všech problémů vychází ze svých zkušeností (tzv. hloubkové znalosti). Také expertní systém musí být schopen **využívat hloubkové znalosti** ("používat zdravý rozum"). Každý expertní systém by měl být zkonstruován tak, aby dokázal předvídat následky všech svých činností (navržených řešení). "*Robotu, který má vařit špagety na*

plynovém vařiči, by nemělo zůstat utajeno, že vařící voda má tendenci z přikrytého hrnce utíkat. Měl by "vědět" to, že přetékající kapalina může uhasit plamen vařiče, aniž by ovlivnila funkci přívodu plynu do hořáku. Bylo by víc než nešťastné, kdyby v takovém případě uvažovaný robot nebyl vybaven schopností předvídat vzrůst koncentrace plynu v místnosti a následný vztah k druhům ohrožení, jaké nás zkušenost učí spojovat s uvažovaným scénářem, např. nebezpečí exploze (v případě přítomnosti otevřeného ohně) či udušení každé živé bytosti v místnosti."

Expertní systém musí být schopen **vysvětlit a zdůvodnit závěry** a **položít** vhodný **doplňující dotaz**. V současnosti jsou běžnou součástí expertních systémů elektronické slovníky, encyklopedie, katalogy apod., které může uživatel využívat k získávání informací o určitých faktech, s nimiž se při práci (v dialogu) s expertním systémem setká.

Báze dat

K řešení určitého konkrétního problému je třeba poskytnout systému data o daném případě. Tato data se ukládají do **báze dat** (množina údajů k danému případu), a takto se "dosadí" do obecně formulovaných znalostí z báze znalostí. Konkrétní data poskytuje uživatel, a to v **dialogovém režimu** s počítačem. Tento dialog evokuje dialog nezkušeného odborníka s expertem. Úlohou expertního systému v tomto dialogu je dotazovat se co nejlépe na informace k dané problematice, analyzovat odpovědi uživatele a na jejich základě (a na základě obecných znalostí uložených v bázi znalostí) zkonstruovat závěr, příp. navrhnout řešení.

Velmi důležitým a rozsáhlým modulem, který je součástí expertních systémů je **komunikační modul**. Tento modul má obecně za úkol zabezpečovat uživateli "přátelské" (user-friendly) chování systému ve všech fázích a režimech činnosti. Konkrétně komunikační modul vykonává následující funkce:

- zabezpečuje plynulý dialog s uživatelem,
- poskytuje relevantní informaci o průběhu konzultace (např. zobrazuje jméno právě využívané báze znalostí, tiskne výsledky),
- poskytuje informace z báze znalostí,
- poskytuje vysvětlení a zdůvodnění (např. co je právě zkoumáno, proč byl položen daný dotaz).

Vysvětlovací část komunikačního modulu:

- zprostředkovává on-line interakci s hypertextovým či hypermediálním systémem (ten obsahuje hierarchicky uspořádaná fakta a znalosti z předmětné oblasti, a to nejen v textové podobě),
- komunikuje v přirozeném jazyce,
- zabezpečuje porozumění pokynům.

Dialog s uživatelem se někdy nahrazuje přímým měřením údajů na reálných objektech či jejich vyhledáváním v bankách dat.

Řídicí mechanismus

Řídicí mechanismus je programový modul, který předem udává strategii využívání znalostí z báze znalostí, zprostředkovává komunikaci mezi bází znalostí a bází dat (resp. bází znalostí a uživatelem expertního systému).

Expertní systémy (ES)

http://cs.wikipedia.org/wiki/Expertní_systém

Výhody ES:

- Poskytuje stále stejné výsledky, může pracovat 24 hodin denně, nemusí si brát dovolenou, nedá výpověď, rozhodování expertního systému neovlivňuje únava, časový stres a jiné faktory.
- Dokáže svůj výrok jednoznačně zdůvodnit.
- Pokud je potřeba více expertů, stačí je pouze kopírovat.

Nevýhody ES:

- Lidský expert v dané oblasti nedokáže své vědomosti expertnímu systému předat přímo, ale je odkázaný na prostředníka – expertního inženýra.
- Lidský expert obvykle nedokáže podrobně popsat všechny aspekty, které se podílí na jeho rozhodnutí.
- Lidský expert provádí svá rozhodnutí také tak, že je nedokáže nijak přesně zdůvodnit. Jednoduše proto, že nezná, nebo zapomněl zdroj svého poznání a tvrdí: „Nevím, ale vždycky to dělám tak a všichni to tak v tomto případě dělají.“
- Lidský expert má další vlastnosti, které do expertního systému nelze vůbec promítnout – tzv. selský rozum, intuici, schopnost rozpoznat velmi vzácné výjimky a okamžitě se jim přizpůsobit.
- Na lidském rozhodování se podílí i další znalosti a schopnosti, které s problémem na první pohled přímo nesouvisí – všeobecný přehled, lidská životní zkušenost a moudrost, zdravý úsudek, vtip apod.

Rozdíl mezi ES a jiným počítačovým programem

Běžný počítačový program je obvykle navržen tak, že zpracovává nějaká vstupní data a pomocí algoritmů, které jsou naprogramovány v určitém programovacím jazyce, dospěje k nějakým závěrům – výsledkům.

Při tvorbě počítačového programu proto musí nejprve specialista (počítačový analytik) velmi přesně a detailně popsat řešení daného problému a rozdělit řešení na dílčí kroky, které jsou algoritmizovatelné. Podle tohoto popisu pak programátor vytvoří počítačový program. Část inteligence programu může být uložena mimo samotný program v souborech nebo databázích, část inteligence je realizovaná v programovém kódu samotném.

Expertní systém používá úplně jiný model. Veškerá inteligence je uložena mimo programový kód. Programový kód řídicího mechanismu má za úkol pouze vyhodnocovat stav, který je ovlivněn expertními znalostmi uloženými v bázi znalostí a informacemi získanými z okolního světa (například odpovědi na otázky kladené uživateli, výsledky měření nějakých čidel apod.). Na základě toho expertní systém samostatně rozhoduje o tom, zda je schopen poskytnout expertní radu nebo zda je potřebné získat ještě další informace. Expertní systém se dokáže rozhodnout i při odpovědích, které nejsou úplně přesné (asi ne, nevím, velký, hodně velký apod.) nebo si vzájemně protirečí.

Vytváření expertního systému proto pobíhá jinak. Řídící mechanismus je nezávislý, a tak nový expertní systém znamená vytvoření nové báze znalostí. Transformovat poznatky lidského experta do této podoby není vůbec jednoduché. Expertní znalosti nejsou reprezentovány pouze holými fakty (tzn. všechno co se musel expert naučit nazpaměť), ale i vztahy mezi jednotlivými aspekty problému se všemi pravidly i výjimkami. Obvykle tuto činnost provádí specialista – znalostní inženýr. Tento mezičlánek je kritickou slabinou dnešních expertních systémů. Proto ještě dnes neexistují softwaroví daňoví poradci, účetní, právníci apod.

Strategie usuzování

Expertní systémy používají při řešení úlohy dvě základní strategie procesu usuzování.

Dopředné řetězení

Jedná se o usuzování řízené daty (dopředné řetězení, *forward chaining*).

Expertní systém postupuje tak, že získává potřebná data a na jejich základě se rozhoduje. Používá se při řešení problému zahrnující syntézu (navrhování, konfigurace, plánování apod.).

Příklad

Expertní systém mi může vyřešit problém, co si dám dnes k večeři. Zeptá se, na co mám chuť (má to být něco sladkého, nebo pikantního?), jaké mám zásoby v ledničce (máš kuřecí maso?) a ve spíži (máš rýži?), a podle toho doporučí ten správný recept.

Zpětné řetězení

Jedná se o usuzování řízené cíli (zpětné řetězení, *backward chaining*).

Expertní systém postupuje tak, že vybere možný závěr a pokouší se dokázat jeho platnost hledáním dat, které tento závěr podporují. Tato strategie je vhodná pro diagnostické problémy, které mají malý počet cílových hypotéz.

Příklad

Dnes večer přijdu domů a budu mít chuť na Kung-pao. Expertní systém mi potvrdí, zda je to možné (na základě stavu mé ledničky a spíže). Na rozdíl od mechanického porovnání seznamu obsahu ledničky s receptem, expertní systém potvrdí, že je to možné i v případě, že bude chybět nějaká nepodstatná záležitost nebo pokud bude možné nahradit požadovanou surovinu něčím jiným.

Expertní systémy (ES)

<http://programujte.com/clanek/2008020100-expertni-systemy-i/>

Úvod

Většina stávajících informačních systémů provádí takové výpočty, pro které existuje přesný postup řešení, kterému říkáme algoritmus. Stále častěji se na druhé straně setkáváme s úlohami, pro které postup řešení nejde najít. Můžeme sice navrhnout projekt s kvalitním interfacem, který bude zajišťovat rutinní funkce, a intuitivní rozhodování ponechat na odborníkovi, ale v současné době již můžeme vytvořit takový systém, který na základě poskytnutých znalostí najde řešení sám.

Protože takové systémy využívají vložených znalostí, říká se jim znalostní systémy. Speciálním případem znalostních systémů jsou **expertní systémy**. Využívají znalostí, které dodali odborníci – experti, z toho je odvozen jejich název.

Závěr

Seznámili jsme se dosud se třemi různými metodami umělé inteligence. Při vývoji inteligentních systémů je vždy důležité zvážit, která metoda je pro určitý typ úlohy vhodná a zda vůbec systém umělé inteligence použít. Následující přehled nám poskytuje alespoň základní vodítko.

- Pokud existuje **přesný algoritmus řešení**, umělou inteligenci vůbec nepoužijeme. Příkladem může být účetnictví, výpočet daní a pojistného apod.
- **Evoluční systémy** se vyznačují tím, že existuje obrovské množství alternativ a neexistuje předpis, jak najít tu nejlepší. Existuje však jednoznačná funkce, jak každou alternativu ohodnotit. Příkladem jsou optimalizační systémy.
- **Učící se systémy** využívají různé typy neuronových sítí a dokážou se učit z příkladů, generalizovat a zevšeobecňovat. Používají se tam, kde existuje rozsáhlá množina naměřených dat a kde tušíme, že mezi nimi existují souvislosti, ale nedokážeme je popsat. Vhodné jsou pro předpovědi ekonomických a technologických veličin, řízení procesů, které nelze popsat algoritmicky, ale probíhají podle jistých, nám neznámých nebo v čase proměnlivých zákonitostí.
- **Expertní systémy** využijeme tam, kde se má informační systém samostatně rozhodovat a nalézt řešení, přičemž existuje zpravidla značný počet (až stovky) pravidel, která nám sdělili odborníci v dané profesi a která mohou při správném postupu k řešení vést. Příkladem jsou opět složité technologické procesy, kde ovšem pravidla nejen tušíme, ale dokážeme přesně popsat. Známým příkladem jsou programy hrající šachy, ale velký význam mají praktické aplikace v průmyslu, např. chemii, strojírenství apod.

Expertní systémy (ES)

Pavel Šenovský: Expertní systémy, VŠB – Technická Univerzita Ostrava, Fakulta bezpečnostního inženýrství, Ostrava, 2007

3.1 Stručná historie expertních systémů

Základy umělé inteligence jako vědy položil v roce 1950 britský matematik a kryptoanalytik Alan Turing ve svém článku *Computing Machinery and Intelligence*. V tomto článku mimo jiné definuje jakým způsobem určit, že systém je skutečně „inteligentní“. Tato pravidla jsou dnes označována jako tzv. *Turingův test*.

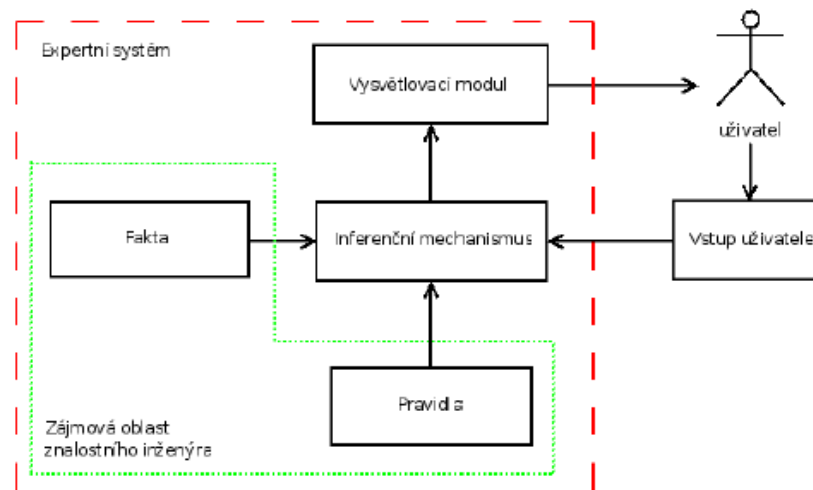
Turingův test spočívá v použití imitační hry. Této hry se účastní několik lidí, kteří mají za úkol imitovat počítač, a počítač, který má za úkol imitovat člověka. Hráč musí rozlišit, aniž by svého protivníka viděl, kdo je počítač a kdo skutečný člověk. Rozhodování probíhá na základě pokládání otázek hráčem a analýze odpovědí protihráče (člověka nebo stroje).

Pro úplnost dodejme, že v historii nebyl zaznamenán případ špatné klasifikace počítače, ale několika lidem se povedlo úspěšně imitovat stroj.

V roce 1965 nastává další milník v oblasti umělé inteligence, v tomto roce totiž Joseph Weizenbaum sestavuje program ELISA. Základním úkolem ELISy je parodovat psychoanalytika. Jedná se o konverzační program, který pracuje tak, že analyzuje věty které se mu předkládají a modifikuje je tak aby z nich položil otázku.

V roce 1974 se konečně objevuje první expertní systém nazvaný MYCIN. Tento systém je zaměřen na doporučování vhodných léků na základě analýzy zdravotní karty pacienta, zejména s ohledem na vzájemné působení jednotlivých léků.

Podívejme se na obecné schéma expertního systému



Obr. 23: Schéma expertního systému

Inferenční mechanismus z obr. 23 tvoří jádro expertního systému. Jeho úkolem je zpracovávat údaje zadané uživatelem a porovnávat je s údaji zjištěnými z báze znalostí (tvořené pravidly a fakty). Výsledek této činnosti je předkládán uživateli ve formě vysvětlovacího modulu.

Inferenční mechanismus je v expertním systému zabudován napevno, základním problémem použití tak bude z našeho hlediska formulace báze znalostí, tedy naplnění expertního systému.